WWW Assisted Browsing by Reusing Past Navigations of a Group of Users

Michel Jaczynski and Brigitte Trousse

INRIA Sophia Antipolis, AID Research Group, 2004 routes des Lucioles,
06902 Sophia Antipolis, France
{Michel.Jaczynski, Brigitte.Trousse}@inria.sophia.fr

Abstract. In this paper, we present our case-based browsing advisor for the Web, called BROADWAY. BROADWAY follows a group of users during their navigations and supports an indirect collaboration to recommend Web pages to visit next. BROADWAY uses case-based reasoning to reuse precise experiences extracted from past navigations with a time-extended situation assessment, i.e. the recommendations are based mainly on the similarity of ordered sequences of past accessed documents. A first experimental evaluation shows that the system improves the information searching task.

1 Introduction

This article focuses on the assistance given to the user during a browsing session on the World Wide Web (Web for short). The Web is a huge hypermedia of heterogeneous and dynamic documents [2] and two main kinds of methods can be used to locate a relevant document through this space: *querying* and *browsing*. Querying is done through a search engine (such as Altavista) and is appropriate when the user has a clear goal which should usually be expressed through a list of keywords. However, even if querying is done in a first step, browsing is necessary to cope with the amount of noisy data retrieved by querying. Browsing is also needed if one wants to use a bookmark or a recommendation given by another user. Thus, browsing is a usual mean for searching information on the Web but its huge size and its lack of structure and homogeneity may disorient the user.

Our approach is based on the design of a *browsing advisor* that follows the user during a browsing session to infer his goal, and then may propose some recommendations of potentially relevant documents to visit next. In order to improve the relevance of recommended documents and to enable a wide use of the system, we want to address this problem with four specific requirements:
- the browsing advisor must be *designed for the Web*: its use must not be restricted to a localized site, and it must take the dynamic aspect of Web documents into account,
- the browsing advisor must *learn from a group of users* how to improve the recommendation computation,

- the recommendation computation must take the *time-extended browsing situation* into account,
- the browsing advisor must be *independent* of the user's browser software.

A time-extended situation represents not only the current state of the observed navigation but also its past sequence of events. We claim as others [5,6] that a particular state of the navigation (current document and/or an *instantaneous* description) is not sufficient to compute relevant recommendations. In order to better describe the user's *implicit intent* during browsing, we want to consider past visited documents and their access order.

Other works have proposed browsing advisors such as Letizia [12], WebWatcher [1], Hospitext [6] and others [5,13,18] but they do not satisfy our requirements (a comparative analysis with our approach is given in [10]). Thus, we propose a browsing advisor, named BROADWAY[1][10]. BROADWAY uses Case-Based Reasoning (CBR) to learn from users' navigations the set of relevant cases which can be reused to improve and to keep updated the recommendation process. The use of CBR is based on the following hypothesis: *if two users went through a similar sequence of similar documents, they might have similar browsing intent, so that we can recommend to one user the documents evaluated as relevant by the other one.*

This paper describes BROADWAY and the first results from its experimental evaluation. In section 2, we give an overview of BROADWAY and we introduce the graphical user interface and the underlying distributed architecture. Then, in section 3, we present in more details the recommendation server that embeds a case-based reasoner. Finally, section 4 gives the results of the experiment made with real users in order to evaluate the assistance given by BROADWAY.

2 Overview of BROADWAY

BROADWAY is a browsing advisor that can be used by a group of users who want to share their browsing experiences on the Web because they have similar topics of interest or they are working in the same team. Each user can use its own browser and may communicate with BROADWAY through a specific graphical interface. Behind this interface, BROADWAY uses a distributed architecture for better efficiency and extendibility.


2.1 User Interface

The user is able to communicate with BROADWAY by two means: the *toolbars* and the *manager*. A toolbar is inserted in each page when a navigation has been started (cf. Fig. 1) and provides a set of contextual actions: evaluation of the page with four levels, consultation of the current navigation, management of textual annotations of

---

[1] BROADWAY : a BROwsing ADvisor reusing pathWAYs.

