Extended Memory (xMem) of Web Interactions

Stefano Ceri Florian Daniel Maristella Matera Francesca Rizzo
Dipartimento di Elettronica e Informazione
Politecnico di Milano
Piazza Leonardo da Vinci, 32 – 20133 Milano – Italy
{ceri,daniel,matera,rizzo}@elet.polimi.it

ABSTRACT

Finding a previously visited page during a Web navigation is a very common and important kind of interaction. Most commercial browsers incorporate history mechanisms, which typically are simple indexes of visited pages, sorted according to the time dimension. Such mechanisms are not very effective and are quite far from giving users the impression of a semantically aware, long-term memory, as it is available to the human brain. In particular they lack associative, semantic-based mechanisms that are essential for supporting information retrieval. This paper introduces xMem (eXtended Memory Navigation) as a new method to access users' navigation history, based upon semantic and associative access. Its aim is to emulate some of the features of the human memory, so as to give users a better understanding of the context of their searches, by exploiting semantic cues characterizing contents of visited pages.

Categories and Subject Descriptors

H.1 [**Information Systems**]: Models and Principles; H.5.4 [**Information Interfaces and Presentation (e.g.,HCI)**]: Hypertext/Hypermedia; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Design, Human factors, Experimentation.

Keywords

Web Navigation History, Semantic Memory, Usability studies, Information extraction.

1. INTRODUCTION

As the amount of information on the World Wide Web continues to grow, efficient hypertext navigation mechanisms are becoming crucial. Among them, special attention must be devoted to effective history mechanisms enabling users to get back to information already met [19].

History tools are common components of Web browsers, whose introduction is motivated by three factors: (1) They help users navigating through the huge quantity of information provided by the Web, thus providing access to information previously visited. From the user perspective, this is consistent with the need for effective navigation in the Web. (2) They substitute search engines for finding old pages and avoid the replication of navigations along intermediate pages to a destination page. In other words, this means that history mechanisms improve the efficiency of navigation in the Web. (3) They positively affect users' activities by reducing cognitive and physical navigation burdens: pages can be retrieved with little effort, and users can easily locate where they have been in the past. More specifically history tools would provide a satisfactory experience to their users.

Most of the results coming from the field of HCI [19, 8, 3] show that more of the 30% of users' activities on the Web are based on the use of the back button or of favorites, but they also show that reverse browsing mechanisms are time consuming and cognitively difficult to use, organize and envision. The reason is that these mechanisms are efficient for the revisitation of the short-term or the most frequent information memories, but fail when supporting people in the retrieval of the old pages. These reasons have led us to design a new method to access navigation histories.

This paper introduces xMem (eXtended Memory Navigation), which aims at providing Web users with advanced memory mechanisms exploiting page classification based on keywords extracted from page contents. Indexing and classificatiaon can be achieved in two ways: (i) the Web application designer can provide xMem with explicit descriptions of page contents through page annotation mechanisms; (ii) when explicit annotations are not available, a *Page Indexer* module extracts keywords from page contents. Keywords are then grouped into meaningful clusters.

The paper is structured as follows. Section 2 outlines the goals of our research and introduces the xMem concepts. Section 3 illustrates a controlled experiment that we have conducted with 45 users to verify the effectiveness of the xMem history with respect to traditional paradigms. Section 4 provides details about the xMem architecture and the mechanisms for deriving keywords and clusters for the visited pages. Section 5 discusses the implementation of the current prototype tool. Section 6 then provides insight into related works. Finally, Section 7 presents some conclusions and gives an outlook over our ongoing and future work.

2. THE XMEM WEB INTERACTION MEMORY

Most commercial Web browsers incorporate history mechanisms. However, users still do not extensively use them. Catledge and Pitkow [3] state that these complex history mechanisms are often under-used, and the 40.6% of the user navigation actions involve the browser's back button. In [8] authors also show that 0.1% of Web page accesses occur through the history list, while 42% of page accesses used the back button.

Users experience frustration in retrieving already visited pages, when a certain amount of time is passed from the first visit. The path-following method (on which the most part of the history mechanisms are based) for retrieving long term history memories imposes users to traverse in reverse order their previously visited pages. This method requires users to remember their navigational behaviors, either because they must recall the visited pages and their sequence, or because they must realize that they can return to a page by retracing a particular pathway. However, very often the context in which that page was being viewed is lost, and this phenomenon may be due to the lack of efficient mechanisms for maintaining context.

