BROADWAY: A World Wide Web Browsing Advisor Reusing Past Navigations from a Group of Users

Michel Jaczynski - Brigitte Trousse

INRIA Sophia-Antipolis, Action AID
2004 route des lucioles - BP 93
06902 Sophia Antipolis Cedex, FRANCE
e-mail: {Michel.Jaczynski, Brigitte.Trousse }@sophia.inria.fr

Abstract: The World Wide Web is a huge hypermedia where finding relevant documents is not an easy task. In this paper, we present our case-based browsing advisor, called BROADWAY. BROADWAY follows a group of users during their navigations on the WWW (proxy-based architecture) and advise them by displaying a list of potentially relevant documents to visit next. BROADWAY uses case-based reasoning to reuse precise experiences derived from past navigations with a time-extended situation assessment: the advice are based mainly on similarity of ordered sequence of past accessed documents. In addition, the dynamic of the WWW is addressed in the reuse step and with a specific method for case forgetting.

1. Introduction

The World Wide Web (WWW) [14] is an hypermedia of heterogeneous and dynamic documents. This virtual space is growing more and more every day, offering to the user a huge amount of data [3]. Two kinds of methods can be used to locate a relevant document through this space: *querying* and *browsing*. Querying is appropriate when the user has a clear goal which should usually be expressed through a list of keywords. Different servers on the WWW (such as Yahoo, Lycos, Altavista) can be then used to retrieve matching documents based on their indexing capability. Browsing is well suited when the user cannot express his goal explicitly or when query formulation by keywords is not adequate. Then, the user must navigate through this space, moving from one node to another, looking for a relevant document. These two approaches can be mixed so that querying gives a list of reasonable starting points for browsing [13].

However, the huge size and the structure of this space make difficult the indexing of the documents required by querying access methods and could disorient the user during a browsing session. This article focuses on the aid to the user during a browsing session, and more precisely on the design of a *browsing advisor*. A browsing advisor is able to follow the user during a browsing session to infer his goal, and then must advise him of potentially relevant documents to visit next. In order to improve the relevance of advised 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 WWW*: its use must not be restricted to a localised site, and it must take the dynamic aspect of WWW documents,

- the browsing advisor must *learn from a group of users* how to improve the advice computation,
- the advice 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 [7,9] that a particular state of the navigation (current document and/or an *instantaneous* description) is not sufficient to compute relevant advice. 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 on the WWW [7,18,9,15,1,19,28] but they do not satisfy our requirements. Thus, we propose a browsing advisor, named BROADWAY[1], and based on the Case-Based Reasoning (CBR) paradigm. CBR is used to learn from users' navigations the set of relevant cases, which can be reused to improve and to keep updated the advising process. The use of CBR is based on the following hypothesis: if two users went through a similar sequence of documents, they might have similar browsing intent, so that we can advise one user of the documents evaluated as relevant by the other one (cf. *Figure 1*).

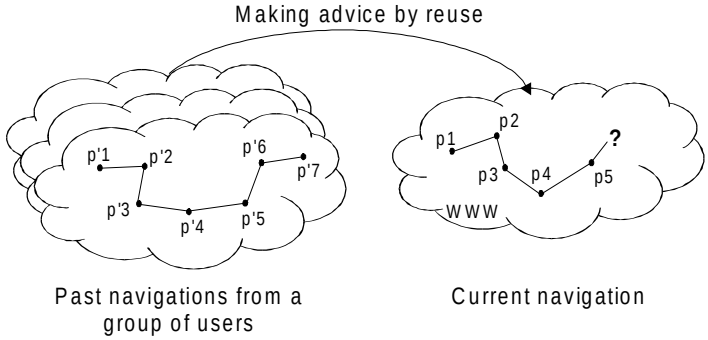

Figure 1: Reusing past navigations to advise users

This paper describes BROADWAY, our browsing advisor. In the section 2, we first introduce the architecture of BROADWAY that is appropriate to follow and advise a group of users independently of their browsers. Then, in the section 3, we propose a representation of navigational cases that integrates time-extended situation. In the section 4, we describe the steps of the case-based reasoner embedded in BROADWAY, and used to compute the list of advised pages displayed to users during their navigations. Finally, we describe briefly in the section 5 the related works according to our requirements.

[1] BROADWAY : a BROwsing ADvisor reusing pathWAYs.

2. BROADWAY Architecture

Based on the analysis given in [17], two kinds of architecture can be used to design a *browsing assistant* on the WWW: *stream transducers* and *browsing associates*. Transducers are inserted into the HTTP (HyperText Transport Protocol) communications between a client and the rest of the web, in order to analyse and possibly alter the stream of requests and responses from the web. Browsing associates are small and autonomous applications accessing the WWW independently to achieve a specific task. Our goal is to follow each user in its navigation through the WWW and the activity of this kind of browsing assistant is highly coupled with the streams of requests and responses. Thus, we have designed BROADWAY as a stream transducer to watch all navigation moves of a group of users. BROADWAY also manages an internal information space that can be accessed by users through a graphical interface mainly to get the advised pages and submit their evaluation. Based on this external architecture, BROADWAY integrates a case-based reasoner that computes advised pages asynchronously from the user's navigation. The overall architecture of BROADWAY is depicted in the Figure 2.

