A platform to support web site adaptation and monitoring of its effects: A case study

Conference Paper · January 2008

6 authors, including:


Marcos Aurélio Domingues
State University of Maringá, Maringá, Brazil


Alípio Mário Jorge
University of Porto


Carlos Soares
University of Porto


José Paulo Leal
University of Porto

A Platform to Support Web Site Adaptation and Monitoring of its Effects: A Case Study

Marcos A. Domingues
Fac. de Ciências, U. Porto
LIAAD-INESC Porto L.A., Portugal
marcos@liaad.up.pt

Alípio M. Jorge and **Carlos Soares**
Fac. de Economia, U. Porto
LIAAD-INESC Porto L.A., Portugal
{amjorge,csoares}@fep.up.pt

José Paulo Leal
Fac. de Ciências, U. Porto
CRACS-INESC Porto L.A., Portugal
zp@dcc.fc.up.pt

Pedro Machado
LIAAD-INESC Porto L.A., Portugal
Univ. do Porto, Portugal
pedro.machado@liaad.up.pt

Abstract

In this paper we describe a platform that enables Web site automation and monitoring. The platform automatically gathers high quality site activity data, both from the server and client sides. Web adapters, such as recommender systems, can be easily plugged into the platform, and take advantage of the up-to-date activity data. The platform also includes a module to support the editor of the site to monitor and assess the effects of automation. We illustrate the features of the platform on a case study, where we show how it can be used to gather information not only to model the behavior of users but also the impact of the personalization mechanism.

Introduction

The management of Web sites has become a constant demand for new information and timely updates due to the volume of services and content that site owners must provide to satisfy the complex and diverse needs and behaviors of the users. Currently, site editors must not only keep the contents of the site up-to-date, but also permanently choose the services and the navigational structure that best helps achieve the aims of both the user and the owner of the site. Such constant labor intensive effort implies high personnel costs.

Several of the management activities of a Web site may be automated, such as the retrieval of new and relevant content, monitoring of existing content and structure, automatic recommendation and personalization. One of the goals of automation is the reduction of the site editor's effort, and consequently of the costs for the owner. The other goal is that the site can adapt itself to the behavior of the user, improving the browsing experience and helping the user in achieving his/her own goals when these are in accordance with the purpose of the site.

In this paper we describe a case study that illustrates the possibilities of the Site-O-Matic (SOM) platform, which enables the easy integration of personalization mechanisms as well as their analysis and evaluation (Domingues et al. 2007). We have used the platform as a personalization facility for IGZP, the site of a computer science course.

Here, personalization consists of a recommender mechanism based on an item-based collaborative filtering technique (Karypis 2001). The IGZP Web site has a simple structure including a search engine and a tree view menu with hyperlinks to about 289 pages. These contain notes and exercises related to graphical user interfaces (Figure 4). The site has about 315 accesses daily. The Web site monitoring tool used to assess the effects of the personalization mechanism, called EdMate, is also based on the SOM platform. EdMate is a tool that enables the editor of a Web site to monitor the quality of its content, structure and usage and, thus, can also be used to monitor the effect of adaptations. It provides numerical and graphical information in a flexible way, which can be explored by the editor with different levels of detail in order to obtain a clear picture of the status and evolution of the site.

This paper is organized as follows. We start by presenting the architecture of the Web adaptation platform. Then we present the Site Activity Information Service which provides data for Web site automation. We describe the adapters used for personalization of the IGZP Web site and present EdMate, a tool for Web site evaluation. Then we discuss the evaluation of the adapters and present some results obtained on this experiment. Finally, we describe some related work and present conclusions and future work.

Architecture

The architecture of the platform for Web adaptation (Domingues et al. 2007) is presented in Figure 1. In this platform, adaptation consists of automatically and dynamically introducing changes in the content that is displayed in the browser. For example, if the adaptation is a list of recommended hyperlinks to be shown to a particular user at a given moment, these hyperlinks are added to the original Web page on the fly, taking into account what the system knows about the user and the content. To implement an adaptation mechanism using this platform it is only necessary to include in the Web page (or template) to be adapted an Ajax script (MDC 2007) that is executed on the client side. This script generates adaptation requests to be handled by the adapters and renders the output of the adaptation on the Web page. Additionally, the script collects usage data that can be used to monitor the site and the adaptations, as illustrated in this paper.

