International Journal on Artificial Intelligence Tools
Vol. 16, No. 5 (2007) 793–828

World Scientific
www.worldscientific.com

BUILDING A KNOWLEDGE BASE FOR IMPLEMENTING A WEB-BASED COMPUTERIZED RECOMMENDATION SYSTEM

JUAN D. VELÁSQUEZ

Department of Industrial Engineering, University of Chile
República 701, Santiago, Chile
jvelasqu@dii.uchile.cl

VASILE PALADE

Computing Laboratory, University of Oxford
Parks Road, Oxford, OX1 3QD, UK
vasile.palade@comlab.ox.ac.uk

Received 11 May 2006
Accepted 13 December 2006

Understanding the web user browsing behaviour in order to adapt a web site to the needs of a particular user represents a key issue for many commercial companies that do their business over the Internet. This paper presents the implementation of a Knowledge Base (KB) for building web-based computerized recommender systems. The Knowledge Base consists of a Pattern Repository that contains patterns extracted from web logs and web pages, by applying various web mining tools, and a Rule Repository containing rules that describe the use of discovered patterns for building navigation or web site modification recommendations. The paper also focuses on testing the effectiveness of the proposed online and offline recommendations. An ample real-world experiment is carried out on a web site of a bank.

Keywords: Web usage mining; knowledge bases; web-based systems; computerized recommender systems; adaptive web sites.

1. Introduction

The last years have witnessed an explosion in business conducted via the Web, illustrated by the growth in the number of web sites and visits to these sites. The result is a massive and growing quantity of data originated in the Web, also called web data.

A good web site should help the users to find the information they are looking for, by having a well-organized web site structure and content, as well as by providing navigation recommendations. A recommender system can improve the

relationship between the customers and the company that owns the web site, which means that the company will be able to attract more easily new customers and retain the old ones. The key in web personalization is to understand the user's desires and needs, and this can be done by applying various mining techniques on web usage data.

Web mining techniques[23,22,49] emerged as a result of the application of data mining theory to pattern discovery from web data. Web mining is not a trivial task, considering that the web is a huge collection of heterogeneous, unlabelled, distributed, time variant, semi-structured and high dimensional data.[28] Web mining must consider three important tasks: data preprocessing, pattern discovery and pattern analysis.[37]

These techniques can be used to provide user behavior patterns and preferences, which can be later validated by human experts. They can often suggest ways about how the patterns are to be used[39] for making recommendations.[44] One result is the development of web personalization systems, where the knowledge representation is implemented easily by using a common programming language like Perl, PHP, Java, etc. In general, the knowledge representation must consider changes in and to the web itself, i.e., changes in the web site structure and content, as well as with respect to the user behavior.

This paper describes how to build a Knowledge Base (**KB**) for implementing a web-based computerized recommender system. The KB is the main repository for the user behavior patterns extracted from web data by using various web mining techniques.[42,44] Making use of a data warehouse architecture,[6,7,19] two repositories for storing the information and the knowledge extracted from web data, respectively, are defined. The first repository stores information taken from web logs and web pages. The discovered knowledge requires a more complex repository. Hence, the KB[14] is composed of a Pattern Repository that contains the patterns extracted from web data and a Rule Repository that contains rules about how to use the patterns. Both repositories represent the information source used by different types of knowledge users in order to perform navigation or web site modification recommendations. The knowledge contained in the KB can be used by a human user or an artificial system, such as an intelligent web site,[29,41] and contribute to improving the relationship with a prospective user of the web site.

The paper is structured as follows. Section 2 is a short introduction about knowledge representation. An introduction to web-based computerized recommendation systems is provided in Section 3. The methodology to extract significant patterns from web data is introduced in Section 4. The construction of a KB for storing the knowledge extracted from web data is detailed in Section 5. In order to test the effectiveness of the proposed methodology, a real-world experiment is performed and shown in Section 6. Section 6 also presents thouroughful investigations on how to test the effectiveness of the proposed online and offline recommendations. Finally, some conclusions are drawn in Section 7.

R_1:	If BelongCluster(c_A) Then RecommendationPage(p_8, p_{15}, p_{28})
R_2:	If VisitPage(p_3) and SpentTime(t_4) Then RecommendationPage(p_{22})
R_3:	If CountPageVisit(p_i) $<$ D Then DeletePage(p_i)
...	...
R_n:	If ...

Fig. 1. Rules for representing the extracted knowledge.

2. Knowledge Representation

Knowledge Representation (KR) is the first task to take into account in developping an automatic system that uses the knowledge discovered from web data to make navigation or web site modification recommendations. Finding a proper method of knowledge representation is not a trivial task.

2.1. *Representing knowledge as rules*

The easiest way to to represent knowledge is to define a set of rules that describe how the discovered patterns are to be used.[8] The rules often specify recommendations, directives and strategies. In a computational form, they are expressed as instructions ***If*** ⟨*condition*⟩ **Then** ⟨*recommendation*⟩.

These expressions allow to easily represent the expert knowledge. However, when the number of rules increases, it becomes difficult to decide which rule is most appropriate to be applied.

The rules associate facts with actions (recommendations) through matching facts and conditions, as shown in Fig. 1.

In this example, if the user visits the page p_3 and spends time t_4 on it, then the recommendation is "go to page p_{22}". Also, if the user browsing behavior belongs to cluster c_A, then the recommended pages to visit are p_8, p_{15}, p_{28}.

2.2. *Knowledge repository*

Maintaining the discovered knowledge is a key problem.[44] A good approach is by storing it in a knowledge repository, employing a similar method used for data. However, we have to bear in mind that knowledge is more complex than just simple data. It consists of patterns discovered after processing data, which are translated into rules on how to use the patterns.

The KB is a general structure for storing facts (patterns) and rules that govern their use. A typical implementation is by keeping track of the rules that share common wisdom.[14] From a practical point of view, a KB must be able to maintain

rules in an easy way. This becomes a complex task when the problem conditions change in time, as it is the case with the knowledge extracted from web data.

3. Web-Based Computerized Recommendation Systems

A well-organized web site structure and content should help the users find the information they are looking for. However, in practice, this does not happen with many of the web sites available. Sometimes the web site structure is complex, hiding useful information and causing a "*lost in hyperspace*" feeling to the user. On the other hand, when the web site contains simple context, like free text only, it may not become attractive to users.

The above situation is inherited from the very early days of the Web. Web designers had always to deal with the following question: "how can we prepare the right web site structure and content in the right moment for the right user?" The answer is not simple and, for the moment, it seems that the key is in understanding the user behavior in a web site, and in using this knowledge to construct systems for personalizing the web site for individual users.

This section describes the main personalization approaches for web-based systems. Special attention will be paid to the adaptive web sites and their contribution to the new portal generation.

3.1. *Recommendation systems*

It is a common practice that any time we want to satisfy a personal desire or need, we ask for help from a person that we consider more advanced in the topic of our interest. Let consider the follow situation in everyday life. A person has a health problem and needs to see a doctor. One way is to search for information about medical practitioners in the area by using several means, e.g., Internet, newspapers, yellow pages, etc. Another usual alternative is to ask some friends for a recommendation about a good doctor or medical institution. The last approach is very common, persons usually ask for recommendations, because the best way to avoid mistakes is by using the experience acquired by others.

We are constantly asking for recommendations for buying, eating, etc., and when a person or institution gives us good recommendations, a very special bond is created. Some experts call this "*creating customer loyalty*".[17] When the business is small, it is not difficult to advise the customer and provide recommendations, but when the business is big or growing, the number of assistants required for providing good recommendations to customers could exceed the physical capacities of the place where the business is based. Also, it would be economically counterproductive if the assistants have to attend people all day, including here companies with personal working in shifts. How to reduce the number of assistants, but support the customer queries? Again, the information technologies seem to offer the answer, by using pre-defined actions when facing a question, in other words something like an artificial assistant.

Artificial recommender systems attend to emulate the human recommendation, by tracking past actions performed on a group of persons (for instance products acquired, Frequently Asked Questions, etc.) in order to make new recommendations to an individual person.

Formally, the recommendation problem can be expressed as follows[2]: Let $U = \{u_1, \ldots, u_m\}$ be the set of all users and $I = \{i_1, \ldots, i_n\}$ be the set of all possible items (books, CDs, DVDs, etc.) to be recommended. Both sets depend on the business and could contain millions of elements. Let Γ be a function that measures the usefulness of item i_k for the user u_j, i.e., $\Gamma : UxI \rightarrow R$, with $R = \{r_1, \ldots, r_l\}$ the set of nonnegative values for Γ function. Then

$$\forall\ u_j \in U\,, \quad I'_{u_j} = \arg\max_{i_k \in I} \Gamma(u_j, i_k)\,, \tag{1}$$

is the set of items to be recommended to user u_j.

The Γ function depends on the recommendation system implementation, but it is usually represented by a rating, for instance, "*the most requested products*" sorted list.

3.2. *Web-based recommender systems*

In early stages, the web-based recommender systems were seen only as a curiosity, but very quickly commercial companies realized the potential of these new tools for increasing and retaining the number of virtual customers.

The interaction between an user and the web site is stored in the web log files. A recommender system uses these data for extracting user patterns and preferences, and generate useful recommendations.

Recommender systems have proven their effectiveness in improving the relationship between the users and the web site, which, from a practical point of view, means an increase in the company's sales and getting a larger virtual market segment.[35] Of course, there were a lot of cases where the recommender systems did not work and the company lost both money and customers. However, it is a consensus thinking that the current web sites will need some kind of recommender systems for supporting the new user requirements and expectations.

In the traditional market, a store like a supermarket offers a limited amount of products for its customers. It is because the physical space and the local customer preferences impose a very selective amount of products. In the digital market, we encounter a completely different situation. Now, the customers are distributed around the world, and the physical space is only an old-fashion concept. Many companies that offer their products through the Web have to satisfy a wide demand, because the customer preferences can be extremely diverse in this case. This situation expresses a strong need in the digital market: "*a customized product for each customer*".[30]

One of the most successful web-based recommender system was developed by Amazon.com. In the own words of its CEO, Jeff Bezos, "*if I have 3 millions*

customers on the Web, I should have 3 million stores on the Web". Amazon.com understood very quickly the need to develop systems to customize the virtual purchase, by using recommender tools.

The web-based recommender systems are mainly used in e-commerce for "*suggesting products to customers and providing customers with information to help them decide which products to purchase*".[35] A classic recommender system's suggestion about a product includes personalized information and an evaluation table that summarizes the opinions of other customers that have bought the product in the past (collaborative filtering). A most advanced version of a recommender system will add information about other complementary products (cross selling), in the form "*others customers that had bought X, also had bought Y and Z*.