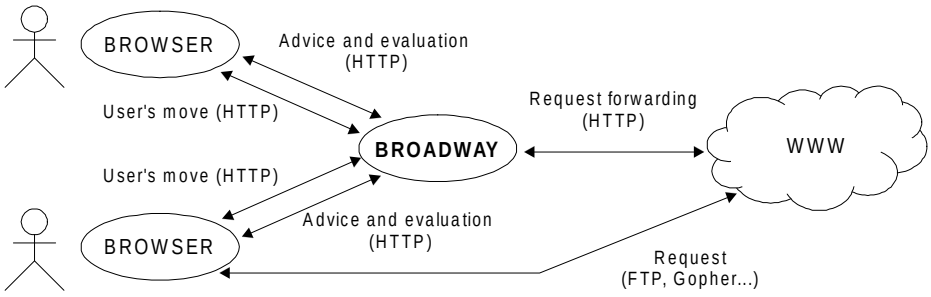

Figure 2: BROADWAY overall architecture

2.1 Watching User's Navigation behind the Scene

BROADWAY integrates a Jigsaw[2] proxy server developed by the W3C (WWW Consortium), in order to implement a stream transducer. A WWW proxy is a special HTTP server that waits for client requests, forwards them when needed to the appropriate server, and returns the answers back to the clients. At the beginning proxies were only used to do security filtering or document caching [16]. In the first case, the proxy, called a firewall, intercepts and checks every HTTP communication between the secured network and the external network. In the latter case, the proxy manages an internal cache of documents, and retrieves the up-to-date document from an external server only when its local copy is obsolete. These uses of proxy servers are usual inside the WWW: any well known browser (Internet Explorer, Netscape Navigator, Hotjava) supports proxy configurations, and the request forwarding is transparent to the users.

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

Proxy architecture is well suited to analyse HTTP communication and the document content in a transparent way, and can also be used to change the content of a reply dynamically before sending it back to the client. These properties have been studied in many works [16,4] and lead to the development of high level applications such as a WWW document annotation server [24] to support a group of users. BROADWAY external architecture is in the straight line of these works. Other browsing advisors have also chosen a proxy-based architecture [19].

BROADWAY accepts multiple simultaneous connections allowing a group of users to share their navigation experiments on the web. 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 its browser, a pop-up window is automatically displayed, asking for its id and password. Once provided, these data will not be asked again until the end of the browser process, and the browser will transmit them with every move of the user. This authentication is important because: it guarantees private access to the proxy which may manage private navigational data and user profiles, and it is used to follow a user through its different moves through the WWW.

BROADWAY intercepts all HTTP replies containing an HTML (HyperText Markup Language) document: all the HTTP headers and the contents of the documents are available for further processing. Above all, these replies are altered in order to *avoid caching* of the documents by a browser or another proxy. Thus, even if the same document is requested several times, the proxy will be aware of these moves. 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.

2.2 Internal Architecture

BROADWAY is composed of three main modules (cf. Figure 3): the Jigsaw proxy, the BROADWAY engine and the case-based reasoner. All these modules are written in the object-oriented Java programming language. The engine and the case-based reasoner use a memory based on Java serialisation for persistency.

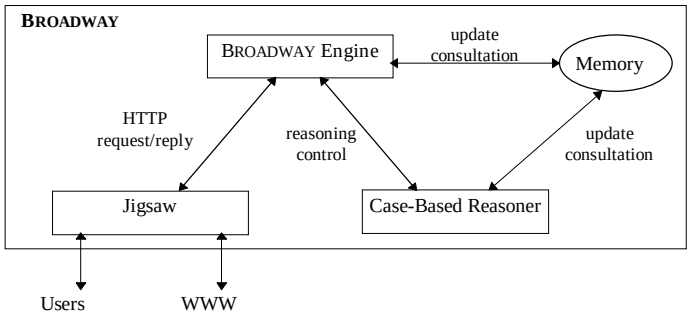

Figure 3: BROADWAY internal architecture

Jigsaw provides basic classes to define specific actions when receiving a request from a client, and when sending back replies in the concept of resource filters. Inside the BROADWAY engine, we have defined our own specific resource filter so that each reply containing an HTML document is analysed. All other types of documents (pictures, videos, animations) are ignored in this processing. The BROADWAY engine also decodes the evaluation given by the user. After the analysis of the HTTP stream, the BROADWAY engine summarises the browsing activity of each user according to four variables[3] that evolve over time:

1. URL of the document,
2. document content description (keywords of the title),
3. user's evaluation of the document,
4. reading time ratio: time spent reading a document relatively to its size (number of characters).