In the light of the previous considerations, the aim of xMem becomes twofold:

(i) *Providing easier and more intuitive history navigation mechanisms.* Web history navigation can be aided by semantically rich and associative navigation paradigms, providing enriched access that builds on page indexing, by means of keywords, and on the classification of navigated contents into meaningful clusters. As shown is Section 3, this promotes xMem as an efficient tool for accessing the users' long-term memory.

(ii) *Fostering ubiquitous accessibility*, by making the navigation history accessible over the Internet. The gathered and classified information must be always reachable, independently from the work place and the adopted device, in order to become an active support that can be easily integrated within the user's browsing environment.

2.1 xMem Concept

Trying to solve the previous issues, the xMem project offers to users a specialized Web site hosting a repository of the individual user's memory. Besides chronologically ordered lists of URLs (as offered by traditional histories), xMem provides further hints in presenting history data. Keywords are associated to groups of navigated pages to recall concepts describing the page contents. Keywords may be provided by Web application designers as meta-description of pages, or they can be extracted automatically from re-materialized pages by detecting the main concepts being displayed by the pages themselves.

In order to take advantage of the navigation history facilities of xMem, users must register as xMem users and install an application on their client PCs. While navigating the Web by means of a common Web browser, users can activate this application for tracking the navigated URLs. Such application also manages the transparent transfer of the tracked URLs to the xMem system. The tracking mechanism can be enabled/disabled at will. Thus, xMem maintains remote log data (with respect to users and 3rd party Web servers) about users' navigation actions.

At server side, xMem identifies keywords representing concepts seen during navigation, which are used for populating the history memory. xMem then provides a Web interface on the history memory that, besides the chronological lists of URLs, also offers a semantic organization of history data based on page keywords recalling page contents at a comprehensible level of abstraction.

2.2 Defining Semantics for Web Pages

Some studies on present browsers [8] have demonstrated that the chronological criterion they exploit to sort their history does not significantly help end users when they try to re-access a previously visited page. Viceversa, the xMem basic idea is that semantically cuing the history list would be more effective in all the situations in which people do not exactly remember when and in which context they visited the page. The enrichment of history data is achieved through the extraction of keywords, describing pages, and through the clustering of pages into categories, based on keyword similarity.

We define a *keyword* k with respect to a page p as a natural language term derived from page p, which serves the purpose of indexing the content of page p. A page p can be characterized by a set of keywords. We then define a *cluster* C as a collection of pages characterized by similar keywords. In our approach, keyword extraction and cluster definition are achieved by means of syntactic techniques.

Keywords and clusters represent semantic classes used by xMem for achieving a hierarchical organization of visited URLs. The first hierarchy level is based on page clusters, grouping *similar* keywords, while the second builds on keywords, used for grouping pages with the *same* keywords. The finest granularity of items presented in the hierarchy is given by the tracked URLs, which allow users to access the actual page searched.

The clustering of history data highlights correlations between visited pages, perhaps not emerged during the original navigation. Such correlations might not be captured adequately by traditional, time-based histories.

3. VALIDATING THE XMEM HISTORY PARADIGM

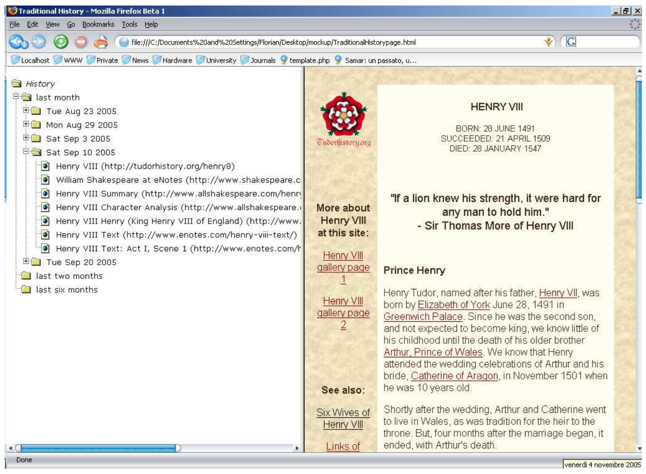
In order to investigate our assumptions about the semantic enrichment of history, we have conducted an experiment with real users. We wanted to compare the performance of users using a traditional history based on chronological order, with the performance of users adopting an enriched history. In particular, we wanted to verify whether the semantic enrichment supplying additional cues about the page contents significantly improves the user experience. Also, we wanted to verify the effectiveness of the hierarchical organization of the enriched history.