the page. This interface allows the user to navigate with multiple windows (or frames) and different toolbars are then displayed. The evaluation of each page is not a feedback of recommendations given by BROADWAY, but a comment the user gives on its own navigation as if he was advising another user. We allow users to write and read annotations related to each page so that textual explanation of the evaluation can be attached and even discussions on a document may start. Pages with annotations will be preferred by BROADWAY in the recommendation computation.
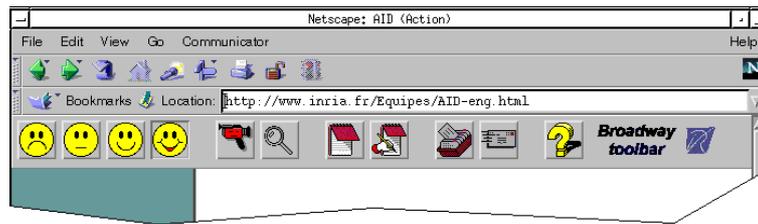
Fig. 1. The toolbar is inserted at the top of each page and can be hidden or displayed
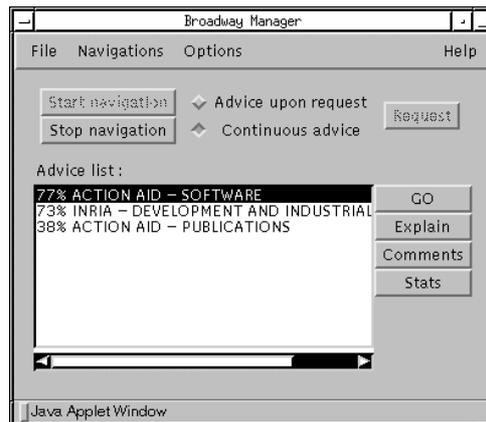
Fig. 2. The manager is used to control the recommendation process (navigation, modes). The user can also select one of the recommended pages and click on the *go* button to display it

The BROADWAY manager is displayed in a separate small window and is used to configure the system, to control the navigation (a navigation can be started or stopped) and to display the recommendations (cf. Fig. 2). Recommendations are presented through a list of URLs with a percentage rate representing an estimated relevance; the user can select one page to display directly. For each user, BROADWAY operates in one of the two following modes: *recommendations upon request* or *continuous recommendations*. In the first mode, the user may ask for help and the reasoning will start to update the recommendation list. In the second mode, the list of recommended pages in the manager will be updated automatically after each navigation move.

2.2 Distributed Architecture

On the server side, BROADWAY is composed of different processes (cf. Fig. 3): the HTTP proxy, the event dispatcher, the page information server, the user information server, the annotation server and the recommendation server. On the client side, different applets (manager, toolbars and watchers) are running in the browser of each user. All these modules are written in the object-oriented Java™ programming language and use ObjectStore PSE[2] for persistency. At the process communication level between the applets and the servers, we use standard Java Remote Method Invocation (RMI) and HTTP (HyperText Transport Protocol).
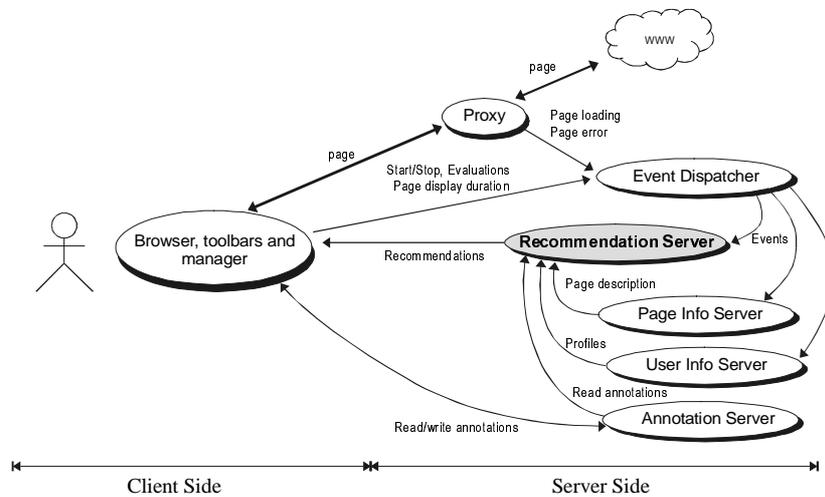

Fig. 3. BROADWAY detailed architecture

BROADWAY accepts multiple simultaneous connections and each user is identified by a user name and a password based on the standard proxy authentication mechanism defined in the HTTP protocol. When a user opens his/her browser, a pop-up window is automatically displayed, asking for his/her Id and password. This authentication is important because it guarantees private access to the system which manages navigational data and user profiles. It is also used to follow a user through his/her different moves through the Web.