For each user, the BROADWAY engine operates in one of the three following modes: *no observation*, background observation with *advice on need*, background observation with *continuous advice*. In the first mode, BROADWAY is used as a simple caching proxy. In the second mode, the values of the variables are recorded into the memory to update the current state of the user. The user may ask for help and the reasoning starts on the control of the BROADWAY engine. In the third mode, all the variables are updated and a reasoning starts at each move in order to obtain continuous advice. In the current version of BROADWAY, all the accesses to the case-based reasoner are synchronised by the engine so that one reasoning at a time is active, but the reasoning is executed in a specific thread of control which allows an asynchronous processing of request forwarding. Thus, the user browses the WWW as usually, and new advice are displayed once the reasoning is finished. The result of the reasoning is a list of URLs with a percentage representing an estimated relevance.

2.3 Browser Independent User Interface

As the users move through the WWW, BROADWAY updates its memory containing: the histories of the navigations, the list of advised pages computed by the embedded case-base reasoner and user preferences. These information are accessed by users through standard HTTP communication and HTML document description providing a browser independent architecture: query forms, URL with possible query string, Java applets. As each user is always identified, his access is restricted to its own information space. This information space has been divided into different resources which can be displayed by the user inside his browser using multiple windows or HTML frames according to his own preferences. A typical layout of the browsing windows with different frames is presented in the Figure 4.

[3] Other data may be extracted as we are able to access all HTTP message headers and to parse the HTML document.

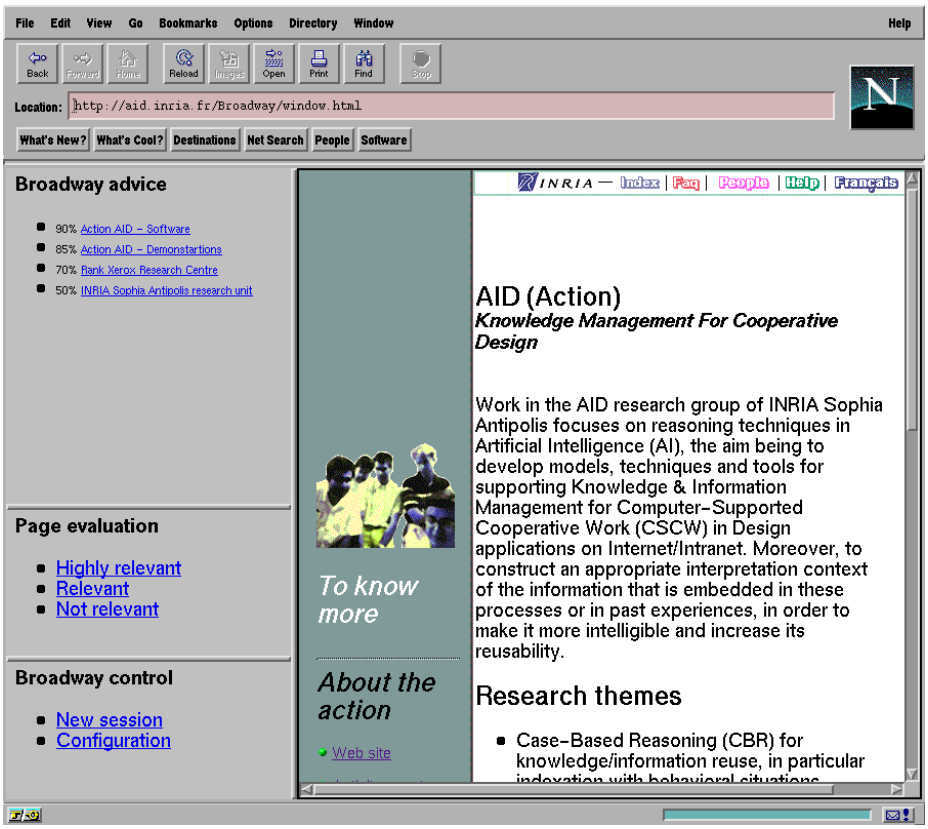

Figure 4: User interface during a browsing session

During the browsing, the user is able to evaluate the current page with one value indicating a negative evaluation (« not relevant ») and two other values for a positive evaluation (« relevant » and « highly relevant »). This evaluation is not a feedback of advice given by BROADWAY, but a comment the user gives on its own navigation as if he was advising another user.