Today, recommender systems for e-commerce is an unquestionable need because they allow to:

- Transform users into customers. Every day, a commercial web site receives a lot of visits. Some of these visits are performed by customers and others by new users that are looking for a product or service information. In most cases, new users represent a non-depreciable source of potential customers; they may be even more valuable than the current web site's customers. The question is how to transform a user into a customer? A technique is to help and assist the user to find what they are looking for, through useful and personalized suggestions prepared by a recommender system.
- Increase the cross selling. When we visit a supermarket, we usually bring with us the "*shopping list*" for buying. Also, it is common that the final purchase list contains items that we have not considered in the original list. It is because the supermarket logistic and product distribution have been organized for promoting the cross-selling between related products; for instance the bread is placed near the jam and eggs, such as a person that in the original shopping list has only bread will consider to be a good idea to buy jam and eggs too. In the digital market, the situation is similar. By tracking the customer preferences and purchase behaviour, it is possible to promote the cross-selling. A very good example can be found again in Amazon.com; when we are looking for a book, we automatically receive the book information and the recommendation about other related books, that have been bought together with the book of our interest.
- Building loyalty. In the digital market, the competition for acquiring new customers is hard. It is well know that the effort to catch a new customer is nearly five times more expensive that to retain a customer. That is why companies have developed mechanisms for retaining customers by creating a value-added relationship. The loyalty construction is performed by a correct tracking of the purchase behaviour of valuable customers mainly. Some customers may not be profitable for the business overall. In the value-added process, recommender systems are used for planning the best strategy to tackle the customer preferences

and to prepare an action to retain customers, by using well know methods like special promotions, discounts, etc.

A good recommender system can improve the relationship between the customer and the company, through useful recommendations for acquiring the exact product and service that the customer is looking for. This practice is very important from the customers point of view, because it shows the company preoccupation for assisting them. However, it is necessary to consider the privacy issues. A lot of badly directed recommendations can be considered an intromission into the customer private life.[13,20]

3.2.1. *Web recommender systems, particular approaches and examples*

In early stages, the automatic recommender systems only performed simple database queries. However, due to the increase in hardware storage capacities and performance, it shortly became possible to apply more complex data analysis methods, like data mining techniques. The first recommender systems used the nearest-neighbour and collaborative filtering algorithms[31] for predicting the product purchase decision and preparing the related recommendations.

In the PHOAKS (People Helping One Another Know Stuff) system,[38] the collaboration filtering approach is applied on usenet messages for the creation of web resources recommendations. Another interesting approach was by using decision tree algorithms. This technique represents the pattern extracted from the input dataset in a tree model, where each branch represents a new decision for the user. Then, after few decisions, the user get the recommendation which is in the tree's leaf.[51]

Traditionally, clustering techniques have been used in marketing for analyzing data containing user preferences, and for extracting significant patterns from the identified clusters. In the case of web-based recommender systems, the pattern extracted by using clustering techniques are used for preparing different kinds of recommendations, which can be grouped in online and offline recommendations. The former are mainly navigation recommendations for the user[25,47] and the latter are straightforward recommendations for the web master for changing the structure and the content of the web site.[29,32,42] The above explained method for analyzing the user preferences demands a high amount of computer resources and is non-linear with the number of customers. This is an important fact to consider in a real world practical realization of a recommender system.

On the other hand, the item-based top-N recommendation algorithms (a complete survey in Ref. 15) focus on analyzing the similarities among various items for identifying similar items to be recommended. This approach does not consider directly the user behaviour in the web site, but generates item recommendations likely to be accepted by the user. Real-world successful cases of companies using top-N algorithms are Amazon.com, Book Matcher, Levi's Style Finder and My CD Now, among others.[35]

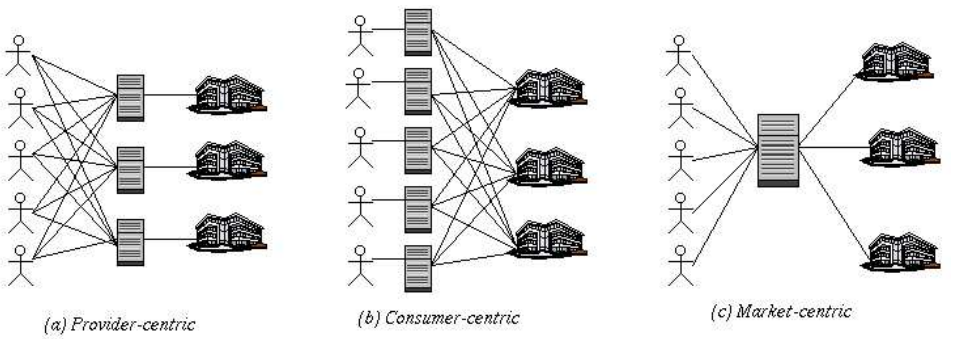

Fig. 2. Web personalization approaches (based on Ref. 1).

3.3. *Computerized personalization approaches*

The computerized personalization is a common approach to providing information hints to users. In e-commerce platforms, the personalization appears as a provider of personalized offering to one or more potential customers. Depending on the complexity of the personalization action, some platforms have developed a "*personalization engine*", i.e., a computerized system for tracking the user behaviour and providing personalized recommendations, such as information hints.

In Ref. 1, an excellent classification of current approaches for personalization is presented, which distinguishes the following architectures: Provider-Centric, Consumer-Centric and Market-Centric.

The Provider-Centric architecture (see Fig. 2a) is maybe the most commonly used in the Web. In order to extract knowledge, the provider gathers information about the user behaviour, which will be used in the personalization action. The classic example for these approaches is represented by the online shopping web sites.

The Consumer-Centric architecture is a software assistant for the user (see Fig. 2b). Whereas in the Provide-Centric approach the personalization action is aimed for increasing the provider benefit, in the Consumer-Centric approach, the personalization action purpose is to increase the consumer benefits. An example of this is e-Buckler,[2] which provides personalized online shopping. This tool gathers information about its users and offers personalized information hints. For instance, if the user is looking for a shirt and put in the e-buckler the query "shirt", the tool will aggregate another previous collected user personal information like the user shirt size and favourite colour, before executing a search in the Web.

Finally, in the Market-Centric approach (see Fig. 2c), the personalization engine works like an infomediary. Knowing the customer's needs and the provider offerings, it performs a matching and prepares the information hints. An example of this approach is Hotels.com. This platform collects information from the web site of several hotels around the world, for instance, room's price, hotel location, hotel's

agreements, etc. Then, when a user requires information about a room in a specific place, Hotel.com searches in its database, selects the hotels with the best matching with the user requirements and returns a list with information about the selected hotels, usually ordered by price, but given to the user the possibility to sort the information using another criterion.

The above approaches can be applied to develop offline and online personalization engines. However, in practice, companies that use e-commerce platforms, are more interested in online systems, because they need to provide fast answers to their customers.

From the provider point of view, the personalization engine is used mainly for yielding information hints, products and service recommendations, e-mail campaign and cross selling products, etc. In general, these activities consider two important aspects[12]:

- Personalization of the presentation, that regards the interface presented to the user, including colours, position on the screan and fonts.
- Personalization of the content; this is the most complex part, where the information must be adapted to the particular user's needs.

In Ref. 12, a good analysis of computerized personalization engines is introduced for several kinds of final users, that include not only text content information needs, but also multimedia contents. In conclusion, the effort for creating personalized web-based systems is remarkable, and this action is called "*web personalization*".

3.4. *Web personalization*

In the literature, there are several definitions for web personalization. In Ref. 26, it is defined as "*how to provide users with what they want or need without requiring them to ask for it explicitly*".

More precisely, for the web-based systems implementation, the web personalization is "*any action that adapts information or services provided by a web site to the needs of a user or set of users, taking advantage of the knowledge gained from the user's navigation behavior*".[16] In other words, in Ref. 24, the web personalization is defined as "*the process to create web-based systems able of adapting to the needs and preferences of individual users*".

From a practical point of view, the web personalization is the process where the web server and the related applications, mainly CGI-Bin[a], dynamically customize the content (pages, items, browsing recommendations, etc.) shown to the user, based on information about his/her behavior in a web site.[23,25] This is different than another related concept called "customization", where the user interacts with the web server using an interface to create his/her own web site, e.g., "My Banking Page".[40]

[a]Common Gateway Interface http://www.msg.net/tutorial/cgi/

The key of web personalization is to understand the user's desires and needs. It allows to design and construct the information repositories using the user transactions data, in order to predict the correct supply of products and services.[5]

Personalization requires to recognize patterns in the user behavior, in order to compare with the patterns of new users and, in this way, be able to make suggestions. A specific model about the user behavior and a measure that allows to compare two behaviors are required.

The personalization can be realized using a general methodology from Knowledge Discovery in Databases (KDD) area. This will show a clear way on how to create information repositories, make user models and extract knowledge from web data. The models and results should be checked by domain experts and, by using their expertise about the business, the cycle is closed. The personalization system will then can use the patterns and knowledge discovered.

4. User Behavior Patterns Extracted from Web Data

In past works,[11,33,49] web mining tools have been applied on data originated in a web site for understanding the user behavior. Clustering techniques have considerably contributed to extracting significant patterns about the user browsing behavior and user text preferences.[50] Before applying web mining techniques, the data are transformed into behavior patterns, using a specific model about the user behavior.

4.1. *User session reconstruction by using the web logs registers*

The process of segmenting the users activities into individual user sessions is called **sessionization**.[10] The sessionization is based on web log registers (see Fig. 3) and the process is not free of errors.[36] It is assumed that a session has a maximum time duration and it is not possible to know if the user pressed the "back" button in the browser. If a page is in the browser cache and the user comes back to it in the same session, that page would not be registered in the web logs. For this, some authors[3,10] have proposed invasive schemes such us sending another application to the browser and capture the exact user browsing. However, this scheme could be easily avoided by the user.

Many authors[3,10,25] have proposed various heuristics to reconstruct sessions from the web logs. In essence, the idea is to create subsets with the users visits and apply mechanisms over them that allow to define a session as a series of events interlaced during a certain period.

The session reconstruction is straightforward with respect to finding the real user sessions, i.e., which pages were visited by a physical human being. In this sense, whatever the chosen strategy to discover real sessions, it must satisfy two essential criteria: the activities performed by a real person can be grouped together, and the activities that belong to the same visit (other objects required for the visited page) also belong to the same group.