We therefore designed a controlled experiment, comparing three different history organizations:

- The *Traditional History* (TH), based on a chronological sorting of visited URLs (see Figure 1).
- A *Hierarchical xMem History* (HX), based on a hierarchical organization of the enriched history using clusters and keywords for classifying URLs (see Figure 2).
- A *Flat xMem History* (FX), where clusters provide a one-level classification of pages, while keywords just extend the descriptions of URLs (see Figure 3).

Our hypotheses were the following:

1. When a user does not remember the access time to a previously visited page, finding that page, based on a


Traditional History - Mozilla Firefox Beta 1
File Edit View Go Bookmarks Tools Help
file:///C:/Documents%20and%20Settings/Florian/Desktop/mockup/TraditionalHistorypage.html
Localhost WWW Private News Hardware University Journals template.php Samar: un passato, u...
Done
venerdì 4 novembre 2005

```
History
  last month
    Tue Aug 23 2005
    Mon Aug 29 2005
    Sat Sep 3 2005
    Sat Sep 10 2005
      Henry VIII (http://tudorhistory.org/henry8)
      William Shakespeare at eNotes (http://www.shakespeare.c
      Henry VIII Summary (http://www.allshakespeare.com/henry
      Henry VIII Character Analysis (http://www.allshakespeare.
      Henry VIII Henry (King Henry VIII of England) (http://www.
      Henry VIII Text (http://www.enotes.com/henry-viii-text/)
      Henry VIII Text: Act I, Scene 1 (http://www.enotes.com/h
    Tue Sep 20 2005
  last two months
  last six months
```

HENRY VIII

BORN: 28 JUNE 1491
SUCCEEDED: 21 APRIL 1509
DIED: 28 JANUARY 1547

"If a lion knew his strength, it were hard for any man to hold him."
- Sir Thomas More of Henry VIII

More about Henry VIII at this site:
Henry VIII gallery page 1
Henry VIII gallery page 2

See also:
Six Wives of Henry VIII
Links of

Prince Henry

Henry Tudor, named after his father, Henry VII, was born by Elizabeth of York June 28, 1491 in Greenwich Palace. Since he was the second son, and not expected to become king, we know little of his childhood until the death of his older brother Arthur, Prince of Wales. We know that Henry attended the wedding celebrations of Arthur and his bride, Catherine of Aragon, in November 1501 when he was 10 years old.

Shortly after the wedding, Arthur and Catherine went to live in Wales, as was tradition for the heir to the throne. But, four months after the marriage began, it ended, with Arthur's death.

Figure 1: Traditional history organization: chronologically ordered list of URLs.

pure temporal sorting, such as the one used by traditional histories, requires a higher cognitive effort. We therefore expect a better performance by xMem users.

2. The hierarchical organization of history data, as also demonstrated in other domains, improves the retrieval task. We therefore expect a better performance of the HX history with respect to the FX history.

3. We finally hypothesize that the enriched (both hierarchical and flat) history enhances user satisfaction. We however expect a higher satisfaction for HX users with respect to FX users.

3.1 Method

The experiment has been designed in order to identify significant differences in the time taken to retrieve a page, using the three different histories.

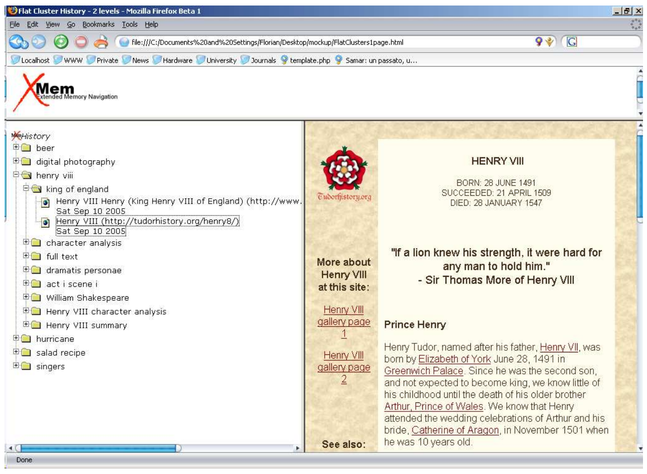
Participants were selected among the undergraduate students of the University of Lugano, Switzerland (Università della Svizzera Italiana). 45 subjects were recruited and allotted into three experimental conditions:

- The TH group was asked to use the chronological history.
- The HX group was asked to use the enriched hierarchical history.
- The FX group was asked to use the enriched flat history.