3. Representation of Navigational Cases

In order to improve the advice given by BROADWAY, our goal is to make a time-extended situation assessment during the reasoning. We have previously led an analysis of the management of time-extended situations in case-based reasoning [12] and we have designed a framework for the representation and the retrieval of cases with time-extended situation [11]. We first summarise the main features of this framework supported by our software library called CBR*Tools written in Java. Then we apply our framework to the representation of navigations and navigational cases.

3.1 Representing Cases Indexed by a Time-Extended Situation

In case-based reasoning, the situation of a case defines when its knowledge is relevant, and we make in our work a clear distinction between case indexing techniques based on an *instantaneous situation*, a set of indices giving the 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 behavioural situations inside cases: robot control [21], process forecast [22,20], process supervision [10], trend prognoses for medical problems [25], medical risk detection and forecast [5], plant nutrition control [11], WWW navigation [7,9] (cf. [12] for a detailed analysis).

To cope with this kind of situations, we have designed and implemented a framework for the representation and the retrieval of the cases indexed by a time-extended situation [11]. In this framework we propose the separation of observation data stored in *records* from the cases which reference relevant data inside a record. Records are used to store the observation data of the dynamic process through a fixed number of variables and during a defined time interval. The evolution over time of each variable is represented as a time series. Above the concept of a record, we have defined three types of cases: *abstract*, *potential* and *concrete* cases. Abstract cases come from the domain knowledge or they are built by manual or automatic generalisation [2]. Potential cases do not have any concrete representation, and the knowledge they represent is hidden inside the records that are stored in memory. They can be identified by a direct search inside the records according to a *potential case template* defining typical situation constraints. These cases cannot be used directly, but due to some new problems, potential cases could become explicit as concrete cases.

This framework has been implemented in an object-oriented library written in Java and called CBR*Tools. The library provides basic classes and facilities to represent and manage cases with time-extended situation. BROADWAY is developed using this library: the records are specialised into the concept of *navigation* and the *navigational cases* reference a precise experience inside a navigation. BROADWAY uses two types of cases[4]: concrete cases to represent an explicit experiences acquired from a navigation, and potential cases which are hidden in the navigations.

3.2 Representation of a WWW Navigation

The WWW is composed of a set of resources which are identified by a unique address, the URL[5] . Each resource may contain other resources or references to other resources so that the WWW appears to be an hypermedia. Many protocols and data types are used in the WWW, and it is important for a browsing advisor such as BROADWAY to define the part of the web it aims to support. In our current version of BROADWAY, the WWW is represented as a directed graph of HTML *pages* identified

[4] Abstract cases are not used in the current version of BROADWAY yet.
[5] Universal Resource Identifier also known as Universal Resource Locator.

through a HTTP URL. This means that other types of documents (image and videos for instance) and other types of URL (such as Gopher, Wais, FTP and files[6]) are not taken into account by BROADWAY but can be still accessed by the user. An HTTP URL is a string composed of different parts[7] :

HTTP_URL = "http://" host [":" port] ["/" path ["?" query] ["#" fragment]]

Based on the requirements about URL comparison[7], we have defined our concept of page address with specific equality constraints. In BROADWAY, two page addresses are *equal* if: they have the same host and port, the same path and the same query. The fragment is ignored because it references a part of the same document. The port 80 is assumed by default. The comparison of the host is case-insensitive, and if two host names are different, their IP addresses are then compared.

Address of page A	Address of page B
http://www.inria.fr	http://www.inria.fr/index.html
http://www.inria.fr :80	http://WWW.inria.fr/index.html
http://www.inria.fr/	http://www.inria.fr/index.html#theme

Figure 5: Example of equal page addresses

Using this representation of the WWW, the *current location* of a user is the last accessed page and a browsing *move* is a transition from a page address to a different one. A *browsing session* also called a *navigation* is mainly a sequence of pages representing the moves of one users over time. A navigation is assumed to be *coherent* in such a way that the user do not mix different browsing intent during a single navigation.

In BROADWAY, the navigations are recorded based on the evolution of the four variables coming from the analysis of the HTTP request/reply stream: page address, page content description, user's evaluation, reading time ratio (cf. Figure 6). 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. We associate to each navigation a *context* containing a synthetic description of the pages accessed during the navigation (most used keywords in page contents, and hosts accessed).

[6] See RFC1738 for information about URL types.
[7] See RFC2068:§3.2.2 and RFC 1738.