#	IP	Id	Acces	Time	Method/URL/Protocol	Status	Bytes	Referer	Agent
1	165.182.168.101	-	-	16/06/2002:16:24:06	GET p1.htm HTTP/1.1	200	3821	out.htm	Mozilla/4.0 (MSIE 5.5; WinNT 5.1)
2	165.182.168.101	-	-	16/06/2002:16:24:10	GET A.gif HTTP/1.1	200	3766	p1.htm	Mozilla/4.0 (MSIE 5.5; WinNT 5.1)
3	165.182.168.101	-	-	16/06/2002:16:24:57	GET B.gif HTTP/1.1	200	2878	p1.htm	Mozilla/4.0 (MSIE 5.5; WinNT 5.1)
4	204.231.180.195	-	-	16/06/2002:16:32:06	GET p3.htm HTTP/1.1	304	0	-	Mozilla/4.0 (MSIE 6.0; Win98)
5	204.231.180.195	-	-	16/06/2002:16:32:20	GET C.gif HTTP/1.1	304	0	-	Mozilla/4.0 (MSIE 6.0; Win98)
6	204.231.180.195	-	-	16/06/2002:16:34:10	GET p1.htm HTTP/1.1	200	3821	p3.htm	Mozilla/4.0 (MSIE 6.0; Win98)
7	204.231.180.195	-	-	16/06/2002:16:34:31	GET A.gif HTTP/1.1	200	3766	p1.htm	Mozilla/4.0 (MSIE 6.0; Win98)
8	204.231.180.195	-	-	16/06/2002:16:34:53	GET B.gif HTTP/1.1	200	2878	p1.htm	Mozilla/4.0 (MSIE 6.0; Win98)
9	204.231.180.195	-	-	16/06/2002:16:38:40	GET p2.htm HTTP/1.1	200	2960	p1.htm	Mozilla/4.0 (MSIE 6.0; Win98)
10	165.182.168.101	-	-	16/06/2002:16:39:02	GET p1.htm HTTP/1.1	200	3821	out.htm	Mozilla/4.0 (MSIE 5.01; WinNT 5.1)
11	165.182.168.101	-	-	16/06/2002:16:39:15	GET A.gif HTTP/1.1	200	3766	p1.htm	Mozilla/4.0 (MSIE 5.01; WinNT 5.1)
12	165.182.168.101	-	-	16/06/2002:16:39:45	GET B.gif HTTP/1.1	200	2878	p1.htm	Mozilla/4.0 (MSIE 5.01; WinNT 5.1)
13	165.182.168.101	-	-	16/06/2002:16:39:58	GET p2.htm HTTP/1.1	200	2960	p1.htm	Mozilla/4.0 (MSIE 5.01; WinNT 5.1)
14	165.182.168.101	-	-	16/06/2002:16:42:03	GET p3.htm HTTP/1.1	200	4036	p2.htm	Mozilla/4.0 (MSIE 5.01; WinNT 5.1)
15	165.182.168.101	-	-	16/06/2002:16:42:07	GET p2.htm HTTP/1.1	200	2960	p1.htm	Mozilla/4.0 (MSIE 5.5; WinNT 5.1)
16	165.182.168.101	-	-	16/06/2002:16:42:08	GET C.gif HTTP/1.1	200	3423	p2.htm	Mozilla/4.0 (MSIE 5.01; WinNT 5.1)
17	204.231.180.195	-	-	16/06/2002:17:34:20	GET p3.htm HTTP/1.1	200	2342	out.htm	Mozilla/4.0 (MSIE 6.0; Win98)
18	204.231.180.195	-	-	16/06/2002:17:34:48	GET C.gif HTTP/1.1	200	3423	p2.htm	Mozilla/4.0 (MSIE 6.0; Win98)
19	204.231.180.195	-	-	16/06/2002:17:35:45	GET p4.htm HTTP/1.1	200	3523	p3.htm	Mozilla/4.0 (MSIE 6.0; Win98)
20	204.231.180.195	-	-	16/06/2002:17:35:56	GET D.gif HTTP/1.1	200	3231	p4.htm	Mozilla/4.0 (MSIE 6.0; Win98)
21	204.231.180.195	-	-	16/06/2002:17:36:06	GET E.gif HTTP/1.1	404	0	p4.htm	Mozilla/4.0 (MSIE 6.0; Win98)

Fig. 3. A typical web log file.

There are several techniques for reconstruction of a real session, which can be grouped in two major strategies: *proactive* and *reactive*.[36]

Proactive strategies aim to identify the user using identification methods like cookies. It consists of a piece of code associated with the web site. When a user visits the site for the first time, a cookie is sent to the browser. Then, when the page is revisited, the browser shows the cookie content to the web server, and an automatic identification takes place. The method has problems from a technical point of view and also with respect to the user's privacy. First, if the site is revisited after several hours, the session will be considered too long; it will actually be a new session. Secondly, some aspects of the cookies seem to be incompatible with the principles of data protection in some countries, like the European Union.[36] Finally, the cookies can be easily detected and deactivated by the user.

Reactive strategies are noninvasive with respect to privacy and they make use of the information contained in the web logs only. They process the registers in order to generate a set of reconstructed sessions, i.e., the set of registers per user.

In the web site analysis, the general scenario is that the web sites usually do not implement identification mechanisms. The utilization of reactive strategies can be more useful. They can be classified into two main groups[5,4,10]:

- Navigation Oriented Heuristics;
- Time Oriented Heuristics.

Navigation Oriented Heuristics assume that the user reaches pages through hyperlinks from others pages. If a page request is unreachable through pages previously visited by the user, a new session is initiated.

Time Oriented Heuristics set a maximum time duration, which is usually 30 minutes for the entire session.[9] Based on this value we can identify the transactions belonging to a specific session by using program filters.

IP	Agent
165.182.168.101	MSIE 5.01
165.182.168.101	MSIE 5.01
165.182.168.101	MSIE 5.01
165.182.168.101	MSIE 5.5
165.182.168.101	MSIE 5.5
165.182.168.101	MSIE 5.5
165.182.168.101	MSIE 5.5
204.231.180.195	MSIE 6.0
204.231.180.195	MSIE 6.0
204.231.180.195	MSIE 6.0
204.231.180.195	MSIE 6.0
204.231.180.195	MSIE 6.0

Date	
.....	16–Jun–02 16:39:02
.....	16–Jun–02 16:39:58
	16–Jun–02 16:42:03
	16–Jun–02 16:24:06
	16–Jun–02 16:26:05
.....	16–Jun–02 16:42:07
	16–Jun–02 16:58:03
.....	16–Jun–02 16:32:06
	16–Jun–02 16:34:10
	16–Jun–02 16:38:40
	16–Jun–02 17:34:20
.....	16–Jun–02 17:35:45

IP	Agent	Date		Sess
165.182.168.101	MSIE 5.01	16–Jun–02	16:39:02	1
165.182.168.101	MSIE 5.01	16–Jun–02	16:39:58	1
165.182.168.101	MSIE 5.01	16–Jun–02	16:42:03	1
165.182.168.101	MSIE 5.5	16–Jun–02	16:24:06	2
165.182.168.101	MSIE 5.5	16–Jun–02	16:26:05	2
165.182.168.101	MSIE 5.5	16–Jun–02	16:42:07	2
204.231.180.195	MSIE 6.0	16–Jun–02	16:32:06	3
204.231.180.195	MSIE 6.0	16–Jun–02	16:34:10	3
204.231.180.195	MSIE 6.0	16–Jun–02	16:38:40	3
204.231.180.195	MSIE 6.0	16–Jun–02	17:34:20	4
204.231.180.195	MSIE 6.0	16–Jun–02	17:35:45	4

Fig. 4. Sessionization process.

A first step in the session reconstruction is to select only the relevant registers, usually those that have a direct relation with the visited pages, and eliminating those that refer to other objects, like pictures, sounds or videos. Only the registers whose status code is not error are considered.

By applying the described procedure to the registers shown in Fig. 3, only a subset of them will go to the next step, as shown in Fig. 4.

The web logs are contained in a stream whose columns are separated by space. Any programming language that can easily process streams, like Perl, C, awk, etc., could be used to group the registers by IP and agents, as shown in the left side of Fig. 4.

The second step is sorting each register group by time stamp. Finally, the registers are selected from a time window of 30 minutes and are grouped together into sessions, as shown in the right hand side of Fig. 4.

A previous and recommended step is to identify abnormal sessions, i.e., registers that do not belong to real human users, but to web robots or spiders. This cleaning step may be implemented by reviewing the agent parameter, since if the user is a robot, the agent usually shows that information. However, if the robot does not identify itself, we will have a firewall situation, i.e., a long session performed by one user. But, in the sessionization process, it is possible to apply a filter to eliminate long sessions.

As a final remark, usually sorting and grouping processes use a big amount of resources and the programming may not be efficient, compared with commercial tools. In this sense, the use of alternative tools, such as relational database engines, that use tables when loading registers and objects like indexes, could accelerate the processes of grouping and sorting.

4.2. *Preprocessing the web site*

The web site is represented by a vector space model.[34] Let R be the number of different words in a web site and Q the number of web pages. A vectorial representation of the web site is a matrix M of dimension RxQ, $M = (m_{ij})$ where $i = 1, \ldots, R$, $j = 1, \ldots, Q$, and m_{ij} is the weight of the ith word in the jth page. To calculate these weights, we use a variant of the *tfxidf-weighting*,

defined as follows[45]:

$$m_{ij} = f_{ij}(1 + sw(i)) * \log\left(\frac{Q}{n_i}\right) \tag{2}$$

where f_{ij} is the number of occurrences of the ith word in the jth page, $sw(i)$ is a factor to increase the importance of special words and n_i is the number of documents containing the ith word. A word is special if it shows special characteristics, e.g. the user searches for this word.

Definition 1 (Page Vector). *It is a vector* $\mathbf{WP^j} = (wp_1^j, \ldots, wp_R^j) = (m_{1j}, \ldots, m_{Rj})$ *with* $j = 1, \ldots, Q$*, that represent a list of words inside a web page.*

This vector represents the jth page by the weights of the words contained in it, i.e., by the jth column of M. The angle's cosine is used as a similarity measure between two page vectors:

$$dp(WP^i, WP^j) = \frac{\sum_{k=1}^{R} wp_k^i wp_k^j}{\sqrt{\sum_{k=1}^{R}(wp_k^i)^2}\sqrt{\sum_{k=1}^{R}(wp_k^j)^2}}. \tag{3}$$

4.3. *Modeling the user browsing behavior*

Our user behavior model uses three variables: the sequence of visited pages, their contents and the time spent on each page. The model is based on a n-dimensional user behavior vector which is defined as follows.

Definition 2 (User Behavior Vector). *It is a vector* $\upsilon = [(p_1, t_1) \ldots (p_n, t_n)]$*, where the pair* (p_i, t_i) *represent the* i*th page visited* (p_i) *and the percentage of time spent on it within a session* (t_i), respectively.