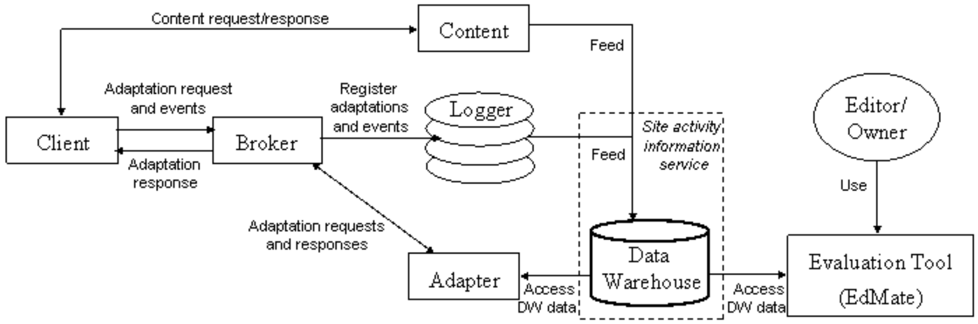

Figure 1: Architecture for the Site-O-Matic platform.

Communication between the Client and the Adapter is mediated by a Broker and uses XML based messages, that separate Web adaptation from the site's specific HTML layout. The Broker is responsible both for storing usage data in the Logger database and for routing adaptation messages to the appropriate Web adapters. Logs about Web site usage and adaptation (including positive and negative user feedback, relative position of the adaptation and the adapter used) are recorded in the Logger database.

Adapters, such as the recommender system we implemented (Section *Adapters*), are pluggable modules that use information from the Site Activity Information Service component. This component consists of a data warehouse that stores information about the Web site activity (page views, hyperlinks, access logs, events, adaptations). The data warehouse is also used by the EdMate tool to generate reports about the Web site, which are accessed by the editor/owner using a Web browser (Figure 3).

Site Activity Information Service

The Site Activity Information Service (SAIS), which collects and compiles information regarding activity in the Web site, is responsible for providing information for Web site adaptation and analysis. In (Domingues et al. 2007) the SAIS has been designed as a data warehouse (Kimball and Merz 2000; Kimball and Ross 2002) whose *fact* and *dimension* tables store information about accesses, content and structure. There are other data, specific to each Web site, which can enrich the information provided by SAIS. For example, content management systems store metadata (data describing content) which varies significantly across systems. These data can be used to enrich the information about the content. In this case study we have exploited solely generic Web site data.

In Figure 2 we present the snowflake schema of the data warehouse. To allow more than one level of dimension tables we have moved from a star schema, proposed in (Domingues et al. 2007), to a snowflake one. The tables *Parameter Page*, *Parameter Referer*, *Adaptation* and *Event* represent complex attributes for the *Dimension Tables* ***Page*** and ***Referer*** and for the *Fact Table* ***Usage***. The tables *Parameter Page* and *Parameter Referer* store the *name* and *value* of the parameters of an URI. The table *Adaptation* stores information about adaptations inserted in each Web page accessed. In this table we record the *adapter* identification, the *parameters* used by the adapter, the page element/tag where the adaptation was inserted (*item*) and an XML *representation* of the adaptation. In the table *Event* we have information about certain events occurring on the Web pages, such as hyperlink clicking, text selection or even scrollbar moving. The table stores the *type* of event and the moment (*date*) when it took place. The relation between tables *Adaptation* and *Event* enables us to analyse the adaptation behavior of the site. These tables are fed with information retrieved from the Logger database. A short description of the other tables of the data warehouse is presented in Table 1. A more detailed discussion about these tables is given in (Domingues et al. 2007).

As we can see, what is proposed in Figure 2 is not a traditional data warehouse schema (Kimball and Ross 2002). However, with this schema we can provide richer data to the adapter and also to monitoring tools. We can, for instance, analyse for each Web access which adaptations were inserted in the page and how the user reacted to them.

Adapters

In our Web adaptation platform (Figure 1), adapters are pluggable modules that process adaptation requests and provide a response that is delivered to the Client component. The platform specifies only the role of the adapters (not their design), and the interface they should provide. Therefore, adapters may run on different hosts and be implemented in different languages and operating systems. Some adapters may be small programs prototyping a new algorithm while others may be full-fledged components optimized for performance. The platform also supports the management of adapters. For instance, the adapters for which a suitable function is provided, will periodically be sent an event to update their models with recent data, as exemplified below.