#	Page address (default prefix: http://www.inria.fr)	Page content description (keywords of the title)	eval.	Ratio
1	/welcome-eng.html	WELCOME, INRIA, WEB, SITE	-	4.4
2	/Recherche/activites-eng.html	INRIA, RESEARCH, ACTIVITIES	-	7.8
3	/Themes/Theme3-eng.html	INRIA, THEME	-	2.8
4	/Equipes/AID-eng.html	AID, ACTION	-	25.4
5	/aid/aid-eng.html	ACTION, AID	-	5.5
6	/aid/people.html	ACTION, AID, PEOPLE	-	2.6
7	/aid/personnel/Michel.Jaczynski/michel-eng.html	MICHEL, JACZYNSKI	-	14.7
9	/aid/personnel/Michel.Jaczynski/pub-fra.html	MICHEL, JACZYNSKI, PUBLICATION	-	7.7
12	/aid/people.html	ACTION, AID, PEOPLE	-	2.1
13	/aid/software.html	ACTION,AID, SOFTWARE	relevant	7.3
14	/aid/people.html	ACTION, AID, PEOPLE	-	1.9
15	http://aid.inria.fr/index.html	ACTION, AID, DEMONSTRATION	highly relevant	-

Figure 6: Extract of a navigation recorded by BROADWAY. In this navigation, the user intent was to find more information about BROADWAY so he navigated through the INRIA server, and found two relevant pages: the software list and the demos descriptions.

3.3 Navigational Cases

In BROADWAY, potential cases (based on a situation template) and concrete cases are composed of:

- a *time-extended situation* from a navigation,
- a list of pages which can be advised in that situation,
- a set of data used to manage the case such as the date of creation, and the user's name.

The time-extended situation has an instantaneous part and a behavioural part. The instantaneous part contains the navigation context which is shared by all the cases based on the same navigation. The behavioural part defines which pages or sequence of pages are relevant in the description of this situation.

We have defined a situation template that can be applied at a precise instant of the navigation to build the situation. The rules described in this template create a behavioural description composed of (cf. Figure 7) :

- The last three pages and their keywords. This selection is used to describe a precise step in the user's navigation and is called the *case position* ;

- A set of past relevant pages (with their content descriptions, user's evaluation and time reading ratio). In this selection relevant pages are pages evaluated by a user (positively or negatively) and/or pages with a high reading time ratio. We use explicit (user's evaluation) and implicit (time reading ratio) features to select relevant pages to improve the selection accuracy [23] ;
- A set of *before* constraints which are used to relate selected pages.

This template is used to create concrete cases. A concrete case references a navigation at a specific instant (*time reference*), making the separation between the past and the future. The list of advised pages represents the future pages of the case, which have been evaluated relevant or highly relevant by the user (cf. Figure 7).

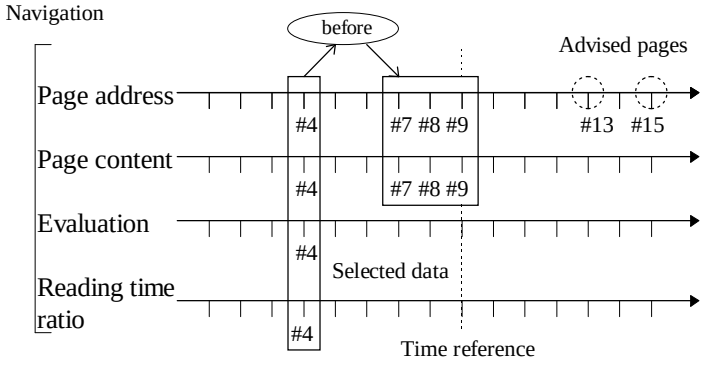

Figure 7: Association of a behavioural description with advised pages inside a concrete case. This example is based on the navigation of the Figure 6. The page #4 is selected because of its high reading time ratio (above 20). The pages advised by this case are #13 and #15 because they have been evaluated as relevant by the user.

4. Reusing Past Navigations

BROADWAY is accessed by multiple users, and when an advice must be given to a user, the BROADWAY engine starts a new reasoning. The user's current navigation represents the indices of the current problem to solve. The reasoner uses three steps: it retrieves cases with similar time-extended situation, then it reuses the past navigational cases to give appropriate advice, and it finally retains this current experience. In this section, we describe the retrieval and reuse steps of the case-based reasoner of BROADWAY and we propose a case forgetting method required to cope with dynamic changes of the WWW.

4.1 Case Retrieval

The case retrieval in BROADWAY uses a complex strategy with two alternatives: concrete case retrieval and potential case retrieval. The retrieval is built using our CBR*Tools library for case-based reasoning that provides simple index structure

which can be composed to define composite index. In addition to these alternatives, we want to emphasise the use of three similarity measures used in the comparison of situations.

4.1.1 Retrieval Alternatives

The first alternative only considers concrete cases. If no concrete cases with enough similarity are found, the second alternative will be executed to retrieve matching potential cases. This strategy encourages the reuse of existing concrete cases rather than identifying potential cases. In addition, this strategy leads to a better efficiency of the retrieval because the first alternative requires less computations.