4.4. *Comparing user sessions*

Let α and β be two user behavior vectors of dimension C^α and C^β, respectively. Let $\Gamma(\cdot)$ be a function that returns the navigation sequence corresponding to a user vector. A similarity measure has been proposed elsewhere to compare user sessions as follows[49]:

$$sm(\alpha, \beta) = dG(\Gamma(\alpha), \Gamma(\beta))\frac{1}{\eta}\sum_{k=1}^{\eta} \tau_k * dp(p_{\alpha,k}, p_{\beta,k}) \tag{4}$$

where $\eta = \min\{C^\alpha, C^\beta\}$, and $dp(p_{\alpha,k}, p_{\beta,k})$ is the similarity Eq. (3) between the kth page of vector α and the kth page of vector β. The term $\tau_k = \min\{t_{\alpha,k}/t_{\beta,k}, t_{\beta,k}/t_{\alpha,k}\}$ is an indicator of the user's interest in the visited pages. The term dG is the similarity between sequences of pages visited by two users.[33]

4.5. *Modeling the user's text preferences*

A web site keyword is defined as *a word or a set of words that makes the web page more attractive to the user.*[46] The task here is to identify which are the most

important words (keywords) in a web site from the user's viewpoint. This is done by combining usage information with the web page content and by analyzing the user behavior in the web site.

In order to select the most important pages, it is assumed that the degree of importance is correlated with the percentage of time spent on each page within a session. By sorting the user behavior vector according to the percentage of time spent on each page, the first ι pages will correspond to the ι most important pages.

Definition 3 (ι — Most Important Pages Vector). *It is a vector* $\vartheta_\iota(v) = [(\rho_1, \tau_1), \ldots, (\rho_\iota, \tau_\iota)]$, *where the pair* (ρ_ι, τ_ι) *represents the* ι*th most important page and the percentage of time spent on it within a session.*

Let α and β be two user behavior vectors. A similarity measure between two ι — most important pages vectors is defined as:

$$st(\vartheta_\iota(\alpha), \vartheta_\iota(\beta)) = \frac{1}{\iota} \sum_{k=1}^{\iota} \min \left\{ \frac{\tau_k^\alpha}{\tau_k^\beta}, \frac{\tau_k^\beta}{\tau_k^\alpha} \right\} * dp(\rho_k^\alpha, \rho_k^\beta) \tag{5}$$

where the term $\min\{\cdot, \cdot\}$ indicates the users' interest in the visited pages, and the term *dp* is the similarity measure Eq. (3).

In Eq. (5), the content similarity of the most important pages is multiplied by the ratio of the percentage of time spent on each page by users α and β. This allows us to distinguish between pages with similar contents, but corresponding to different user interests.

4.6. *Applying clustering techniques*

Similar user behaviors are grouped into clusters with common characteristics, such as the navigation sequence or the preferred web pages.

4.6.1. *Clustering the user sessions*

For clustering the user sessions, a Self-organizing Feature Map (SOFM)[21,50] was applied using the similarity measure in Eq. (4). The SOFM requires vectors of the same dimension. Let H be the dimension of the user behavior vector. If a user session has less than H elements, the missing components up to H are filled with zeroes. Otherwise, if the number of elements is greater than H, only the first H components are considered.

The accept/reject criterion was based on whether the page sequence in the cluster centroid was really a feasible sequence, following the current web site hyperlink structure. If affirmative, the cluster was accepted. But if a hyperlink between two consecutive pages in the centroid does not exist, then the pages are substituted by the closest or nearly similar pages in the web site in terms of content. If the situation persists — that is there is no hyperlink between pages — then the cluster is definitively rejected.

4.6.2. *Clustering the ι — most important pages vectors*

A SOFM is used to find groups of similar user sessions. The most important words for each cluster are determined by identifying the cluster centroids. The importance of each word with respect to each cluster is calculated by:

$$kw[i] = \sqrt[\iota]{\prod_{p \in \zeta} m_{ip}} \tag{6}$$

for $i = 1, \ldots, R$, where kw is an array containing the geometric mean of the weights of each word Eq. (2) within the pages contained in a given cluster. Here ζ is the set of pages contained in the cluster. By sorting kw in descending order, the most important words for each cluster can be selected.

The accept/reject criterion is a simple one: if the pages in the cluster centroid have the same main theme, then the cluster is accepted — otherwise it is rejected.

5. Building the Knowledge Base for Storing Web Data

By representing patterns and recommendations as rules may result in generating a large set of rules. Due to frequent changes in the user's interest and the web site itself, the recommendations might become obsolete in a short period of time.

This paper proposes to maintain the patterns by storing them in a database-like repository, and the rules as an independent program that consult the patterns repository when preparing the recommendations. Because the repository will contain patterns discovered in different time periods, it is convenient to apply the data mart architecture. Also, it is necessary to develop generic parametric rules.

5.1. *Overview*

Figure 5 shows the method used for acquiring, maintaining and managing knowledge about web-user behavior.[47] On the left there are three repositories: the Web Information Repository (WIR), the Pattern Repository (PR) and the Rule Repository (RR). The WIR stores the web data to be analyzed while the PR stores the analysis results, and the RR contains domain knowledge from human experts. The two final structures make up the Knowledge Base (KB) about user behavior. This framework allows online navigation recommendations, as well as offline changes to the web site structure and the text contents.

The WIR can be implemented under the data mart architecture by applying the star model. It contains information from web data, for example, user session information (visited pages, time spent, page sequence, etc.) and the web page contents. By construction, the repository stores historical information and allows the direct application of web mining tools at any time. By applying web mining techniques to WIR, it is possible to discover new and hidden knowledge about the user browsing behavior and preferences.[48]

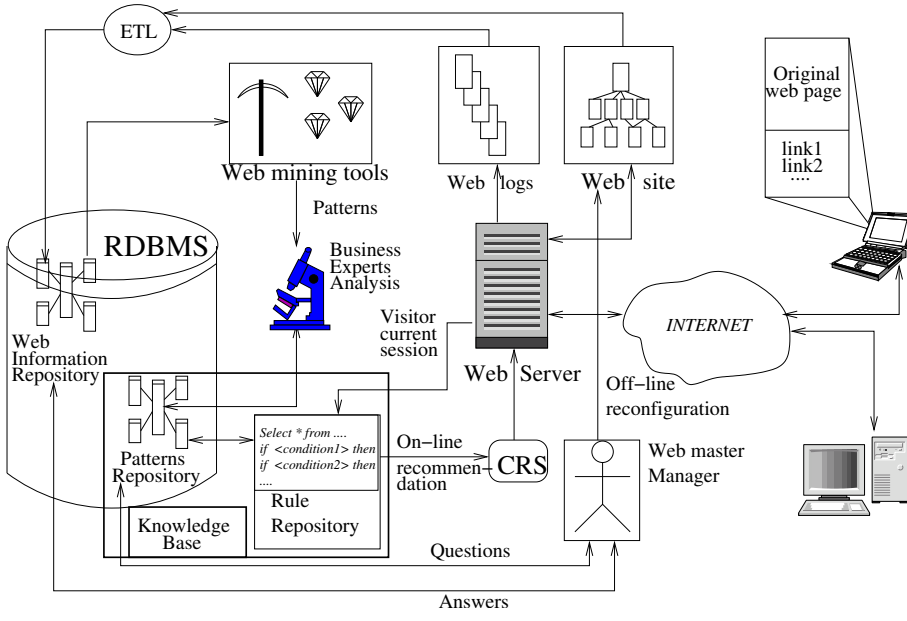

Fig. 5. A framework for implementing a web-based CRS.

As a first step, the behavior patterns extracted by the web mining tools should be validated by a business expert prior to being loaded into the PR. Then the behavior patterns are converted into rules and loaded into the RR. Both PR and RR constitute the KB's complete structure,[8] which are then used to make recommendations. Both repositories hold historical information, so that the impact of future web sites changes can be measured against past changes and used to extrapolate future behaviour patterns.

This procedure allows for two different users - human beings and artificial systems. Human beings consult the KB as a Decision Support System (DSS) and propose changes to and in the web site. These are usually made manually, although some of them can be automated. Artificial systems use the PR and return navigation recommendations as a set of links to web pages. In Fig. 5, the Computerized Recommender System (CRS) creates a dynamic web page containing the online navigation recommendations, received as input by the web server and then sent in turn to users.

In the next subsections, each element of the framework proposed above will be explained in detail.

5.2. *Analyzing and representing the extracted knowledge*

The patterns discovered after applying the web mining tools correspond to the cluster centroids extracted from the user behavior vectors and most important page vectors.

The cluster interpretation is a subjective task.[39,44] While for some persons a cluster may not make much sense, others discover new knowledge in it. That's why it is convenient to be assisted by a business expert for a relatively good interpretation, and see "how to use the patterns found". Because the usage of this new knowledge could derive in several actions, it is convenient to focus on a selected group of them. In our case, we are interested in support recommendations for the web site structure and content modifications.

By analizing the first group of centroids that correspond to the user browsing behavior, two types of actions could be implemented:

- Online recommendations. Given a new user and his/her behavior vector, it is possible to construct an online vectorial representation about the visited pages and the time spent on each page. Then, the user behavior is classified by selecting the nearest centroid using the similarity measure in Eq. (4). From the information contained in the centroid, a prediction about which would be the most interesting pages for the user can be made.
- Offline recommendations. These correspond to structure and content changes proposed by the web master. The centroids are a summary of the typical user behavior, i.e., they show what pages have been searched. A page may be erroneously placed in the web site, making difficult for users to find it. Then, a structural change with respect to this could be suggested.

In order to prepare good recommendations, it is important to take into account the statistics on the web page visits.

By analizing the second group of centroids, corresponding to text preferences, the most significant words for the user are extracted. For the moment, only offline recommendations about the web site content can be made. The keywords can be used as:

- Link words — typical words that have a link to a web page.
- Marked words — words marked with different colors to display the importance of some concepts.
- Searching words — Several search engines, like google, yahoo, altavista, etc., have the option to customize the storage of the web site. The web site owner (or the web master) specifically wants that the search engine crawler rescue the complete web site and index its content, paying a special attention to a set of words. Then, when the user is looking for a specific page that contains some words of his/her interest, the search engine can come up with the web site pages more straightforward.

Any representation of the knowledge described above must consider that:

- Different users have different goals;
- The behavior of a user changes over time;

- A site tends to grow in time by accumulating many pages and links, without being restructured according to new needs.

5.3. *A KB for storing knowledge extracted from web data*

Using his/her expertise, a domain expert can interpret these patterns and build rules for a given task, in our case for online navigation suggestions.