For our case study we implemented two adapters (we call them ***adapter_1*** and ***adapter_2***). They are recommender systems whose recommendation models are based on the item-based collaborative filtering technique (Karypis 2001), where an item is an accessed hyperlink. In this case study the performance of the two adapters is compared. To build the similarity matrix between all the pairs of items, ***adapter_1***

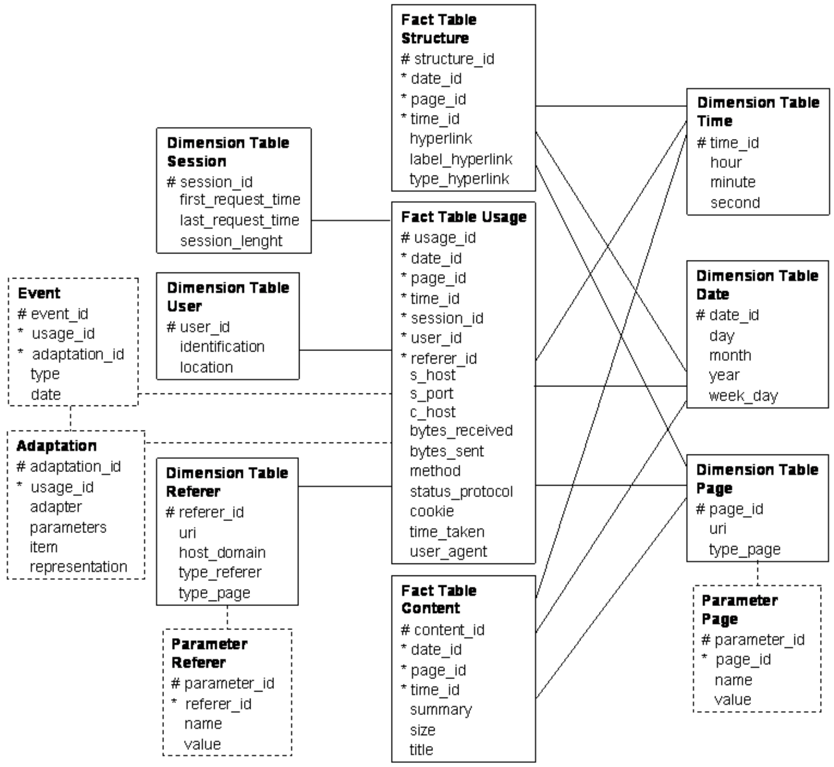

Figure 2: Snowflake schema of the data warehouse.

uses the cosine angle as similarity measure, which is given by

$$sim(i,j) = cos(\overrightarrow{i}, \overrightarrow{j}) = \frac{\overrightarrow{i} \cdot \overrightarrow{j}}{||\overrightarrow{i}|| * ||\overrightarrow{j}||}, \quad (1)$$

where $\overrightarrow{i}$ and $\overrightarrow{j}$ are vectors representing the users that accessed the hyperlink/item i and j, and "." denotes the dot-product of the two vectors.

For ***adapter_2*** we introduced an adjustment to the similarity measure, as defined below

$$sim(i,j) = cos(\overrightarrow{i}, \overrightarrow{j}) * co\_depth\_weight(i,j), \quad (2)$$

$$co\_depth\_weight(i,j) = log_2(min(depth(i), depth(j))), \quad (3)$$

where $depth(i)$ and $depth(j)$ are the current depth of items i and j on the site navigation tree and min is a function which return the minimum value. The adjustment introduced by co_depth_weight favours recommendations of items that are placed deeper in the navigation tree. This increases the probability that more specific items are recommended.

Every 24 hours the recommender models of the adapters are refreshed using data from the Site Activity Information Service. The adaptation is triggered by the platform by calling a function that is provided by the adapter. The connection to the data warehouse is made using standards mechanisms, such as Open Data Base Connectivity (ODBC) and Java Data Base Connectivity (JDBC), and the data is retrieved using SQL queries. Each adapter retrieves all the hyperlinks accessed by each user of the Web site as a set of baskets. Each basket is of the form $B = <id, item>$, where id and $item$ respectively identify the user and the accessed hyperlink. Additionally, ***adapter_2*** retrieves the depth for each hyperlink. The data fields used are *user_id* from the *Dimension Table* ***User***, *uri* from the *Dimension Table* ***Page***, and *label_hyperlink* from the *Fact Table* ***Structure***. Then the adapters compute a matrix with the similarities between all the pairs of items, according to the distance functions described earlier.

When a page is loaded on the client side, the adaptation request is sent to the two adapters identifying the current page and user. The adapters use the identification of the current page (uri) to select the most similar items, based on the similarity matrix, and generate one recommendation each, consisting of a hyperlink that is returned to the client and integrated in the rendered page (Figure 4).

EdMate Tool

EdMate is a Web based tool to monitor the quality of Web sites (Soares, Jorge, and Domingues 2005; Domingues, Soares, and Jorge 2006). Edmate provides different types of reports, which include simple statistics and more complex analysis methods. Information is presented both textually and graphically. The tool is quite flexible, enabling the editor to explore the information at various levels of detail and

Table 1: Short description of the fact and dimension tables in the data warehouse.