The first alternative is composed of two steps:

1. *concrete case position filtering*: inside its behavioural situation, each case defines its position. Each case position has the same structure and defines a vector of values. BROADWAY uses a K-d tree [27] approach[8] to filter the cases which are above a similarity threshold. The aims of this step are to ensure a minimal relevance of retrieved cases and to speed up the retrieval process.
2. *concrete case selection*: for a fine grained selection, additional selected pages with their evaluation and temporal constraints are used to select best cases through a KNN search.

The second alternative requires three steps :

1. *navigation filtering*: a crisp comparison is made between the current context and the past navigation contexts. The goal of this step is to discard navigations that have totally different contexts from the retrieval in an efficient way. BROADWAY uses hashtables to get navigations that have at least one host or one keyword in common with the current context.
2. *potential case position filtering*: the identified navigation are scanned through a sequential search to find positions that are above a similarity threshold. For these positions, the situation template is instanciated and a potential case is created.
3. *potential case selection:* as in the concrete case selection a fine grained KNN search is done on the potential cases. Selected potential cases will be then stored in the case base as concrete cases by the retain step of the reasoner.

4.1.2 Similarity Measures

BROADWAY defines local similarity measures for each variable and for the temporal constraints comparison. Then a standard global similarity measure (weighted mean) is used to aggregate local similarities. We describe the three main local similarity measures used to compare: page addresses, page contents and temporal constraints.

[8] This feature is not yet implemented, a simple KNN search is used temporary.

Similarity measure of page addresses. The similarity measure of page addresses uses the underlying hierarchical structure of addresses (cf. Figure 8). 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.

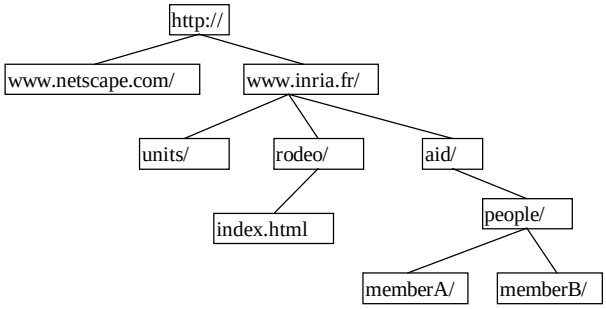

Figure 8: Hierarchical structure of a page address.

A page address is then identified by its last node and we use the following similarity measures [6], if *p* and *q* are two address of pages:

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

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

For instance:
S_{ad}(http://www.inria.fr/aid/pepole/memberA, http://www.inria.fr/aid/pepole/memberB) = 0.66
S_{ad}(http://www.inria.fr/aid/pepole/memberA, http://www.inria.fr/rodeo/index.html) = 0.28
S_{ad}(http://www.inria.fr/rodeo, http //www.inria.fr/aid) = 0.5

Similarity measure of page contents. Each page content is summarised by a list of keywords. A similarity measure based on the number of common words is computed (cf. [26]), if *v* and *w* are two sets of words:

$$S_c(v,w) = \frac{Card(v \cap w)}{Card(v \cup w) + Card(v - w) + Card(w - v)}$$

Similarity measure of temporal constraints. We use an optimistic similarity measure, based on the number of satisfied constraints in the current navigation according to the constraints defined in each case. If *v* is the set of constraints in a case and *w* is the set of constraints satisfied in a problem:

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

4.2 Case Reuse

The retrieval step has identified a list of relevant cases and each case gives a list of advised pages. In the reuse step, the k-most reusable pages are selected and ordered according to their *reusability*. The reusability of each page is based on a weighted mean of the following features:

- the number of cases that advise this page,
- the best similarity of the cases that advise this page,
- the best user's evaluation of the page in the retrieved cases,
- the minimum number of moves from the case reference time to the page,
- the number of successful access to this page,
- the number of access failures,
- the average loading time of the page.

Thus, BROADWAY selects the pages not only for their relevance but also for their access features. The WWW is a dynamic hypermedia where pages are deleted, moved or modified. In addition, depending on the location of the user and the page server, different loading time may be observed. BROADWAY keeps track of access to pages and these data are taken into account in the reuse step for better advice.

4.3 Case Forgetting

The WWW is a dynamic space and some past experiences may become obsolete. For this reason, we have design a module which runs on a regular basis in order to delete from the memory obsolete cases and navigations (*case forgetting*). A concrete case is obsolete when the advised pages are all obsolete or when more than 25% of the pages selected in its time-extended situation are obsolete. A page is considered obsolete when it is no more accessible (after multiple retry) or when the similarity of the page content stored in the case and its current content is under a threshold. Navigations are also checked in the same way, and if the navigation is obsolete then all its cases and the navigation itself are forgotten. Thus, potential cases will not be retrieved again in this navigation.

5. Related Works

We have analysed the different hypermedia browsing advisors according to our goals. We have separated case-base approaches from others and the comparison is summarised at the end of the section in Figure 9.