The Knowledge Base[8,44] implements wisdom representation through the use of "if-then-else" rules based on the discovered patterns. Figure 6 shows the structure of the proposed Knowledge Base. It is composed by a Pattern Repository, where the discovered patterns are stored, and a Rule Repository which contains the general rules about how to use the patterns.

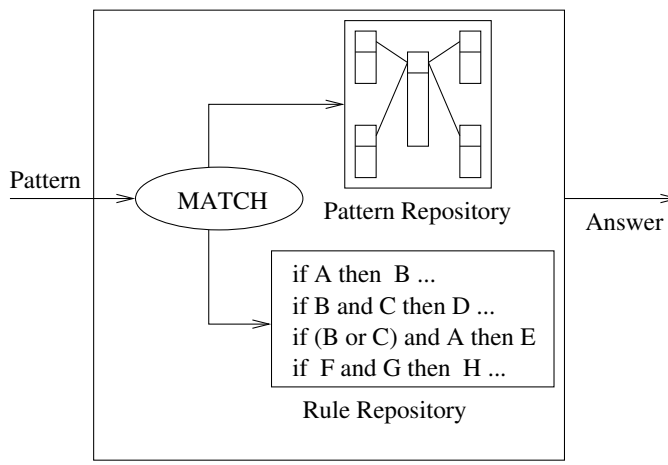

Fig. 6. The conceptual KB structure proposition.

In order to use the KB, when a pattern is presented, a matching process is performed in order to find the most similar pattern in the Pattern Repository. With this information available in the Rule Repository, the set of rules that will create the suggestion of the KB is then selected.

5.3.1. *Pattern Repository*

In the literature, only web access data are stored (see for example Refs. 6, 19, 48 and 44. We propose to store the discovered patterns.

The Pattern Repository stores the patterns revealed from the Information Repository by applying web mining techniques. Figure 7 shows a generic model of the Pattern Repository, which is based on the Data Mart architecture.

The pattern extraction process uses a matching function (column **formula** in table **browsing_behavior**) to find the most similar patterns, within the Pattern

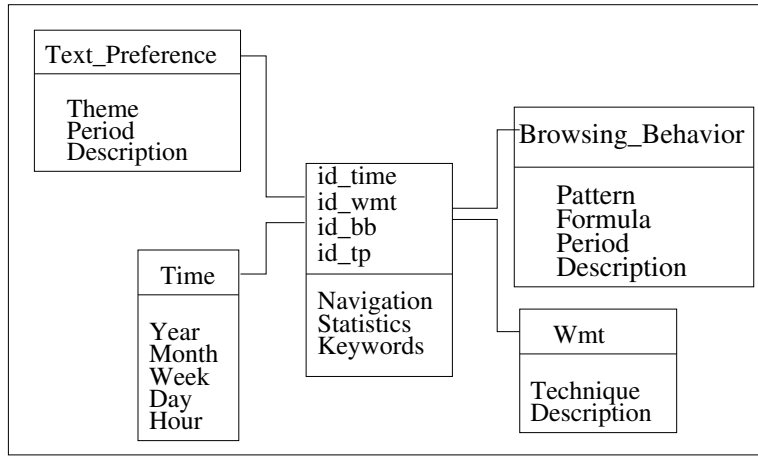

Fig. 7. A star model for implementing the Pattern Repository model.

Repository, to the sample presented to the system. This repository is implemented using the data mart architecture in star model.[18] In the fact table shown in the middle of Fig. 7, the measures are **navigation**, **statistics** and **keywords**. These measures are non-additives[19] and contain the Web page navigation suggestions, related statistics, such as the percentage of visits in the period of study and the keywords discovered. The dimensional table **time** contains the date of application of the Web mining technique over the information repository. The **browsing_behavior** table contains the patterns found about the user browsing behavior. In the *formula* column, the specific expression used for the feature vector comparison is stored, and the *description* column contains the details. The *period* column contains the period of time when the data to be analyzed was generated, e.g. "01-Jan-2003 to 31-Mar-2003". The **text_preferences** table contains, in the *theme* column, a context description of the Web site keywords. For instance, a context for keywords related to credit cards is "promotion about VISA credit card". The table **wmt** (Web mining technique) stores the applied mining technique, e.g., "Self-Organizing Feature Map (SOFM)", "K-means", etc.

When the **KB** is consulted, the Pattern Repository returns a set of possible web pages to be suggested. Based on this set and additional information, such as statistics of the accessed web pages, the Rule Repository makes the final recommendations.

5.3.2. *Rule Repository*

The goal of applying the Rule Repository is to recommend a page from the current web site, i.e. to make a navigation suggestion. Using an online detection mechanism like a cookie, the visited pages and the time spent on them during a session can be obtained.

Using these data, we first match the current visit with the visit patterns stored in the Pattern Repository. This requires a minimum number of visited pages to understand the current user's behavior.

Then the Rule Repository is applied to the matching results to give an online navigation suggestion about the next page to be visited, based on the history of previous navigation patterns.

If the suggested page is not directly connected with the current page, but the suggestion is accepted by the user, then new knowledge can be generated about his/her preferences. This can be used to reconfigure the web site by reorganizing the links among the pages.

```
select navigation, statistics into S
from Pattern_Repository;
...
  if S is empty then
   send("no suggestion");
...
while S not empty loop
  if S.navigation == compare_page(ws,S.navigation) then
    S.navigation ∉ actual_web_site;
...
  if S.navigation ≠ last_page_visited and S.statics > δ then
   send(S.navigation);
...
end loop
```

Fig. 8. A sample of the pseudo-code contained in the Rule Repository.

Figure 8 shows a part of the rule set of Rule Repository. The SQL-query extracts, from the Pattern Repository, the patterns to be used in the creation of the recommendation. In this sense, the function "formula" compares the storage patterns with values originated in the current user session. The parameter ϵ is used to identify those patterns that are "close enough" to the current visit, and the parameter δ filters the recommendations whose statistics are above a given threshold, i.e. it is mandatory that the page acquires a minimum percentage of visits.

Since the Pattern Repository contains historical information, a suggested page may not appear in the current web site. In this case, the function "compare_page" determines the page of the current web site, whose content is most similar to that of the suggested page (by using Eq. (3)).

The data structure "S" is used to store the query results. It consists of two variables: navigation and statistics, that show the navigation sequence and associate statistics. They are used for preparing the recommendation.

6. A Real-World Application

We applied the above described methodology to the web site of the first Chilean virtual bank, where all transactions are made using electronic means, like e-mails, portals, etc. (see www.tbanc.cl). We analyzed all the visits done between January and March 2003. Approximately eight millions of raw web log registers were collected. The site had 217 static web pages with texts written in Spanish, which were numbered from 1 to 217, to facilitate the analysis. In Table 1, the web pages are grouped by their main topic.

6.1. *Using SOFM for browsing pattern discovery*

As mentioned above, the pages in the web site were labelled with a number to facilitate the analysis. Table 1 shows the main content of each page.

The SOFM we used had 6 input neurons and 32*32 output neurons with a thoroidal topology in the feature map.

The cluster identification is performed by using a visualization tool supported by a density cluster matrix, called winner matrix. It contains the number of times the output neurons win, during the training of the SOFM.

By checking the information contained in the winner matrix, eight clusters were identified; however following the accept/reject criteria introduced in Section 4.6.1, only four of them were accepted by the business expert.

Table 2 shows the clusters found by the SOFM. The second column contains the centroid of the cluster, represented by the sequence of visited pages, and the third column indicates the time spent in each centroid.

Table 1. Bank web site pages and their content.

Pages	Content
1	Home page
$2, \ldots, 65$	Products and Services
$66, \ldots, 98$	Agreements with other institutions
$99, \ldots, 115$	Remote services
$116, \ldots, 130$	Credit cards
$131, \ldots, 155$	Promotions
$156, \ldots, 184$	Investments
$185, \ldots, 217$	Different kinds of credits

Table 2. User behavior clusters.

Cluster	Visited Pages	Time Spent in Seconds
1	$(162, 157, 172, 114, 105, 2)$	$(3, 71, 112, 110, 32, 3)$
2	$(1, 5, 7, 10, 135, 191)$	$(30, 61, 160, 110, 175, 31)$
3	$(72, 87, 154, 188, 140, 85)$	$(8, 57, 31, 3, 71, 91)$
4	$(110, 104, 128, 126, 31, 60)$	$(25, 73, 42, 65, 98, 15)$

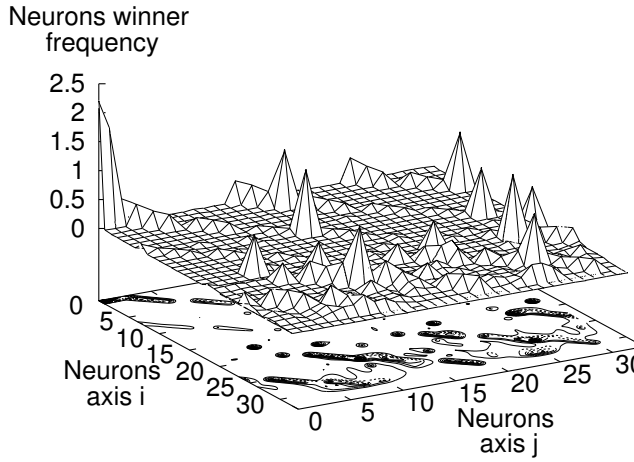

Fig. 9. Identifying clusters of Important Page Vectors.

A simple cluster analysis shows the following results:

- Cluster 1: users that search information about credit cards;
- Cluster 2: users interested in investments and remote services offered by the bank;
- Cluster 3: users interested in agreements between the bank and other institutions;
- Cluster 4: users that are interested in general products and services offered by the bank.

6.2. *Web site keywords*

By assuming that there is a correlation with the maximum time spent per page in a session, a method to find the web site keywords is introduced along with the Important Page Vector definition.

We fixed to 3 the maximum size of this vector. Then, a SOFM with 3 input neurons and 32 output neurons was used to find clusters of Important Page Vectors. The neural network training was carried out on a Pentium IV computer with 1 Gb RAM running under Linux Operating System, distribution Redhat 8.0. The training time was 25 hours and the number of epochs was set to 100.

Figure 9 shows, on the x, y axis, the neurons positions in the SOFM. The z axis is the normalized winning frequency of a neuron in the training set.

Figure 9 shows 12 main clusters which contain the information about the most important web site pages. However, following the criteria introduced in Section 4.6.2, only 8 were accepted by the business expert. The cluster centroid are shown in Table 3. The second column contains the center neurons (winner neuron) of each cluster, representing the most important pages visited.