Tables	Description
*Fact Table **Content***	Summaries of the representation of Web page content and its changes.
*Fact Table **Structure***	Stores information about every hyperlink in the Web site, keeping the history of the Web site topology.
*Fact Table **Usage***	The table is filled with information about accesses to the Web pages of the site, extracted from access logs (Hallam-Baker and Behlendorf 1996).
*Dimension Table **Session***	The table tags sequences of events/accesses carried out by users.
*Dimension Table **User***	Stores information about users, which is very important to distinguish different types of visitors to the Web site.
*Dimension Table **Referer***	The referer dimension identifies the page containing the hyperlink that was followed to the current page.
*Dimension Tables **Time** and **Date***	Stores the date and time of each activity carried out in the Web site.
*Dimension Table **Page***	The page dimension locates and classifies each page.

also to analyse the evolution of measures.

In (Soares, Jorge, and Domingues 2005; Domingues, Soares, and Jorge 2006), EdMate was used to monitor the quality of the meta-data describing content in Web portals. Here, we have adapted this tool to evaluate the impact and quality of Web adaptation actions, using pre-processed data stored in SAIS. In Figure 3 we present a screen of the EdMate tool.

To assess the adaptations in a Web site, the tool provides different types of reports, including:

- Statistics about the number of different pages, users, accesses, sessions, hits on a page and their evolution in time;
- Percentage of accesses to the Web site that follow from recommendations provided by the adapters (recommendation adhesion);
- Percentage of recommendations that lead to relevant pages (recommendation efficacy). Here, we assume that a page is relevant if it is visible for longer than t seconds and it is not the last page in the session, where t is defined by the user;
- Visualization of hyperlinks recommended by access and session;
- The time spent in recommended pages.

With these reports, we can measure the impact of the adaptations (recommendations, in this case) from different perspectives. We can assess what users actually do when interacting with the site and, in particular, how they react to personalization. This enables us to better understand the behavior of users, identify usage patterns, problems with the content and structure of the site and unmet needs. It also enables us to identify issues that would not be evident in a laboratorial study. Additionally, the EdMate tool also provides an easy to use SQL interface for selection and exportation of different data sets that can be used for offline analysis.

Evaluation of Personalization Adapters

Evaluation of personalization systems is still a challenge due to the lack of understanding of which factors affect user satisfaction. Hence, different strategies must be used, which makes evaluation a difficult and time consuming process. Additionally, evaluation can be done along several dimensions, including (Anand and Mobasher 2003; Herlocker et al. 2004): User Satisfaction, Accuracy, Coverage, Utility, Robustness, Performance, Scalability.

In our case we have adopted two strategies to measure the impact of the two adapters described earlier on a live Web site (IGZP): offline and online validation. In Figure 4 we can see how the generated recommendations are presented (highlighted region). In offline validation the potential of the adapters is assessed on past usage data. In online validation, we integrate the adapters on the site and measure the impact of the adaptations when they are shown to actual users. In general, offline evaluation enables the validation of proposed adapters before they are deployed, without interfering with the working site. Online validation allows a more realistic evaluation of the adapters. The results of the two evaluation procedures are stored in SAIS and are analysed using the EdMate Tool. We also assess the users' perception of the adaptations, based on surveys filled-in by the users.

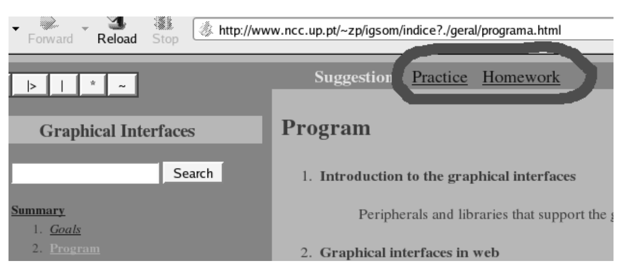

Figure 4: Recommendations as labeled hyperlinks at the academic Web site.

The experiment covers 47 days of usage of the platform in the IGZP Web site. The adapters were activated after 21 days of data collection. The first recommendation models were created with data from the first 21 days.