Subjects' performance was measured on the task completion time. At the end of the experiment, participants were required to fill in a questionnaire combining three different dimensions: (i) the previous experience of participants with history mechanisms; (ii) the knowledge of participants about page contents, (iii) the subjects' satisfaction with the tools.

3.1.1 Materials

Three mockups were purposely implemented for the study, one for each history under test (see Figures 1 – 3). Each subject was exposed to the same list of 40 URLs.


Flat Cluster History - 2 levels - Mozilla Firefox Beta 1
File Edit View Go Bookmarks Tools Help
file:///C:/Documents%20and%20Settings/Florian/Desktop/mockup/FlatClusters1page.html
Localhost WWW Private News Hardware University Journals template.php Samar: un passato, u...
XMem
Extended Memory Navigation
Done

```
History
  beer
  digital photography
  henry viii
    king of england
      Henry VIII Henry (King Henry VIII of England) (http://www.
      Sat Sep 10 2005
      Henry VIII (http://tudorhistory.org/henry8/)
      Sat Sep 10 2005
    character analysis
    full text
    dramatis personae
    act i scene i
    William Shakespeare
    Henry VIII character analysis
    Henry VIII summary
  hurricane
  salad recipe
  singers
```

HENRY VIII

BORN: 28 JUNE 1491
SUCCEEDED: 21 APRIL 1509
DIED: 28 JANUARY 1547

"If a lion knew his strength, it were hard for any man to hold him."
- Sir Thomas More of Henry VIII

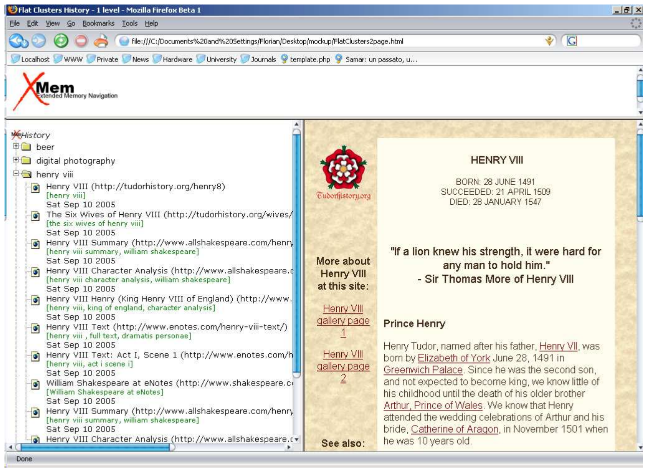
More about Henry VIII at this site:
Henry VIII gallery page 1
Henry VIII gallery page 2

See also:

Prince Henry

Henry Tudor, named after his father, Henry VII, was born by Elizabeth of York June 28, 1491 in Greenwich Palace. Since he was the second son, and not expected to become king, we know little of his childhood until the death of his older brother Arthur, Prince of Wales. We know that Henry attended the wedding celebrations of Arthur and his bride, Catherine of Aragon, in November 1501 when he was 10 years old.

Figure 2: Hierarchical xMem history: URLs are accessed through semantic classes based on clusters and keywords.