To get the web site keywords, a final step is required, corresponding to analyzing which words in each cluster have a greater relative importance in the entire web site.

Table 3. Centroids for extraction of keywords.

Cluster	Visited Pages
1	(110, 130, 45)
2	(163, 172, 191)
3	(115, 102, 1)
4	(3, 9, 147)
5	(7, 15, 186)
6	(108, 131, 62)
7	(161, 175, 209)
8	(87, 178, 141)
9	(91, 154, 101)
10	(81, 201, 144)

Table 4. An example of the web site keywords discovered.

#	Keywords	
1	Crédito	Credit
2	Hipotecario	House credit
3	Tarjeta	Credit Card
4	Promoción	Promotion
5	Concurso	Contest
6	Puntos	Points
7	Descuento	Discounts
8	Cuenta	Account

By applying Eq. (6), the keywords and their relative importance in each cluster are obtained. For instance, if the cluster is $\zeta = \{7, 15, 186\}$, then $kw[i] = \sqrt[3]{m_{i7} m_{i15} m_{i186}}$, with $i = 1, \ldots, R$.

Finally, by sorting the kw in descending order, we can select the k most important words for each cluster, for instance $k = 8$.

In Table 4 a selected group of keywords from all clusters is shown. The keywords alone do not make much sense. They need a context in a web page and they could be used as special words, e.g., marked words to emphasize concepts or links to other pages.

The specific recommendation is to use the keywords as "words to write" in a web page, i.e., the paragraphs written in the page should include some keywords, and some of them may be even used as links to other pages.

The keywords could also be used as index words in a search engine, i.e., some of them could be used in the customization of the crawler that visits the web site and load the pages. Then, when a user is looking for a specific page in the search engine, the probability of getting the web site will increase.

6.3. *Loading the knowledge base*

The Knowledge Base presented in Section 5.3 was used to load the patterns and the rules about how to use the patterns in the bank web site. The knowledge stored is mainly used to create online navigation recommendations, but it could be also used to prepare offline recommendations.

6.3.1. *Pattern Repository*

In Fig. 7, the general structure of the pattern repository was presented. The measures in the fact table correspond to the recommended page and to some statistics about its use and the web site keywords for the period under analysis. These patterns are consulted using the information contained in the dimensional tables. An example of this content is the following:

- **Time**. (2003,Oct,4,26,18), i.e. "18:00 hours, October 26th, fourth week, year 2003".
- **Browsing_Behavior**. The cluster centroids discovered by the web mining process and shown in Table 2, as well as with the formula in Eq. (4).
- **Wmt**. Self-Organizing Feature Map with thoroidal architecture, 32x32 neurons.

The table **Time** shows the date when the web mining tools were applied over the Information Repository. In the dimensional table **Browsing_Behavior**, the information about the clusters centroid is displayed. The **Wmt** table contains information about the specific web mining tool applied.

A human user can apply the results of his/her query in the preparation of an offline structural change recommendation. The same query scheme could be used by an automatic system to prepare online navigation recommendations.

6.3.2. *Constructing rules for navigation recommendations*

First, we need to identify the current user session. Since the selected web site uses cookies, this tool can be used for online session identification.

In order to prepare the online recommendation for the $(m+1)$th page to visit, we compare the current session with the patterns in the Pattern Repository. The comparison needs a minimum of three visited pages ($\delta = 3$) to determine the cluster centroid most similar to the current visit and, in this way, allowing to prepare the recommendation for the fourth page to be visited. This process can be repeated after the user has visited more than three pages, i.e., in the recommendations for the fifth page, we use the four pages visited in the current session.

The final online recommendation is made using the developed rule base together with the domain expert. For the four clusters found, sixteen rules were created. We suggest at most three links to follow for the fourth, fifth and sixth pages, i.e. $k = 3$.

6.4. *Building online and offline recommendations*

With the help of a bank's business expert, a list of recommendations was proposed. It includes navigation recommendations, links to be added and/or eliminated from the current site, and words to be used in future pages as content recommendations. Here, only a few recommendations are shown due to a confidentiality agreement with the bank. Some of these recommendations are currently under evaluation at the bank before their final implementation on the web site.

6.4.1. *Structure recommendations*

Based on the clustering of similar visits, we made offline recommendations for the bank web site reconfiguration of the link structure. Some of these recommendations are:

Add links intra clusters. The aim is to improve the accessibility of pages within each cluster from other pages belonging to the same cluster.

Add links inter clusters. The aim is to improve the accessibility of pages belonging to different clusters that share many common users.

Eliminate links. Inter-clusters links that are rarely used can be eliminated.

6.4.2. *Content recommendations*

The web site keywords represent a set of concepts that could motivate the user interest to visit the web site. Their use as isolated words do not make much sense, since a cluster represents different contexts through a set of keywords. Then, for any recommendations is good "to use the word in the paragraphs", i.e., if the page writer wants to write about a certain topic, he needs to include the web site keywords related to that topic.

6.4.3. *Navigation recommendations*

The idea is to use the discovered clusters, the statistics associated to each page and the rules, for creating a correct navigation recommendation. This process needs the online session identification; therefore a mechanism like a cookie must be implemented in all sessions.

We can classify the user browsing behavior into one of the discovered clusters, by comparing the cluster centroid with the current navigation, using the similarity measure introduced in Eq. (4).

The online navigation recommendations are created as follows. Let $\alpha = [(p_1, t_1), \ldots, (p_m, t_m)]$ be the current user session and $C_\alpha = [(p_1^\alpha, t_1^\alpha), \ldots, (p_H^\alpha, t_H^\alpha)]$ the centroid with the maximum $sm(\alpha, C_i)$, where C_i are the centroids discovered and H the dimension of C_i defined in the SOFM. The recommendations are created as a set of pages whose text content is related to p_{m+1}^α. These pages are selected with the expert collaboration.

Let $R_{m+1}(\alpha)$ be the online navigation recommendation for the $(m+1)$th page to be visited by user α, where $\delta < m < H$ and δ the minimum number of pages visited to prepare the suggestion. Then, we can write $R_{m+1}(\alpha) = \{l^{\alpha}_{m+1,0}, \ldots, l^{\alpha}_{m+1,j}, \ldots, l^{\alpha}_{m+1,k}\}$, with $l^{\alpha}_{m+1,j}$ the jth link page suggested for the $(m+1)$th page to be visited by user α, and k the maximum number of pages for each suggestion. In this notation, $l^{\alpha}_{i+1,0}$ represents the "no suggestion" state.

The recommendations are activated when the user clicks the third page in the session. The similarity measure is used to define to which cluster belongs the current user. The suggested pages appear at the bottom of the selected page.

For instance, if a user session matched with cluster "1" (see Table 3), the most likely pages to be recommended are those relative to Products and Services, Promotions and Credit Cards. Using the statistics related to pages and the associated rules, the specific pages to create the recommendations are selected.

6.5. *Testing the recommendation effectiveness*

Usually, the application of any recommendation needs the permission of the web site owner, because any change in the web site may represent a potential risk for the business, as some users may consider the modification a "bad idea". In short, "*sometimes the cure may be worse than the disease*", i.e., the users may not agree with the change or navigation suggestions and prefer other web sites.

For institutions where the web site is the core business, as is the case of the virtual bank, the loss of customers due to modifications in the site is permitted under a narrow-margin and only if it is possible to demonstrate that in a short period of time the changes will attract new customers and retain the exiting ones. In order to estimate the user loss potential, some a priori test can be used. In the next sections, different methods for estimating the loss of users will be introduced.

6.5.1. *Testing offline structure recommendations*

The main idea is to simulate what would the reaction of the user in front of a new web site structure be. In this sense, a secondary web site is created following the structure change recommendations and an usability test is applied to measure the user's reaction. Because we are interested in knowing if the new site structure helps the users to find what they are looking for, the usability test is focused to argue if the recommendations will contribute to get this objective.

Users can be grouped in two classes: amateurs and experienced. The first ones are persons not familiarized with a particular web site and probably with the web technology. Their behavior is characterized by an erratic browsing and, in many cases, they do not find what they are looking for. The second group represents users with experience in the site or other related sites and with the web technology in general. Their behavior is characterized by spending little time in pages with low interest and concentrating on the pages they are looking for, where they spend a significant amount of time.

As amateurs gain experience, they slowly become experienced users. Only experienced users are aware of the features of a particular web site, therefore any recommendations must be based on them.

Usually, a usability experiment considers five persons to be enough for testing the site.[27] In this case, there are two amateur and three experienced users. The first group's profile corresponds to persons without experience in bank web sites or similar contents. The second group profile corresponds to persons with experience in web sites and who have used banking web sites before.

The experiment was developed under the following conditions:

- Each user was asked to search the general information and promotion for three different products (credit card, account, credit, etc), one of them nonexistent in the web site. Then, the users had to write a simple description of three lines with the required information or the "not found" warning in case they did not find what they were looking for.
- Each product information search is considered a new session. It is necessary that the user finishes the current session (using the finish session button).
- The users used the same Internet connection, in this case, a Local Area Network.
- Because the information requested can be obtained in 4 or 5 clicks, the user is considered in a "not found" or "lost in the hyperspace" status when the user visits 6 or more pages, even if the information was ultimately found.

The new web site page content is described in Table 5.

Table 5. New bank web site pages and their content.

Pages	Content
1	Home page
$2, \ldots, 70$	Products and services
$71, \ldots, 105$	Agreements with other institutions
$106, \ldots, 118$	Remote services
$119, \ldots, 146$	Credits cards
$147, \ldots, 155$	Promotions
$156, \ldots, 180$	Investments
$181, \ldots, 195$	Different kinds of credits

The nomenclature used in Tables 6 and 7 is "A" for amateur and"E" for experienced user, respectively. In the column "Find information?", the user's answers are given, and, in some cases, between parenthesis, the real situation when it is different than the user opinion.

The information requested in the question is contained mainly in the following page ranges: 71 to 118 and 147 to 155. From Table 6, we see that user 1 spent a significant time on pages without information related to the question and finally he did not find what he was looking for, although he thought that he did it. The

Table 6. Navigation behavior searching the real web page A.

#	User	Pages Visited	Spent Time	Find Information?
1	A	$(1, 4, 15, 18, 72, 79, \ldots)$	$(3, 50, 8, 30, 4, 5, \ldots)$	Yes(No)
2	A	$(1, 12, 25, 98, 150)$	$(3, 15, 35, 42, 45)$	Yes
3	E	$(1, 73, 83, 152)$	$(2, 55, 71, 10)$	Yes
4	E	$(1, 101, 77, 152)$	$(3, 28, 50, 62)$	Yes
5	E	$(1, 95, 81, 153)$	$(3, 31, 53, 64)$	Yes

Table 7. Navigation behavior searching the real web page B.