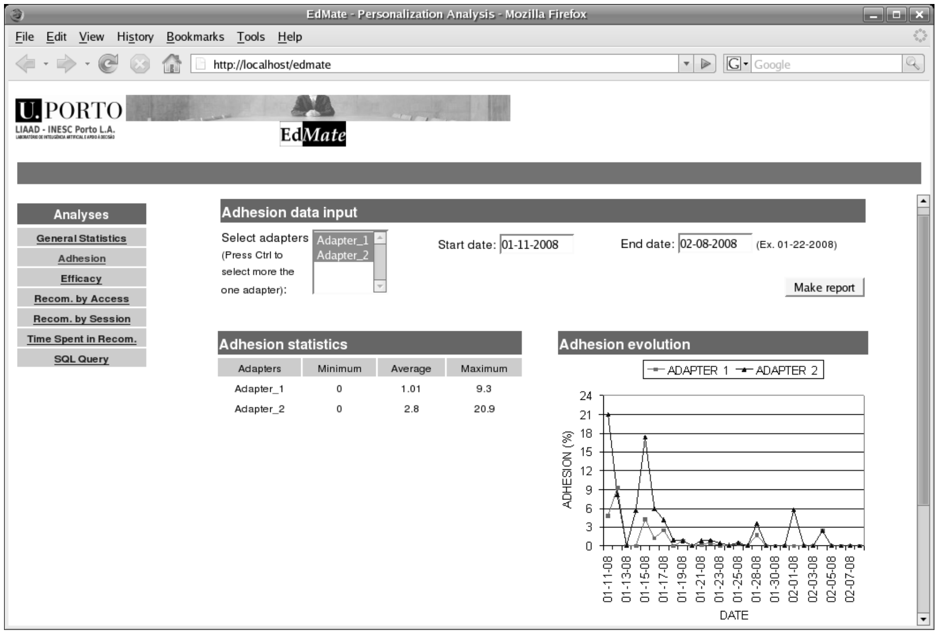

Figure 3: EdMate screen showing a kind of Web adaptation analysis.

Offline Validation

This is a common approach, in which the deployment of recommender systems is simulated and evaluated using previously collected usage data. The recommendations generated by the systems are compared to the actual behaviour of the users. With this strategy, users are not provided with recommendations, which, on one hand, means that it only provides a rough estimate of the expected performance of recommender systems. But, on the other hand, it allows us to test models without the risk of providing bad recommendations to the users. Therefore, it is especially used for preliminary testing of new adapters.

To measure the potential accuracy we used the All But One protocol described in (Breese, Heckerman, and Kadie 1998). In this protocol, the baskets in the data set are split randomly into train and test. The training set is used to generate the recommendation model (similarity matrix). From each basket in the test set we randomly delete one pair $< id, item >$. The set of deleted pairs is called the hidden set (*Hidden*). The set of baskets with the remaining pairs is called the observable set.

One model is evaluated by comparing the set of recommendations it makes (*Rec*), given the observable set, against the items in the hidden set. The set of recommendations $r_1, r_2, ..., r_N$ for a given user id is represented as $\{< id, r_1 >, < id, r_2 >, ..., < id, r_N >\}$. N is the number of recommendations produced by the model. Once we have calculated *Hidden* and *Rec*, we measure Recall, Precision and the F1 metric as defined below (Yang and Liu 1999; Sarwar et al. 2000).

$$Recall = \frac{|Hidden \cap Rec|}{|Hidden|}, \quad (4)$$

This is a global measure for the whole set of users in the test set. Recall corresponds to the proportion of relevant recommendations. It tends to increase with N.

$$Precision = \frac{|Hidden \cap Rec|}{|Rec|}, \quad (5)$$

Precision gives us the quality of each individual recommendation. As N increases, the quality of each recommendation decreases. Global recall and precision are obtained by averaging the respective individual user values.

$$F1 = \frac{2 \times Recall \times Precision}{Recall + Precision}, \quad (6)$$

F1 has been suggested as a measure that combines Recall and Precision with an equal weight. It ranges from 0 to 1 and higher values indicate better recommendations. We calculate the F1 measure from the global values of Recall and Precision.

Over the period of usage of the platform on the IGZP Web site we recorded 14,046 accesses. As the recommendation models were refreshed daily, we assessed the daily evolution of the models. For each day we ran an experiment where the access data of that day was used as the test set and the accesses of previous days were used as training set.

In Figure 5 we see the evolution of the models. We used $N = 1$, since we wanted to give clear choices to the users. Using $N = 1$ and the All But One protocol, we got the same values for Recall, Precision and F1. In the following we will only refer to recall. The model generated by ***adapter_1*** got a minimal value of 0, an average of 0.39 and a maximal value of 0.84 for Recall. For ***adapter_2*** we got a minimal value of 0.08, an average of 0.40 and a maximal value of 0.82. If we compare the daily evolution of the two models we see that both adapters had a similar behavior. Higher initial values occurred due to a great number of accesses on only two Web pages in the site: the home page of the Web site and the page with the schedule of the lessons.