BROADWAY integrates an extended version of the Jigsaw[3] proxy server developed by the W3C (Web Consortium). The proxy intercepts all HTTP replies containing an HTML document: all the HTTP headers and the content of the documents are available to be processed by the page information server. In addition, BROADWAY provides caching by itself (by using Jigsaw caching) which is more effective than browser caching because the cache is shared by several users. This proxy-based

---

[2] http ://www.odi.com/products/pse/psej.html

[3] http://www.w3c.org/Jigsaw/

architecture is well suited to analyze the HTTP communications and the document content in a transparent way [3]. The proxy forwards the requests and alters the replies by inserting dynamically two Java applets in each HTML page: the watcher and the toolbar. Watchers will send events when the pages are displayed, even when pages are loaded from a personal cache memory [17].

The event dispatcher collects browsing events from users: page loading and errors, page content, display duration, navigation start and stop, page evaluation, etc. These events are then transmitted to the other appropriate modules. The page information server analyzes the content of pages to extract mainly the titles, the headers and the keywords. The user information server stores the users' Ids and profiles (name, e-mail, password). The annotation server is used to retain textual annotations linked to Web pages. Finally, the recommendation server computes the recommendations using case-based reasoning. This distributed architecture is the result of the specialization and the combination of a set of tools, called BROADWAY*Tools, that facilitates the development and the maintenance of this kind of architecture.

3   The Recommendation Server

While BROADWAY is accessed by multiple users, the recommendation server keeps tracks of the user's behaviors, and manages the cases that represent useful experiences extracted from the navigations. When a recommendation must be given to a user, the recommendation server starts a new reasoning.

3.1 Representation of Cases

The case representation is based on two levels: raw data and relevant knowledge. The raw data are stored in navigations and represented by time series. A useful piece of knowledge is then identified by a case which references a precise instant in a navigation through a *time-extended situation*.

Recording the Navigations of Users. The recommendation server receives different events from the actions of users and summarizes their behavior according to a set of variables that evolve over time for each user. Four variables have been defined:

1. URL of the document,
2. document content description (stems of extracted keywords),
3. user's evaluation of the document,
4. display time ratio: time spent displaying a document relatively to its size.

The evolution of each variable is represented by a time series. These time series are *sampled* since the unit of the chosen model of time is a change of pages. A *navigation* groups together the time series of a user for a given time interval. We associate to each navigation a *context* containing a synthetic description of the pages accessed during

the navigation: the most used keywords in page contents, and the hosts accessed. Fig. 4 gives an example of a navigation.

Fig. 4. Extract of a navigation recorded by BROADWAY that can be displayed to the user. This is an example of a navigation where the user intent was to find more information about BROADWAY; so the user navigated through the INRIA server, found the research team AID and finally found one relevant page about this software

Potential and Concrete Cases. In case-based reasoning, a case is a useful past experience and the *situation* of a case (a set of indices) defines when its use is relevant. In the recommendation server, a case references a precise experience inside a navigation and is composed of:
- a time-extended situation,
- a list of pages which can be recommended in that situation,
- a set of data used to manage the case (date of creation and user's id).

Only few works in CBR deal with situations based on time series, and we make in our work a clear distinction between case indexing techniques based on an *instantaneous situation*, a set of indices giving the relevant state of the world at a particular instant, and a *time-extended situation*, a set of indices describing mainly the evolution of this state. Few existing applications in case-based reasoning have tried to

represent and use behavioral situations inside cases (robot control [14], plant nutrition control [8], Web navigation [5,6], cf. [9] for a detailed analysis). To cope with this kind of situations, we have designed and implemented a framework for case-based reasoning where cases are indexed by a time-extended situation [8,11].