#	User	Visited Pages	Spent Time	Find Information?
1	A	$(1, 75, 82, 148, 154)$	$(2, 20, 40, 75, 42)$	Yes
2	A	$(1, 116, 94, 118, 154)$	$(2, 40, 28, 65, 42)$	Yes
3	E	$(1, 75, 108, 147)$	$(2, 50, 41, 67)$	Yes
4	E	$(1, 82, 87, 151)$	$(3, 35, 43, 50)$	Yes
5	E	$(1, 84, 149, 150)$	$(3, 49, 45, 62)$	Yes

Table 8. Navigation behavior searching a non-existing web page.

#	User	Visited Pages	Spent Time	Find Information?
1	A	$(1, 72, 75, 84, 151, 152, 87, \ldots)$	$(2, 10, 11, 14, 20, 25, \ldots)$	Yes (No)
2	A	$(1, 110, 78, 150, 146, 155)$	$(2, 6, 10, 8, 9, 2)$	No
3	E	$(1, 104, 105, 76)$	$(2, 6, 5, 4)$	No
4	E	$(1, 95, 153)$	$(3, 6, 8)$	No
5	E	$(1, 102, 147)$	$(2, 4, 4)$	No

another amateur user found the information, but he had to visit five pages and he spent a significant time on pages with irrelevant information for the purpose of searching.

In the case of experienced users, they found the information looked for.

In Table 7, the users, in general, acquired experience and all of them were able to find the requested information. It is interesting to note that the users tend to spend a significant time on pages whose content is related to the searching purpose.

Finally, Table 8 shows the result of "searching a non existing product". The experienced user very quickly noted that the relative information did not appear in the site. However, the amateur user tried to find it, looking in the site and being confused in his answers.

From the results of this experiment, we can conclude that the users are able to find the information that they are looking for in a reduced number of visits, which is an objective of the bank web site.

The second experiment is a simple questionnaire for the same group of users, about the impression of the web site structure. The questions aim to understand in which grade the new site structure contribute to helping the users in their

Table 9. Usability test for the web site hyperlink structure.

#	Question	Acceptability Opinion				
		Totally Opposite	Moderately Opposite	Some Agree Some Opposite	Moderately Agree	Totally Agree
1	Does the site structure allows to find what is looking for?		1	1	3	
2	Does the site provide a consistent navigation?			1	4	
3	Does the site hide information?	2	2	1		
4	Is the site structure easy to understand?			1	3	1

information search tasks. Table 9 shows the questionnaire results. The majority of the asked users agreed with the improvement of the web site hyperlink structure for searching information purposes.

By using the usability test, we can get an approximation about the effectiveness of the recommendations about changes in the web site structure. However, the real test will be when the the new web site version is released, with the users visiting it and visualizing the structural changes. Because applying all the changes could cause a "lost in the hyper-space" feeling for the users, it is also recommended to implement the modifications gradually and, at the same time, to review the users' reaction.

6.5.2. *Testing offline content recommendations*

Essentially, isolated web site keywords do not have much sense. Therefore, it is necessary to put them in a context, for instance in a text created for a particular theme by including the related keywords. More exactly, for testing the web site keywords effectiveness, i.e., the capacity to attract the users attention during their visit to a page, a textual fragment, such us a paragraph, should be created. However, elements such as the text style and the nature of the contained information can also attract the user's attention. Holding up the web site keywords effectiveness analyzed, it was decided to use texts belonging to the web site content under study. These texts are one of the data sources used in the the web site keyword identification and the style and information were intrinsically contained when the keywords were extracted.

From the entire texts in the site, five paragraphs were selected: two of them containing a major number of web site keywords in the site, and the remaining

Table 10. Testing the web site keyword effectiveness.

#	Including Web Site keywords?	Acceptability Opinion				
		Irrelevant	Moderately Relevant	Some Information	Moderately Irrelevant	Relevant
1	Yes				3	2
2	Yes			1	2	2
3	No	2	2	1		
4	No		3	2		
5	No		4	1		

ones were extracted randomly. All these paragraphs were shown to the same group of amateur and expert users introduced in the previous section.

Table 10 shows the result of applying the effectiveness test for the web site keywords. For the paragraphs that contain keywords, the users show a good receptivity, considering them interesting and with relevant information, which allow us to extrapolate that the words used had important information from the user's point of view, i.e., these words made the paragraphs more attractive.

Web site keywords provide the web site designer with some general lines about the specific text content. Of course, the utilization of the keywords does not guarantee the success of the paragraph. The combination with other elements, like the semantic content, style and the paragraph meaning are also important for transmitting the right message to the users.

6.5.3. *Testing online navigation recommendations*

In order to test the effectiveness of the recommendation,[43] a method based on the same web data used in the pattern discovery stage is proposed. A part of the complete web data is used to extract significant patterns, and for these we define a set of rules on how to use them. Then, we test the effectiveness with the remaining part of the web data.

Let $ws = \{p_1, \ldots, p_n\}$ be the web site and the pages that compose it. By using the distance introduced in Eq. (3) and with the collaboration of a web site content expert, we can define an equivalence class for pages, where pages belonging to the same class contain similar information. The classes partition the web site in disjoint subsets of pages.

Let Cl_x be the xth equivalent class for the web site. It is such as $\forall\, p_z \in Cl_x, p_z \notin Cl_y,\ x \neq y \bigcup_{x=1}^{w} Cl_x = ws$ where w is the number of equivalence classes. Let $\alpha = [(p_1, t_1), \ldots, (p_H, t_H)]$ be a user behavior vector from the test set. Based on the first m pages actually visited, the proposed system recommends for the following page $(m+1)$ several alternatives, i.e., possible pages to be visited.

We test the effectiveness of the recommendations made for the $(m+1)$th page to be visited by user α following the procedure described next. Let Cl_q be the

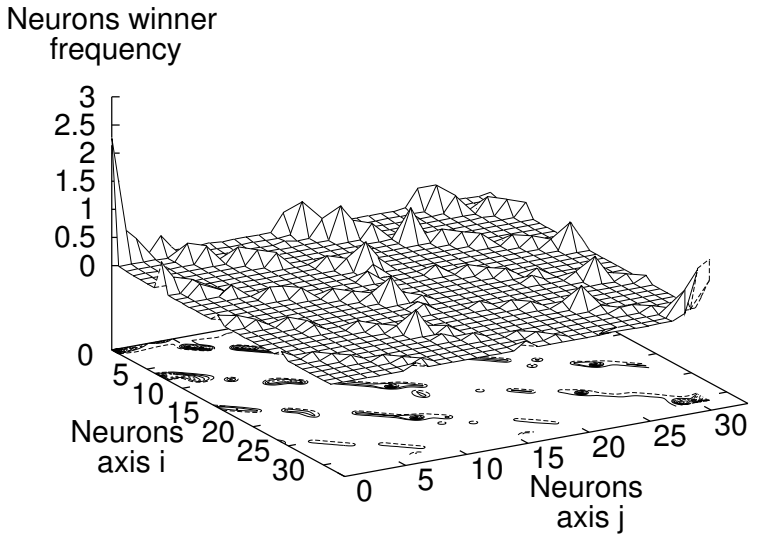

Fig. 10. Clusters of user behavior vectors using 70% of the data.

equivalence class for p_{m+1}. If $\exists l^{\alpha}_{m+1,j} \in R^{\alpha}_{m+1}/l^{\alpha}_{m+1,j} \in Cl_q,\ j > 0$ then we assume the recommendations were successful.

By construction of the recommendation, the set of link pages to be included could be large, which may confuse the user with respect to which page to follow next. We set in k the maximum number of pages per recommendation. Using the page distance introduced in Eq. (3), we can extract the closest k pages to p_{m+1} in the recommendation.

$$E^{k}_{m+1}(\alpha) = \{l^{\alpha}_{m+1,j} \in sort_k(sp(p_{m+1}, l^{\alpha}_{m+1,j}))\}\,, \tag{7}$$

with sp the page distance introduced in Eq. (3). The "$sort_k$" function sorts the result of sp in descendent order and extracts the "k" link pages with the largest distance to p_{m+1}. A particular case is when $E_{m+1}(\alpha) = \{l^{\alpha}_{m+1,0}\}$, i.e., no recommendation is proposed.

The above methodology was applied to the data originated in the bank web site. The recommendations are activated when the users click the third page in their sessions. From the 30% of the user behavior vectors that belong to the test set, only those with six real components are selected, i.e., it was not necessary to complete user vectors with zeros to get the six components. Given this selection, we have obtained 11,532 vectors to test the effectiveness of the online navigation suggestions.

Figure 10 shows the clusters identified using 70% of the data. The centroid are presented in more detail in Table 11.

It is interesting to notice the similarity between the clusters identified using the complete set of data and those using 70% of data. In essence, the pages contained in each centroid are similar.

Table 11. User behavior clusters using 70% of the data.

Cluster	Visited Pages	Time Spent (in Seconds)
1	(2, 11, 25, 33, 136, 205)	(10, 50, 120, 130, 150, 58)
2	(120, 117, 128, 126, 40, 62)	(30, 41, 52, 68, 101, 18)
3	(81, 86, 148, 190, 147, 83)	(8, 55, 40, 11, 72, 101)
4	(161, 172, 180, 99, 108, 1)	(8, 71, 115, 97, 35, 9)

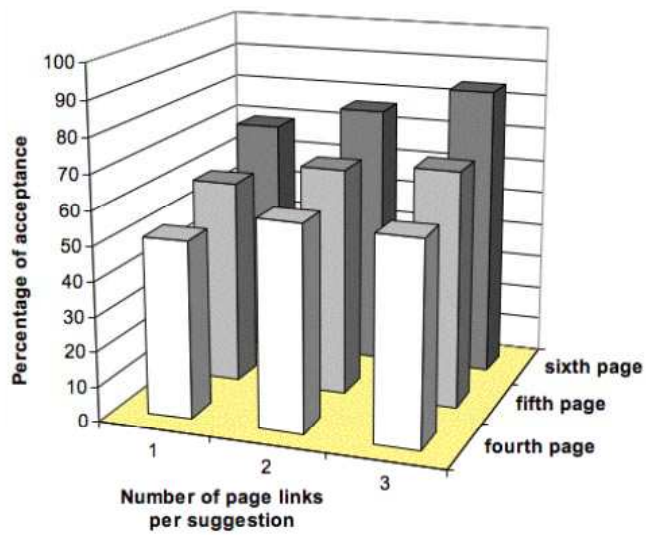

Fig. 11. Percentage of accepting the online navigation recommendations.