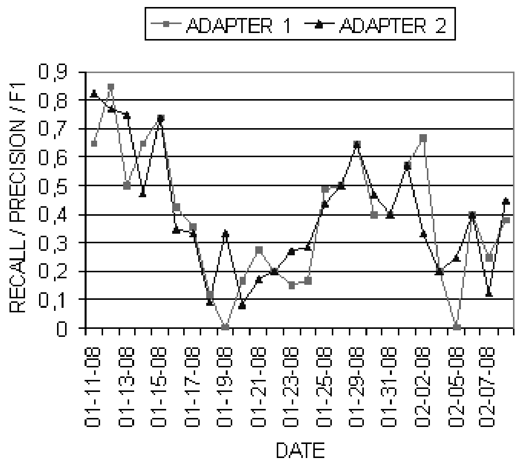

Figure 5: Daily evolution of ***adapter_1*** and ***adapter_2***.

Online Validation

Although offline validation gives us an important assessment of the potential of the adapters, we have to measure the impact of the adaptations when they are shown to actual users. Again, this information is stored in the SAIS as the Web site can be analyzed on the fly or a posteriori using the EdMate tool. In the following, we show some results obtained using the tool.

We measured the recommendation adhesion, which is calculated as the percentage of accesses to the Web site that are recommendations followed by the user.

$$Adhesion = \frac{|R|}{|A|} \times 100, \tag{7}$$

where A is the set of all accesses in the Web site and R is the set of accesses representing recommendations that were followed by the users.

We got an average rate of recommendation adhesion of 1.66 %. The low value is understandable regarding that the Web site is small and has a simple navigation structure. The evolution of the recommendation adhesion rate for ***adapter_1*** and ***adapter_2*** is presented in Figure 6.

In Figure 7 we measured the efficacy of the recommendations. This is calculated as the percentage of recommendations that lead to page visits longer than 10 seconds, but which are not the last in the session. The definition thus is

$$Efficacy = \frac{|R_{10}|}{|R|} \times 100, \tag{8}$$

where R is the set of all accesses that are a consequence of recommendations and R_{10} is the set of recommendations that lead to page visits longer than 10 seconds.

The length of the page visit is measured as the difference between two consecutive page views. This is obviously a proxy for the actual page view time, which is not available.

Comparing the values in Figure 6 and 7 we see that recommendations produced by ***adapter_2*** are more easily followed than the ones produced by ***adapter_1***, and that a user spends more time on recommendations of ***adapter_2***. This may be due to a possible interest of the users for items more specific than the general ones and the adjustment introduced in ***adapter_2***, which favours the recommendations of more specific items. The average time spent visiting the recommended pages was 21.3 seconds.

We also compared the adhesion rates in Figure 6 with the recall values of the adapters, obtained in the offline validation (Figure 5). The comparison showed us that usually when the two adapters had higher values of recall, the adhesion rates were also higher. For example, we compared the values of ***adapter_1*** for the date “01-15-08” and we got a recall value of 0.75 and an adhesion rate of 4.1%. For the date “01-17-08” we got a recall of 0.36 and an adhesion rate of 2.9%.

Regarding the efficiency of the platform, the average time a recommendation takes to be shown to the active user after a Web page is entered was 1.61 seconds. We note that we used prototype implementations of the adapters, because our main concern at this point is to test the accuracy of the adapters. The recommendation time could, thus, be improved by fine-tuning the implementation. More importantly, we this time represents the time between the request for adaptation and its delivery to the client. Given that we are using asynchronous communication with Ajax, during that period, the client is rendering the rest of the Web page. In other words, the user is not left with a blank page while waiting for the recommendation.

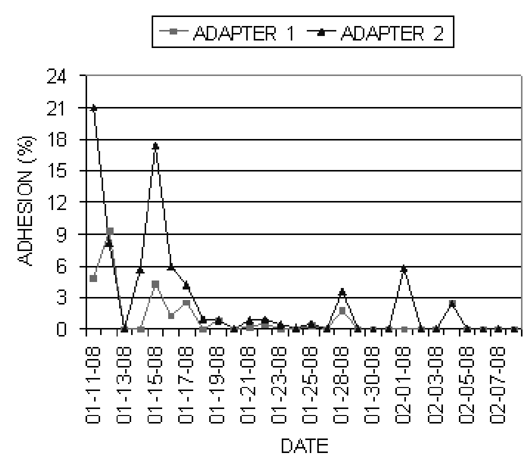

Figure 6: Evolution of the recommendation adhesion rate.