5.1 Case-Based Browsing Advisors

Radix [7] is based on a deeper description of the browsing session than BROADWAY. The observation of users' actions, such as bookmark selection, page address edition, back or forward link selection, are required to represent a session and its components. A case is an entire information retrieval session whereas in BROADWAY a case represents a specific experience in a navigation. A time-extended situation is taken into account with a event-driven similarity [8] during the retrieval. It is not

clear how Radix could manage a group of users even if it uses an object-oriented database to store cases. Radix uses a specific browser with customised functions to watch user's actions, so its use is restricted to a specific platform.

Hypercase [18] is a browsing advisor designed to help a user on a delimited hypermedia. The definition of each node is required and this approach is not appropriate to address WWW navigation assistance. In addition, Hypercase cannot learn from real navigations of a group of users because its reasoning is only based on pre-stored cases built by experts.

The goal of Hospitext [9] is to assist different types of users during the browsing of medical records of patients. It learns from past navigation and a hierarchy of navigation behaviour is built. Hospitext uses domain knowledge mainly in the definition of a taxonomy of documents used in the case matching. This approach is not appropriate from WWW browsing since a taxonomy of documents seems hard to define.

5.2 Non Case-Based Browsing Advisors

Letizia [15] is an agent that assists a user by browsing concurrently and autonomously from the current page. Letizia uses the time spend by the user reading the current document to explore the neighbourhood and anticipate user's browsing. However, Letizia does not learn from its experiences and it builds only a description of the user's interest for the current session by recording its actions.

WebWatcher [1] follows the user along its navigation and highlights the hyperlinks of the current page that are of interest. WebWatcher learns from previous navigation of different users based on several machine learning algorithms. However, the situation assessment does not take the past sequence of events into account, but only an instantaneous summary through vector of keywords. WebWatcher is based on a HTTP stream transducer not implemented by a proxy, and the all HTML pages must be analysed and altered to redirect hyperlinks to itself. PersonalWebWatcher (PWW) [19] is similar to WebWatcher but it aims to assist only one user with this time a proxy-based architecture.

Yan *et al*. [28] propose a modified WWW server that logs all documents access for a user session. The browsing sessions are then analysed off-line to build clusters of access patterns. Then when the user gets connected to this specific site, his document access is analysed, a set of matching clusters are identified and the documents not yet accessed are advised to the user. However, the order of the accessed pages is not taken into account and inaccuracy may be caused by caching mechanisms from proxies and browsers.

Browsing advisors	designed for the WWW	learn from a group of users	time-extended situation	browser independent
Radix [7]	yes	-	yes	no
Hypercase [18]	no	no	no	-
Hospitext [9]	no	yes	yes	-
Letizia [15]	yes	no	no	no
WebWatcher [1]	yes	yes	no	yes
PWW [19]	yes	no	no	yes
Yan *et al* [28]	no	yes	no	yes

Figure 9: Comparison of related works according to our goals

6. Conclusion

In this paper we have introduce BROADWAY, our browsing advisor for the World Wide Web, which uses case-based reasoning to advise pages by reusing past navigations of a group of users. BROADWAY is based on a proxy architecture so that it is able to follow a group of users on any sites of the Web. As it communicates with the user through standard protocols, BROADWAY is independent of the browser which enables its use on different platforms, with an up-to-date browser chosen by each user.

BROADWAY integrates a case-based reasoner with concrete and potential cases. Concrete cases represent specific experiences in a navigation which can be retrieved efficiently. When concrete case cannot be used, potential cases are searched in the past navigation for knowledge discovery. BROADWAY takes time-extended situation into account: its advice are based on similarity of past navigational behaviours. We have defined a similarity for each page using its address and content. The similarity of page ordering is computed using temporal constraints between pages. We take into account that WWW pages can be altered or deleted in the reuse step and we have defined a specific method for case forgetting.

In addition, BROADWAY is implemented in Java using our software library CBR*Tools that supports our framework for the management of cases with time-extended situations [11,12]. BROADWAY is the second application of this framework (cf. [11]). Our aim is to provide generic CBR tools and methods to support reasoning with time-extended situations.

The relevance of advice given by BROADWAY must still be carefully evaluated. We also plan to improve some part of the design: multi-threaded reasoning, integration of user and session profiles in the navigation context, and use of expert knowledge about navigations. The two latter perspectives are currently studied with RXRC[9].

[9] Rank Xerox Research Centre of Grenoble - France.

Acknowledgement

We would like to thank A. de Boussineau for her useful comments.

References

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

[2] R. Bergmann and W. Wilke. On the role of abstraction in case-based reasoning. In I. Smith and B. Faltings, editors, *Advances in Case-Based Reasoning*, volume 1168 of *Lecture Notes in Artificial Intelligence*, pages 28–43. Springer, 1996.