Based on this framework, the current representation of indices used in Broadway is as follows. A time-extended situation has an *instantaneous part* and a *behavioral part*. The instantaneous part contains the navigation context (keywords and hosts) which is shared by all the cases based on the same navigation. The behavioral part is composed of (cf. Fig. 5):

- − The last three pages and their keywords. This selection is used to describe a precise step in the user's navigation;
- − A set of past selected relevant pages using the four variables. The relevance of each page is based on an explicit feature (evaluation done by a user) and an implicit feature (high display time ratio). Combination of explicit and implicit features generally improves the selection accuracy [16];
- − A set of *before* constraints which are used to order selected pages.
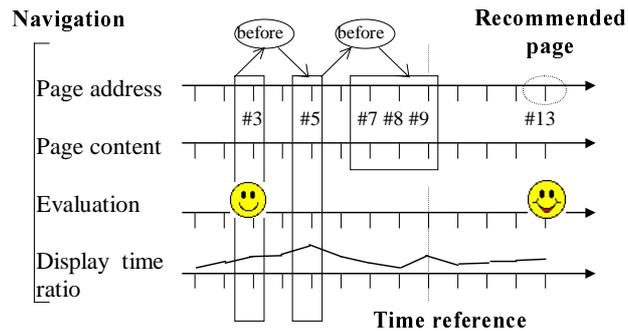
Fig. 5. This figure shows an example of the association of the behavioral part of a case and the recommended pages. This example is based on the navigation of the Fig 4. The page #3 is selected because of its evaluation, and the page #5 is selected because of its high display time ratio (above 20). The page #13 is recommended by this case because it has been evaluated as relevant by the user.

Two kinds of cases have been defined: *concrete cases* and *potential cases*. A concrete case has an explicit representation and a potential case is defined through a template. A potential case may become concrete when its template is instantiated at a precise date inside a navigation.

3.2 The reasoning process

First we present an overview of the reasoning process and then we give more details on the similarity measures used in the retrieval step.

Overview of the Reasoning Process. Cases are stored in a memory and indexed by their time-extended situations. The computation of recommendations uses a reasoning in three steps and takes the current navigation of a user as target case.

The retrieve step. This step retrieves from the memory at most k cases that have the most similar situation to the current one. The retrieval process uses a similarity measure that takes into account the data stored into a situation, mainly the sequence of URLs and the page contents. This step is based on a compound strategy [10] where the concrete cases are first analyzed. If no concrete case with enough similarity is found, the second alternative is executed to retrieve matching potential cases by scanning the navigations. This strategy encourages the reuse of existing concrete cases rather than identifying potential cases. In addition, this strategy leads to a better efficiency. Each retrieved case gives a list of recommendations.

The reuse step. In the reuse step, the pages recommended by the retrieved cases are evaluated, selected and ordered to build the solution, i.e. the final list of recommendations. During this evaluation, the *reusability* of each page is computed with a weighted average of features, such as the number of cases and the best similarity of the cases that recommend this page, the best user's evaluation of the page in the retrieved cases, the presence of annotation, and the average loading time of the page. The pages are ordered according to their reusability, and the best pages are returned. We propose an open model of this reuse strategy so that features can be added or removed [11].

The retain step and the forgetting process. Potential cases that have been identified during the retrieval process are stored in the memory as concrete cases for a direct access. The current navigation is also stored in the memory to be used in the future, only if it is relevant, i.e. the navigation must be longer than three pages and must contain at least one evaluation. In addition, we have defined a forgetting process to cope with the dynamic of the Web where pages can be modified or deleted. Thus, obsolete cases and navigations are deleted from the memory on a regular basis. A concrete case or a navigation is obsolete when a given amount of referenced pages are themselves obsolete. A page is obsolete when it is no more accessible (after multiple retry) or when the extracted keywords have changed significantly.

Similarity Measure Computation. When a target case and a concrete case are compared to each other, three main local similarity measures are used (page URL (1), page content (2) and temporal constraints (3) described in this section) in addition to the standard similarity measure of numbers in an interval for the display time ratio and page evaluation. These local similarities are computed and aggregated with weighted average functions according to these four steps:
1. The local similarity on the last three pages is first computed (cf. similarity measures (1) and (2)) and aggregated. Cases under a given quality threshold are discarded.

2. A backward scanning in the target case is used to identify the last similar pages (on the URL and the keywords) to the selected pages of the source case. Each page is then compared with the four variables. The results of page comparisons are aggregated to build the overall similarity on the selected pages.
3. According to the identified pages in the target case, the ordering similarity is computed using the temporal constraints (cf. similarity measure (3)).
4. The similarities from the steps 1, 2 and 3 are aggregated to get the global similarity.