Flat Clusters History - 1 level - Mozilla Firefox Beta 1
File Edit View Go Bookmarks Tools Help
file:///C:/Documents%20and%20Settings/Florian/Desktop/mockup/FlatClusters2page.html
Localhost WWW Private News Hardware University Journals template.php Samar: un passato, u...
XMem
Extended Memory Navigation
Done

```
History
  beer
  digital photography
  henry viii
    Henry VIII (http://tudorhistory.org/henry8)
    [henry viii]
    Sat Sep 10 2005
    The Six Wives of Henry VIII (http://tudorhistory.org/wives/
    [the six wives of henry viii]
    Sat Sep 10 2005
    Henry VIII Summary (http://www.allshakespeare.com/henry
    [henry viii summary, william shakespeare]
    Sat Sep 10 2005
    Henry VIII Character Analysis (http://www.allshakespeare.c
    [henry viii character analysis, william shakespeare]
    Sat Sep 10 2005
    Henry VIII Henry (King Henry VIII of England) (http://www.
    [henry viii, king of england, character analysis]
    Sat Sep 10 2005
    Henry VIII Text (http://www.enotes.com/henry-viii-text/)
    [henry viii , full text, dramatis personae]
    Sat Sep 10 2005
    Henry VIII Text: Act I, Scene 1 (http://www.enotes.com/h
    [henry viii, act i scene i]
    Sat Sep 10 2005
    William Shakespeare at eNotes (http://www.shakespeare.c
    [William Shakespeare at eNotes]
    Sat Sep 10 2005
    Henry VIII Summary (http://www.allshakespeare.com/henry
    [henry viii summary, william shakespeare]
    Sat Sep 10 2005
    Henry VIII Character Analysis (http://www.allshakespeare.c
```

HENRY VIII

BORN: 28 JUNE 1491
SUCCEEDED: 21 APRIL 1509
DIED: 28 JANUARY 1547

"If a lion knew his strength, it were hard for any man to hold him."
- Sir Thomas More of Henry VIII

More about Henry VIII at this site:
Henry VIII gallery page 1
Henry VIII gallery page 2

See also:

Prince Henry

Henry Tudor, named after his father, Henry VII, was born by Elizabeth of York June 28, 1491 in Greenwich Palace. Since he was the second son, and not expected to become king, we know little of his childhood until the death of his older brother Arthur, Prince of Wales. We know that Henry attended the wedding celebrations of Arthur and his bride, Catherine of Aragon, in November 1501 when he was 10 years old.

Figure 3: Flat xMem history: URLs are accessed through clusters. Keywords extend the URL descriptions.

3.1.2 Tasks

Subjects had to retrieve contents already visited on the Web. Their task was assigned in form of a written scenario, outlining the presumed past navigation actions. Based on this scenario, subjects were asked to retrieve, by means of the history mechanisms they were assigned to, a page showing contents about "Henry VIII". The scenario purposely did not provide complete indications about the time of visit of the page, so as to simulate the lack of memory along the temporal dimension.

3.1.3 Procedure

The experiment was conducted using a classical paradigm. Subjects were asked to retrieve the page about "Henry VIII" using one of the three mockup interfaces, depending on the group they belonged to. Each subject was assisted singularly during task execution for clarifying possible doubts or uncertainties concerning the scenario only. For each subject, the exact time interval from the start of the navigation within the history till the retrieval of the target page was measured. History navigations exceeding 10 minutes were considered unsuccessful.

At the end of the page retrieval task, subjects were required to fill in questionnaires on user's satisfaction and browsing experience.

3.2 Data Analysis

In order to check whether our experimental hypotheses were verified, we performed a cross-comparison of the collected data for the three groups. An ANOVA test performed on the completion times showed a significant difference for the means of the three groups ($F_{(44)} = 3.248, p < .049$). Furthermore:

- A t-test showed a significant difference in retrieval time between the TH and the HX group in favour of the xMem history organization ($t_{(28)} = -3.073, p < .005$).

- A t-test comparing the performance of group HX and of group FX did not show any significant difference ($t_{(28)} = -1.521, p < .139$).

- A further t-test comparing the performance of TH users and FX users did not produce a significant difference ($t_{(28)} = -.927, p < .362$).

The values of the means of the retrieval times for the three groups ($TH = 116sec., HX = 64sec., FX = 93.3sec.$) show that the average page retrieval time for the group TH is almost twice the value for the HX group, while the FX group only slightly improves the retrieval time.

The general satisfaction was expressed by users on a 5-point scale of (1 = *very negative*, 5 = *very positive*). The satisfaction mean values ($TH = 2.67, HX = 3.67, FX = 3,60$) showed a significant difference between both the TH group and the HX group ($t_{(28)} = 2.617, p < .016$) and the TH group and the FX group ($t_{(28)} = 2.514, p < .020$).