[3] T. Bray. Measuring the Web. In *Proc. of the 5th International World Wide Web Conference*, Computer Network and ISDN Systems, 28 :993–1005, 1996.

[4] C. Brooks, M. S. Mazer, S. Meeks and J. Miller. Application-specific proxy servers as HTTP stream transducers. In. *Proc. of the 4th International World Wide Web Conference*, Boson, 1995.

[5] M. Bull, G. Kundt, and L. Gierl. Case-based risk detection and forecasting in a geographic-medical system. In R. Bergmann and M. Wilke, editors, *German Workshop on Case-Based Reasoning*, pages 59–64, March 1997.

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

[7] F. Corvaisier, A. Mille, J.M. Pinon. Information retrieval on the World Wide Web using a decison making system. In *Proceedings of RIAO'97*, June 97.

[8] F. Corvaisier, A. Mille, J.M. Pinon. Information retrieval on the WEB using a CBR system: focusing on the similarity problem. Internal report, CPE Lyon, March 97.

[9] 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, Mai 1997. In French.

[10] B. Fuch, A. Mille, and B. Chiron. Operator decision aiding by adaptation of supervision strategies. In M.Veloso, K.D. Althoff, and M. M. Richter, editors, *Case-Based Reasoning Research and Development*, Lecture Notes in Artificial Intelligence, pages 23–32. Springer, 1995.

[11] M. Jaczynski. A framework for the management of past experiences with time-extended situations. In *Proceedings of CIKM'97*, Las Vegas, 1997. To appear.

[12] M. Jaczynski and B. Trousse CBR*Tools: an object oriented library for indexing cases with behavioural situation. Research Report n°3215, INRIA, July 1997. In French.

[13] K-H. Jerke, P. Szabo, A Lesh and H. Rossler. Combining hypermedia browsing with formal queries. In D. Diaper et al., editors, *Human-Computer Intercation - Interact'90*, pages 593–598, 1990.

[14] T.B.Lee *et al*. The World Wide Web. *Communication of the ACM*, 37(8) :76–82, 1994.

[15] H. Lieberman. Letizia: An agent that assists web browsing. In Proceedings of International Joint Conference on Artificial Intelligence, Montreal, August 1995.

[16] A.Luotonen and K. Altis. World-Wide Web Proxies. In *1st International World Wide Web Conference*, Geneva, 1994.

[17] W. S Meeks, C. Brooks and M. S. Mazer. Transducers and associates: circumventing limitations on the World Wide Web. In *Proceedings. of etaCOM'96*, Portland, Oregon, May 1996.

[18] 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.

[19] D. Mladenic. Personal WebWatcher: design and implementation. Technical report IJS-DP-7472, School of Computer Science, Carnegie Mellon University, October 1996.

[20] G. Nakhaeizadeh. Learning prediction from time series: a theoretical and empirical comparison of CBR with some other approaches. In *Topics in Case-Based Reasoning*, volume 837 of *Lecture Notes in Artificial Intelligence*, pages 65–76. Springer, 1994.

[21] A. Ram and J.C. Santamaria. Continuous case-based reasoning. In *AAAI Case-Based Reasoning Worksop*, pages 86–93, 1993.

[22] S. Rougegrez. Similarity evaluation between observed behaviours for the prediction of processes. In *Topics in Case-Based Reasoning*, volume 837 of *Lecture Notes in Artificial Intelligence*, pages 155–166. Springer, 1994.

[23] H. Sakagami and T. Kamba. Learning personal prefences on line newspaper articles from user behaviors. In *Proc. of the 6th Internatinal World Wide Web Conference*. 1997.

[24] M. A. Schicker, M. S Mazer and C. Brooks. Pan-browser support for annotations and other meta-information on the World Wide Web. In *Proc. of the 5th International World Wide Web Conference*, Computer Network and ISDN Systems, 28:1063–1074, May 1996.

[25] R. Schmidt, B. Heindl, B. Pollwein, and L. Gierl. Abstraction of data and time for multiparametric time course prognoses. In I. Smith and B. Faltings, editors, *Advances in Case-Based Reasoning*, volume 1168 of *Lecture Notes in Artificial Intelligence*, pages 377–391. Springer, 1996.

[26] A. Tversky. Features of similarity. *Psychological Review*, 84(4), 1977.

[27] S. Wess, K.D. Althoff, et G. Derwand. Using K-d Trees to improve the retrieval step in case-based reasoning. In S. Wess, K.D. Althoff and M. M. Richter, editors, *Lecture Notes in Artificial Intelligence, Topics in Case-Based Reasoning*, pages 167-181, Springer, 1994.

[28] T.W. Yan, M. Jacobsen, H.Garcia-Molina and U. Dayal. From user access patterns to dynamic hypertext linking. In *Proc. of the 5th International World Wide Web Conference*, Computer Network and ISDN Systems, 28:1007–1014, 1996.