URL Similarity Measure. The similarity measure of page URLs uses their underlying hierarchical structure (cf. Fig. 6). Pages are indeed grouped into directories. Each directory has an implicit meaning and contains pages that are somehow related. In addition, deeper directories contain more precise information. We must take these features into account in the similarity measure.
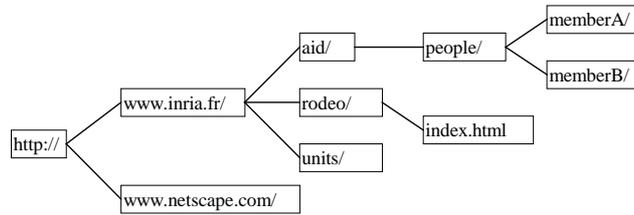
Fig. 6. Hierarchical structure of page URL

A page URL is then identified by its last node, and we use the following similarity measures [4], if p and q are two addresses of pages:

$$S_{ad}(p,q) = 1 - \frac{h(p, \text{MSCA}(p,q)) + h(q, \text{MSCA}(p,q))}{h(p, \text{root}) + h(q, \text{root})} \tag{1}$$

where h() gives the number of links between two nodes and MSCA() gives the most specific common abstraction of two nodes.

Page Content Similarity Measure. Each page content is summarized by a list of keywords extracted from the page after the removal of stop words and a stemming process. A similarity measure based on the number of common words is computed, if v and w are two sets of words:

$$S_c(v,w) = \frac{\text{Card}(v \cap w)}{\text{Card}(v \cup w)} \tag{2}$$

Constraint Similarity Measure. Finally, we use a similarity measure based on the number of satisfied temporal constraints in the target case according to the constraints

defined in a source case. If *v* is the set of constraints in a source case and *w* is the set of constraints satisfied in the current situation:

$$S_{tc}(v,w) = \frac{\text{Card}(w)}{\text{Card}(v)} \qquad (3)$$

4 Experimental Evaluation

An experiment with real users has been lead by ergonomists to evaluate whether BROADWAY effectively assists users during the information searching so that the time spent and the number of visited pages are reduced [7]. For this experiment, 10 students in Psychology were chosen to play the role of users looking for information on the Web. Two groups with equal skills in Web browsing were constituted: the first one had access to recommendations of BROADWAY unlike the second one. All the users were given the same goal : « find information about the biography of Patrick Sommier, a contemporary artist that works probably in the dramatic art ». The starting page was constituted by the first five links given by the Altavista search engine. This subject was carefully selected: none of the students had previously heard about Patrick Sommier and his biography was not directly accessible from the starting page. A preliminary test was made to ensure that the information was reasonably accessible within 30 minutes without assistance so that the searching task was considered as a failure after this delay. An initial case base was created and the goal of the experiment was to compare behaviors of these two groups using different criteria (cf. Table 1): number of success, duration and length of navigations.

Table 1. Main results of the experiment

	With BROADWAY (4 people)	Without BROADWAY (6 people)
Number of success (percentage)	75%	33%
Duration of successful navigations (average)	18 min	24 min
Length of successful navigations (average)	19 pages	39 pages

Thus, in the context of this experiment, the use of BROADWAY has facilitated the information retrieval: there are more successful retrievals, the duration of navigations is reduced and their lengths too. This test shows the first evaluation with real users who were not linked to the development of BROADWAY. Of course we need more experiments with larger sets of users to generalize the results but the basis of our approach is confirmed.

However, during this experiment, the users were told to evaluate systematically every visited pages (this is not required by BROADWAY), and the expectations were that evaluations would be homogeneous. In fact, some users seemed to evaluate pages

based on the theme and others based on the precise request. However, there was a consensus on the most positive evaluation level. Thus, in order to improve the results we are currently studying different alternatives: reduction of the evaluation levels, explicit definition of the semantic of each level, and modification of the evaluations at the end of the navigation by taking the most positively evaluated pages into account.

5   Conclusion

In this paper, we have presented BROADWAY, our browsing advisor for the World Wide Web, which uses case-based reasoning to recommend pages by reusing past navigations of a group of users. Our original approach leads to an indirect collaboration system for Web information searching. This system has been evaluated experimentally with a group of real users and the results showed that BROADWAY assistance was effective in that context. More works have to be done to get the optimum configuration of our system and to generalize the results.

This configuration process is natural in a domain where the knowledge about behaviors of users is poor. That is why the design and the implementation of BROADWAY are based on tools that facilitate its configuration and maintenance: our object-oriented framework CBR*Tools [11] (including our framework for time-extended situation management [8,9]) for the implementation of case-based reasoners and BROADWAY*Tools for the integration of our recommendation server through a distributed architecture. Thus, key concepts such as time-extended situation, similarities, retrieval and adaptation steps can be modified and tuned up. In addition, we are currently studying the use of enhanced user profiles (preferences, topics of interest) and the integration of a revision step.