From the previous experimental results, it seems possible to claim that the hierarchical order produces a significant effect on the retrieval of pages when users do not exactly remember the time in which the page has been visited. Conversely, our expectations that a simple semantic enrichment, such as the one provided by the flat history prototype, would improve page retrieval times, has not been totally confirmed. Indeed, the flat history organization, contrarily to the hierarchical xMem history, does not introduce any significant improvement with respect to the traditional paradigm. Moreover, the mean for the FX group suggests a deterioration of the users' performance with respect to the HX group.

Even data about the recalling performance of keywords, derived from the post-experiment questionnaire, indicate that none of the subjects in the FX group was able to remember a keyword. This can suggest that the keywords used to describe the URLs in the FX prototype did not receive attention by subjects. As a consequence, it might be said that the performance on the experimental task for the FX group seems not to depend on the additional semantic characterization of keywords, but rather on the access path through which this information is made accessible to end users.

In more detail, the results of the t-tests reported above are consistent with the experimental hypothesis 1, since they indicate that in general subjects' performance improves when

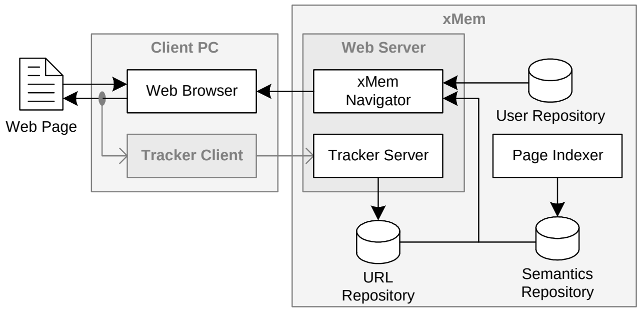


Figure 4: xMem functional Architecture.

using enriched histories. Results also suggest a clear and significant advantage of the HX group over the TH group. A possible reason for this result is that, when the page access time is not exactly known, the user needs to scan the whole URL list, while holding in the working memory one of the already scanned items, temporarily considered the most pertaining. From a cognitive perspective, this implies that a high amount of cognitive resources are spent on scanning, matching and judging the remaining results in the list. In addition, the demand for cognitive resources needed to carry on the comparison and evaluation activities increases as the information elaboration process goes on. This results in a competition between the cognitive resources needed to maintain information and those needed to elaborate it.

The reason of the better performance of the HX group lies therefore in the classification carried by clusters and keywords, which enables a more efficient information processing from the cognitive point of view. As better discussed in [18], we also observed that the occurrence of the better performance of the HX group does not depend on the subjects' expertise on the knowledge domain ($r = -.078; p < .782$), nor on their expertise on history mechanisms ($r = -.078; p < .783$).

Supported by the previous results, we have decided to adopt the hierarchical organization for visualizing the enriched history.

4. DETAILED DESIGN OF XMEM

xMem consists of several components that share the same data source. The implementation of the correspondent database depends on the expected load at runtime and can consist either in a single database on the xMem server itself or in a freely distributed server architecture. The functional architecture of the xMem tool is primarily influenced by two goals of our approach: (i) adopting a remote logging mechanism for (ii) providing online access to logged data. Remote logging builds on the client-server paradigm. Online log access requires splitting the overall architecture into two logical components, one for each communication direction. Figure 4 graphically depicts the resulting functional architecture, roughly divided into *Client PC* and *xMem Server*.

The *Tracker Client*, installed at the client side, is in charge of tracking navigated URLs and of transmitting them to the *Tracker Server*. The Tracker Client also allows activating/deactivating the tracking mechanism. On the other side, the Tracker Server is responsible for feeding the incoming messages into the *URL Repository*. For each registered user, such repository contains the actual log data in form of

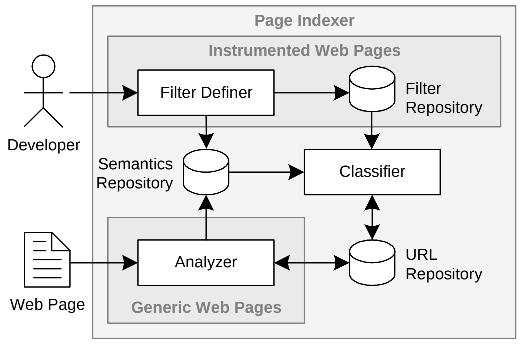


Figure 5: Defining semantic classes for Web sites.