Finally, we had feedback from users about the personalization mechanism on the Web site. The questions and answers obtained with user surveys are presented in Table 2. Taking into account the *usually* and *always* answers we see that 78 % of the users noticed the recommendations, only 37 % declare that they followed them regularly, 68 % say they

Table 2: Results of the user surveys about the personalization mechanism of the site.

Questions	Answers				
	Never	Rarely	Usually	Always	**Total Result**
"Did you notice the recommendations?"	2	2	5	10	**19**
"Did you follow any recommendation?"	4	8	6	1	**19**
"Did you understand how the whole thing worked?"	3	3	9	4	**19**
"Did you find the recommendations useful?"	2	9	7	1	**19**
Total Result	**11**	**22**	**27**	**16**	**76**

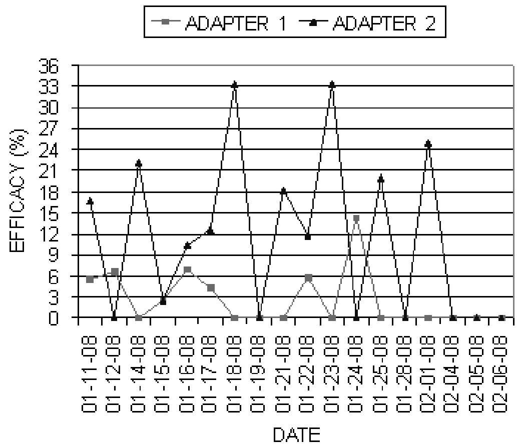

Figure 7: Evolution of the recommendation efficacy rate.

understood how the whole thing worked and 42 % found the recommendations useful.

Related Work

Mobasher *et al.* (Mobasher et al. 2002) described a platform for Web page recommendation based on usage and content profiles. In their system, the adaptation process is divided into two components: the offline one which is comprised of the data preparation and Web mining tasks to create the profiles, and the online component which uses the profiles to make real-time recommendations. In contrast to our platform, this one does not consider data about the structure of the Web site in the adaptation process.

IKUM is an integrated Web personalization platform based on content structure, user behavior and semantic information (Eirinaki, Vlachakis, and Anand 2005). More specifically, Web usage logs are enriched with semantics extracted from the content of the pages in the Web site, which are then used to produce recommendations to the user accessing semantically similar content. Although our platform has not been designed to use directly semantic information, the SAIS provides enough data for an adapter to enrich its adaptation model with semantic information. Like the former platform, IKUM also does not use the structure of the Web site to enrich the adaptation process.

AWESOME is a data warehouse-based recommender system that captures and evaluates user feedback on presented recommendations (Thor, Golovin, and Rahm 2005). The system also selects dynamically the most promising recommenders, based on measured recommendation feedback, to carry out the Web personalization. In our platform the set of adapters used by a site may also change dynamically. Changes may be manual (i.e., a new adapter is included) or automatic (i.e., a meta-adapter chooses the adapter that provides a response to a request based on previous performance of all available adapters).

Wu *et al.* (Wu, Ng, and Huang 2004) proposed an integrated data warehousing and data mining framework for Web site management, pattern discovery and personalization. The merit of the framework is that it combines multi-dimensional Web databases to support online and offline analytical process. Based on their framework, they can further propose and implement some algorithms to discover interesting user access patterns for Web site optimization, personalization, recommendation, etc.

Most of platforms presented in this section use only Web access logs gathered from the Web server, which is often inappropriate because it makes full user, transaction and/or session identification impossible or inaccurate due to two big impediments to get these data, local caching and proxy servers (Cooley, Mobasher, and Srivastava 1999). In (Thor, Golovin, and Rahm 2005) this problem was addressed using tailored application server logging to record usage information. However, this solution requires explicit user log-in. In our platform we have proposed a solution that works by including an Ajax script into every Web page. Therefore we can assess whether the user only enters the page or reads also the information on it (hyperlink clicking, text selection or even scrollbar moving). Hence, our platform uses not only access logs gathered from the server but also logs gathered from the client, high quality access logs.

Additionally, only the system proposed here and the ones in (Wu, Ng, and Huang 2004; Thor, Golovin, and Rahm 2005) provide mechanisms to evaluate the recommendations when they are shown to actual users. This type of evaluation is very important. It can reveal what users actually do in their own real contexts, and it may point out issues that had not been foreseen in offline simulations.

With respect to the platforms presented in this section, we can see that our system is most similar in nature to the work of (Wu, Ng, and Huang 2004; Thor, Golovin, and Rahm 2005). However, when it is compared to the other two platforms, our system provides a more generic framework and several facilities to carry out different types of Web personalization.