More generally, we are also studying the synergy of querying and assisted browsing, and we plan to apply our approach in the electronic commerce domain. In this context, we want to reuse customers' behaviors to guide and advise them through an electronic commerce site on the Web. Our approach is interesting to improve a *one-to-one market* instead of the usual mass market approach.

Acknowledgment

References

1. R. Amstrong, D. Freitag, T. Joachims and T. Mitchell. WebWatcher: A Learning Apprentice for the World Wide Web. In *Proceedings of AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, 1995.

2. T. Bray. Measuring the Web. In *Proceedings of the 5th International World Wide Web Conference*, Computer Network and ISDN Systems, 28:993–1005, Elsevier, 1996.

3. C. Brooks, M. S. Mazer, S. Meeks and J. Miller. Application-Specific Proxy Servers as HTTP Stream Transducers. In. *Proceedings of the 4th International World Wide Web Conference*, pages 539–548, 1995.

4. Cognitive System. Remind: developer's reference manual. 200-230, Commercial St., Boston MA 02109, 1992.

5. F. Corvaisier, A. Mille, J.M. Pinon. Information Retrieval on the World Wide Web using a Decison Making System. In *Proceedings of RIAO'97*, 1997.

6. S.Elkassar and J. Charlet. Représentation de connaissances et aide à la navigation hypertextuelle à partir de cas : application au dossier médical. In *Journée Ingénierie des connaissances et apprentissage automatique (JICAA'97)*, pages 387–401, 1997. In French.

7. C. Hébraud, M. Rech and A. Thébault. Test of the Usefulness of the Browsing Assistant BROADWAY. Internal Report, University of Provence, Aix en Provence, 19 pages, 1998. In French.

8. M. Jaczynski. A Framework for the Management of Past Experiences with Time-Extended Situations. In *Proceeding of the 6th International Conference on Information and Knowledge Management (CIKM'97)*, pages 32–39, ACM Press, 1997.

9. M. Jaczynski and B. Trousse CBR*Tools: An Object-Oriented Library for Indexing Cases with Behavioral Situations. Research Report n°3215, 38 pages, INRIA, 1997. In French.

10. M. Jaczynski and B. Trousse. BROADWAY: a World Wide Web Advisor Reusing Past Navigations from a Group of Users. In *Proceedings of 3rd UK Workshop on Case-Based Reasoning (UKCBR3)*, 1997.

11. M. Jaczynski and B. Trousse. An Object-Oriented Framework for the Design and the Implementation of Case-Based Reasoners. In *Proceedings of 6th German Workshop on Case-Based Reasoning (GWCBR'98)*, University of Manchester, pages 33–42, 1998.

12. H. Lieberman. Letizia: An Agent that Assists Web Browsing. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 924–929, 1995.

13. A. Micarelli and F. Sciarrone. A Case-Based System for Adaptive Hypermedia Navigation. In I. Smith and B. Faltings, editors, *Advances in Case-Based Reasoning*, volume 1168 of *Lecture Notes in Artificial Intelligence*, pages 266–279, Springer, 1996.

14. A. Ram and J.C. Santamaria. Continuous Case-Based Reasoning. *Artificial Intelligence*, (90)1-2:25-77, 1997.

16. H. Sakagami and T. Kamba. Learning Personal Preferences on line Newspaper Articles from User Behaviors. In *Proceedings of the 6th International World Wide Web Conference*, Computer Network and ISDN Systems, 29:1447–1455, Elsevier, 1997.

17. C. Shahabi, A. M. Zarkesh, J. Adibi and V. Shah. Knowledge Discovery from Users Web-Page Navigation. In *Proceeding of 7th International Workshop on Research Issue in Data Engineering, High Performance Database Management for Large-Scale Applications*, pages 20–29, IEEE Comput. Soc. Press, 1997.

18. T.W. Yan, M. Jacobsen, H.Garcia-Molina and U. Dayal. From User Access Patterns to Dynamic Hypertext Linking. In *Proceedings of the 5th International World Wide Web Conference*, Computer Network and ISDN Systems, 28:1007–1014, Elsevier, 1996.