URL strings of the visited Web pages. User data are maintained in a *User Repository*, which stores psychographic information and user preferences. These data are the basis for managing access rights over history data.

By means of the so-called *xMem Navigator* Web application, users can then access their personalized navigation history over the Internet and browse logged data by means of keywords that are representative for the visited contents. Such keywords are stored in the *Semantics Repository*. Their extraction is managed by the *Page Indexer* module, described in the following.

4.1 Semantic Enrichment of History Data

The xMem *Page Indexer* is in charge of enriching tracked history data. More precisely, page keywords are gathered along two mechanisms, implemented through some *Page Indexer* sub-modules. One is based on filters provided by application designers, which allow associating descriptions to pages, based on the analysis of the URL strings. We call pages coupled with such description *instrumented Web pages*. When filters are not provided at design time, a second mechanism automatically extracts keywords from *generic Web pages* by means of syntactic extraction techniques.

Figure 5 graphically describes the sub-modules for defining or extracting keywords. The *Filter Definer* and the *Classifier* module are respectively used for defining filters and using them for URL classification. The *Analyzer* module is instead used for extracting keywords from generic Web pages. The following sections provide details about the two xMem components.

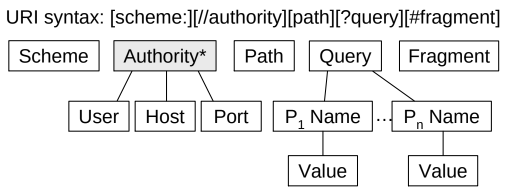
4.1.1 Explicitly Providing Keywords

Instrumented Web pages require application developers to subscribe to the xMem project. Subscription means providing, through an appropriate Web front-end, a description of the application, which can serve the purpose of classifying pages. In particular, the *Filter Definer* allows developers to input a set of keywords characterizing the most relevant application pages. These keywords are stored in the *Semantics Repository* as categories for page classification. Through the Filter Definer, developers also define URL filters for associating keywords and pages. A dedicated *Filter Repository* contains the set of filters defined for each application.

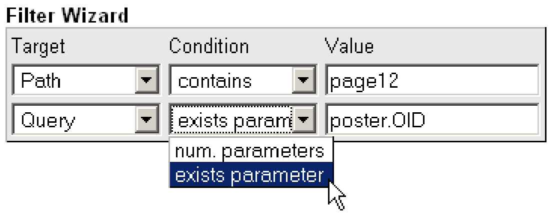
According to their scope within xMem, we distinguish between global filters and local filters:

- *Global filters* check whether tracked URLs belong to one of the subscribed applications. Their scope is global within xMem and thus they are applied to all tracked requests. Suitable global filters are automatically created during the subscription of an application.

- Once a URL has been associated to a particular application through a global filter, *local filters* are applied for performing the association of the page to one of the keywords previously provided by the application developer.



(a) Decomposition of URLs into their constituent parts. The component *Authority* is accessed through its subcomponents.


Filter Wizard

Target	Condition	Value
Path	contains	page12
Query	exists param	poster.OID

(b) Example Filter Wizard at work.

Figure 6: Decomposing URL strings and defining rules over its components.

Since several URLs could correspond to the same concept, several filters might be defined for the same keyword. Filters consist of one or more rules and one association. For performing the association between URLs and semantic classes, all rules of the filter must evaluate to true. A rule is a triple $< target, condition, value >$, where:

- *Target* elements are the portions in which a URL can be decomposed, on which filters are applied. Figure 6(a) graphically illustrates the applied decomposition of URLs.

- *Conditions* compare URL targets with values. A *Filter Wizard* provides an interactive user interface for defining filter conditions (see 6(b)).

- *Values* are provided by designers. They consist of actual values against which targets must be compared, as established by conditions.

The so defined filters are translated at runtime into suitable regular expressions, one for each filter condition. A *Classifier* module then runs the filters over the tracked URLs for constructing associations between URL's in the *URL Repository* and semantic classes or keywords of the *Semantics Repository*. It therefore acts as an interpreter of syntactic filters.

4.1.2 Automatically Deriving Keywords

Web pages not provided with filters are re-materialized by the *Analyzer* module (see Figure 5), which parses the respective HTML code for automatically extracting significant keywords. Such keywords represent further *semantic classes* stored in the *Semantics Repository*.