Figure 11 shows a histogram representing the percentage of the accepted suggestions using our validation method. As can be seen, acceptance increases if more pages are suggested for each page visit.

If using the proposed methodology just one page is suggested, it is accepted in slightly more than 50% of the cases. This has been considered a very successful suggestion by the business expert, since we are dealing with a complex web site with many pages, many links between pages, and a high rate of users that leave the site after few clicks.

Furthermore, it should be mentioned that the percentage of acceptance would have been even higher if we had actually suggested the respective page during the session. Since we are comparing past visits stored in log files, we could only analyze the behavior of users that did not actually receive any suggestion we proposed.

7. Conclusions

This paper introduced a methodology for creating a Knowledge Base (KB) for implementing a web-based computerized recommendation system.

The KB is a complex structure that stores the discovered patterns and the rules about how to use the patterns. They are maintained separately in a Pattern Repository and a Rule Repository, respectively. As web sites and user behaviors vary greatly, the classical rule base representation (by using rigid rules) generates too many rules, complicating the rules' storage. A set of parametric rules solve for this problem and that follows the practice of consulting the Pattern Repository to prepare recommendations.

These can be classified as offline and online recommendations. Offline recommendations are performed manually and consist in changes in the web site structure and content. Here, the Pattern Repository and the WIR are consulted to create an appropriate recommendation.

Online recommendations are mainly navigation recommendations to the user, i.e., links to pages to be visited. They are provided by an automatic computerized recommender system that interacts with the KB and returns a HTML output file with the navigation recommendations. This is the input file to the web server that sends the page to the users.

Because the accept/rejection of the recommendation by the user results in new knowledge, which can be used for improving future recommendations, in our future work we intend to expand the current KB structure for storing this new knowledge.

Acknowledgement

The authors thank to the Millennium Scientific Nucleus on Complex Engineering Systems, Chile, which has partially funded this work.

References

1. G. Adomavicius and A. Tuzhilin, Personalization technologies: A process-oriented perspective. *Communications of ACM*, 48(10):83–90, October 2005.
2. G. Adomavicius and A. Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, June 2005.
3. B. Berendt, A. Hotho and G. Stumme, Towards semantic web mining. In *Proc. in First Int. Semantic Web Conference*, pp. 264–278, 2002.
4. B. Berendt, B. Mobasher and M. Spiliopoulou, Web usage mining for e-business applications. Tutorial, ECMML/PKDD Conference, August 2002.
5. B. Berendt and M. Spiliopoulou, Analysis of navigation behavior in web sites integrating multiple information systems. *The VLDB Journal*, 9:56–75, 2001.
6. F. Bonchi, F. Giannotti, C. Gozzi, G. Manco, M. Nanni, D. Pedreschi, C. Renso and S. Ruggieri, Web log data warehousing and mining for intelligent web caching. *Data and Knowledge Engineering*, 32(2):165–189, 2001.
7. A. Bonifati, F. Cattaneo, S. Ceri, A.Fuggetta and S.Paraboschi, Designing data marts for data warehouses. *ACM Transactions on Software Engineering Methodology*, 4:452–483, 2001.
8. M. Cadoli and F. M. Donini, A survey on knowledge compilation. *AI Communications*, 10(3-4):137–150, 1997.

9. L. D. Catledge and J. E. Pitkow, Characterizing browsing behaviors on the world wide web. *Computers Networks and ISDN System*, 27:1065–1073, 1995.
10. R. Cooley, B. Mobasher and J. Srivastava, Data preparation for mining world wide web browsing patterns. *Journal of Knowlegde and Information Systems*, 1:5–32, 1999.
11. R. W. Cooley, *Web Usage Mining: Discovery and Application of Interesting Patterns from Web Data*. PhD thesis, University of Minnesota, Minnesota, USA, 2000.
12. N. Correia and M. Boavida, Towards an integrated personalization framework: A taxonomy and work proposals. In *Workshop on Personalization Techniques in Electronic Publishing on the Web: Trends and Perspectives*, May 2002.
13. L. F. Cranor, 'i didn't buy it for myself': Privacy and ecommerce personalization. In *In Procs. of the Second ACM Workshop on Privacy in the Electronic Society*, pp. 111–117, New York, USA, 2003.
14. J. Debenham, Knowledge base maintenance through knowledge representation. In *Procs. 12th Int. Conf. on Database and Expert Systems Applications*, pp. 599–608, München, Germany, September 2001.
15. M. Deshpande and G. Karypis, Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems*, 22(1):143–177, 2004.
16. M. Eirinaki and M. Vazirgannis, Web mining for web personalization. *ACM Transactions on Internet Technology*, 3(1):1–27, 2003.
17. D. Gefen, Customer loyalty in e-commerce. *Journal of the Association for Information Systems*, 3:27–51, 2002.
18. W. H. Inmon, *Building the data warehouse (2nd ed.)*. John Wiley and Sons, New York, 1996.
19. R. Kimball and R. Merx, *The Data Webhouse Toolkit*. Wiley Computer Publisher, New York, 2000.
20. A. Kobsa, Tailoring privacy to users' needs. In *In Procs. of the 8th International Conference in User Modeling*, pp. 303–313, 2001.
21. T. Kohonen, *Self-Organization and Associative Memory*. Springer-Verlag, 1987.
22. R. Kosala and H. Blockeel, Web mining research: A survey. *SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery & Data Mining*, 2(1):1–15, 2000.
23. Z. Lu, Y. Yao and N. Zhong, *Web Intelligence*. Springer-Verlag, Berlin, 2003.
24. B. Mobasher, B. Berendt and M. Spiliopoulou, Kdd for personalization. Tutorial, KDD Conference, September 2001.
25. B. Mobasher, R. Cooley and J. Srivastava, Creating adaptive web sites through usage-based clustering of urls. In *Procs. Int Conf IEEE Knowledge and Data Engineering Exchange*, November 1999.
26. M. Mulvenna, S. Anand and A. Buchner, Personalization on the net using web mining. *Communication of ACM*, 43(8):123–125, August 2000.
27. J. Nielsen, Quantitative studies: How many users to test? Report, Usable Information Technology, 2006. Also available as `http://www.useit.com/alertbox/quantitative_testing.html`.
28. S. K. Pal, V. Talwar and P. Mitra, Web mining in soft computing framework: Relevance, state of the art and future directions. *IEEE Transactions on Neural Networks*, 13(5):1163–1177, September 2002.
29. M. Perkowitz and O. Etzioni, Towards adaptive web sites: Conceptual framework and case study. *Artificial Intelligence*, 118(1-2):245–275, April 2000.
30. B. J. Pine, *Mass Customization*. Harvard Business School Press, Boston, USA, 1993.
31. P. Resnick and H. R. Varian, Recommender systems. *Communications of the ACM*, 40(3):56–58, March 1997.

32. S. Ríos, J. Velásquez, E. Vera and H. Yasuda, Improving the web text content by extracting significant pages into a web site. In *In Procs. 5th IEEE Int. Conf. on Intelligent Systems Design and Applications*, pp. 32–36, September 2005.
33. T. A. Runkler and J. Bezdek, Web mining with relational clustering. *International Journal of Approximate Reasoning*, 32(2-3):217–236, Feb 2003.
34. G. Salton and C. Buckley, Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, Feb 1988.
35. J. B. Schafer, J. A. Konstan and J. Riedl, E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, 5:115–153, 2001.
36. M. Spiliopoulou, B. Mobasher, B. Berendt and M. Nakagawa, A framework for the evaluation of session reconstruction heuristics in web-usage analyis. *INFORMS Journal on Computing*, 15:171–190, 2003.
37. J. Srivastava, R. Cooley, M. Deshpande and P. Tan, Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1(2):12–23, 2000.
38. L. Terveen, W. Hill, B. Amento, D. McDonald and J. Creter, Phoaks: A system for sharing recommendations. *Communications of the ACM*, 40(3):59–62, 1997.
39. S. Theodoridis and K. Koutroumbas, *Pattern Recognition*. Academic Press, 1999.
40. C. Vassiliou, D. Stamoulis and D. Martakos, The process of personalizing web content: techniques, workflow and evaluation. In *Procs Int. Conf. on Advances in Infrastructure for Electronic Business, Science and Education on the Internet*, 2002.
41. J. D. Velásquez, *A Study on Intelligent Web Sites: Towards the New Portals Generation, Dissertation for degree of Doctor of Philosophy*. University of Tokyo, Japan, 2004.
42. J. D. Velásquez, P. Estévez, H.Yasuda, T. Aoki and E. Vera, Intelligent web site: Understanding the visitor behavior. *Lecture Notes in Computer Science*, 3213(1):140–147, September 2004.
43. J. D. Velásquez and V. Palade, Testing online navigation recommendations in a web site. In B. Gabrys, R. J. Howlett, and L. C. Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems*, volume 4253 of *LNAI*, pp. 487–496. Springer, 2006.
44. J. D. Velásquez and V. Palade, A knowledge base for the maintenance of knowledge extracted from web data. *Journal of Knowledge Based Systems (Elsevier)*, page to appear, 2007.
45. J. D. Velásquez, H. Y. S. Ríos, A. Bassi and T. Aoki, Towards the identification of keywords in the web site text content: A methodological approach. *International Journal of Web Information Systems*, 1(1):11–15, March 2005.
46. J. D. Velásquez, R. Weber, H. Yasuda and T. Aoki, A methodology to find web site keywords. In *Procs. IEEE Int. Conf. on e-Technology, e-Commerce and e-Service*, pp. 285–292, Taipei, Taiwan, March 2004.
47. J. D. Velásquez, R. Weber, H. Yasuda and T. Aoki, Acquisition and maintenance of knowledge for web site online navigation suggestions. *IEICE Transactions on Information and Systems*, E88-D(5):993–1003, May 2005.
48. J. D. Velásquez, H. Yasuda, T. Aoki and R. Weber, A generic data mart architecture to support web mining. In *Procs. 4th Int. Conf. on Data Mining*, pp. 389–399, Río de Janeiro, Brazil, December 2003.
49. J. D. Velásquez, H. Yasuda, T. Aoki and R. Weber, A new similarity measure to understand visitor behavior in a web site. *IEICE Transactions on Information and Systems, Special Issues in Information Processing Technology for web utilization*, E87-D(2):389–396, February 2004.

50. J. D. Velásquez, H. Yasuda, T. Aoki, R. Weber and E. Vera, Using self organizing feature maps to acquire knowledge about visitor behavior in a web site. *Lecture Notes in Artificial Intelligence*, 2773(1):951–958, September 2003.
51. T. Zhang and V. S. Iyengar, Recommender systems using linear classifiers. *J. Mach. Learn. Res.*, 2:313–334, 2002.