Conclusion and Future Work

In this paper we described a platform that enables the easy integration of personalization mechanisms as well as of tools that support their analysis and evaluation. We have shown a case study where we measured the impact of the recommendations (directly with user surveys, and indirectly from site activity), as well as the latency times for recommendation generation.

The main potential advantage of our proposal is that we can easily set up a number of concurrent or collaborative personalization facilities and efficiently analyse their performance and impact, using all the benefits of the pre-processed site activity data in the data warehouse (SAIS) and of the EdMate tool. The main difficulties are the communication overheads caused by the modular architecture, with some of its parts possibly relying on network connections.

We are currently working on a second experiment on a larger scale commercial Web site. We are also implementing OLAP cubes to provide multi-dimensional analyses for the EdMate tool. Next we will further exploit the multi-dimensional data available in the Site Activity Information Service to improve Web adaptation services.

Acknowledgments. Supported by the Fundação para a Ciência e a Tecnologia (SFRH/BD/22516/2005). The financial support of the POSC/EIA/58367/2004/Site-O-Matic Project (Fundação para a Ciência e a Tecnologia) co-financed by FEDER is gratefully acknowledged.

References

Anand, S. S., and Mobasher, B. 2003. Intelligent techniques for web personalization. In *Intelligent Techniques for Web Personalization (ITWP 2003), LNCS 3169*, 1–36.

Breese, J. S.; Heckerman, D.; and Kadie, C. M. 1998. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, 43–52.

Cooley, R.; Mobasher, B.; and Srivastava, J. 1999. Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems* 1(1):5–32.

Domingues, M. A.; Jorge, A. M.; Soares, C.; Leal, J. P.; and Machado, P. 2007. A data warehouse for web intelligence. In *Proceedings of the 13th Portuguese Conference on Artificial Intelligence (EPIA 2007)*, 487–499.

Domingues, M. A.; Soares, C.; and Jorge, A. M. 2006. A web-based system to monitor the quality of meta-data in web portals. In *WI-IATW '06: Proceedings of the 2006 IEEE/WIC/ACM international conference on Web Intelligence and Intelligent Agent Technology*, 188–191. Washington, DC, USA: IEEE Computer Society.

Eirinaki, M.; Vlachakis, J.; and Anand, S. S. 2005. Ikum: An integrated web personalization platform based on content structures and user behavior. In Mobasher, B., and Anand, S. S., eds., *Intelligent Techniques for Web Personalization (ITWP 2003), LNAI 3169*, 272–288.

Hallam-Baker, P. M., and Behlendorf, B. 1996. Extended log file format: http://www.w3.org/tr/wd-logfile-960221.html. Access date: 01/02/2008.

Herlocker, J. L.; Konstan, J. A.; Terveen, L. G.; and Riedl, J. T. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)* 22(1):5–53.

Karypis, G. 2001. Evaluation of item-based top-n recommendation algorithms. In *CIKM'01: Proceedings of the 10th International Conference on Information and Knowledge Management*, 247–254.

Kimball, R., and Merz, R. 2000. *The Data Webhouse Toolkit: Building the Web-Enabled Data Warehouse*. New York, NY, USA: John Wiley & Sons, Inc.

Kimball, R., and Ross, M. 2002. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. New York, NY, USA: John Wiley & Sons, Inc.

MDC. 2007. Mozilla developer center. asynchronous javascript and xml (ajax): http://developer.mozilla.org/en/docs/ajax. Access date: 01/02/2008.

Mobasher, B.; Dai, H.; Luo, T.; and Nakagawa, M. 2002. Discovery and evaluation of aggregate usage profiles for web personalization. *Data Mining and Knowledge Discovery* 6(1):61–82.

Sarwar, B.; Karypis, G.; Konstan, J.; and Riedl, J. 2000. Analysis of recommendation algorithms for e-commerce. In *Proceedings of the 2nd ACM Conference on Electronic Commerce*, 158–167.

Soares, C.; Jorge, A. M.; and Domingues, M. A. 2005. Monitoring the quality of meta-data in web portals using statistics, visualization and data mining. In *Proceedings of 12th Portuguese Conference on Artificial Intelligence (EPIA 2005), LNAI 3808*, 371–382.

Thor, A.; Golovin, N.; and Rahm, E. 2005. Adaptive website recommendations with awesome. *VLDB Journal* 14(4):357–372.

Wu, E. H.; Ng, M. K.; and Huang, J. Z. 2004. A data warehousing and data mining framework for web usage management. *Communications in Information and Systems* 4(4):301–324.

Yang, Y., and Liu, X. 1999. A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99)*, 42–49.