The keyword extraction algorithm developed so far is based on pure syntactic heuristics, considering text enclosed in some relevant HTML tags (e.g. `<TITLE>`, `<H1>`, `<B>`, ...). Each tag t is associated with a weight w. Tag weights refer to a scale of importance (from 1 to 100). For example, the `<TITLE>` or `<META>` tags provide keywords with the highest importance, while plain text within the `<BODY>` tag has less relevance, since a page title has more chances to provide significant keywords than plain text. We thus order tags based on their weight, assigning weight $w = 1$ to non-formatted text inside the page's `<BODY>` tag and higher values of w to more important tags.

As illustrated in Figure 7, keyword extraction proceeds along the following activities:

- *Parsing*, for transforming the HTML code into a well-formed XML document and eliminating irrelevant tags (e.g. `<SCRIPT>`, `<STYLE>`,...).

- *Keyword extraction*, for identifying keywords based on their frequency within the page content. Keyword frequency for a page p is defined as

$$f_p(k) = \frac{n}{N},\ 0 < f_p(k) \leq 1$$

  where n is the number of occurrences of k in p, while N is the total number of words in p.

- *Stemming*, for removing keywords suffixes checking for keyword equivalence. Stemming is based on the Krovetz stemming algorithm [14].

- *Keyphrase extraction*, for identifying combinations of different keywords forming meaningful phrases. This activity introduces some novelties with respect to other methods for keyword extraction, since it also supports the extraction of keyphrases of up to 5 words. Several keyword extraction tools just consider up to three words (see for example, [11]).

- *Removal of stop-words*, for eliminating keyword corresponding to common words (i.e., articles, prepositions) that are useless to index a page.

- *Relevance assignment* to keywords and keyphrases, based on weights associated to HTML tags and the relative frequency of keywords and keyphrases within tags. Be $f_{p|t_i}(k)$ the *within-tag frequency* of keyword k restricted to tag t_i in page p. The relevance R of the keyword k for a page p is then a percentage value defined as

$$R_p(k) = \sum_i f_{p|t_i}(k) * \frac{w_i}{W} * 100, 0 < R_p(k) \leq 100$$

  where w_i is the weight associated to t_i, and $W = \sum_i w_i$. Relevance values are used for selecting only the n (e.g. 5) most relevant keywords.

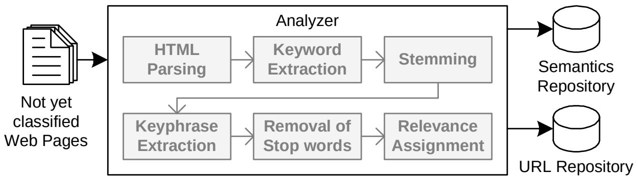


Figure 7: The Analyzer module in detail. Keyword extraction involves several refinement tasks.

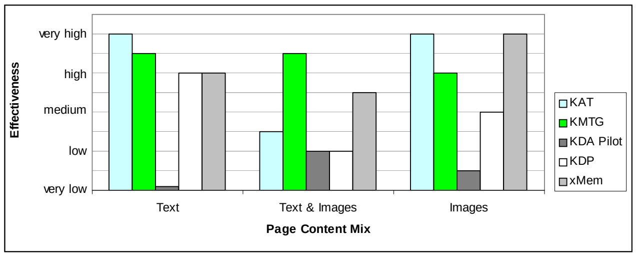


Figure 8: Comparison analysis of the algorithm performance.

4.1.3 Comparison with other Tools

The effectiveness of our algorithm for keyword extraction has been evaluated through a comparison with other keyword-based ranking algorithms, by addressing the capability to find relevant keywords matching page contents. More specifically, four algorithms have been considered [10, 13, 16, 12].

The evaluation has been conducted as follows. Six end users have been asked to analyze six different Web pages and extract five keywords judged as the most relevant. Pages have been chosen applying three criteria: (a) pages with a prevalence of text; (b) pages with a prevalence of images; (c) pages that balance text and images. The keywords extracted by the users represented the baseline on which to compare the algorithm performance. The algorithm effectiveness has been evaluated as the number of keywords matching the end users' keywords. A 5-level scale has been elaborated for quantifying the performance: from *very low* (indicating that the algorithm matches from 0% to 20% of users' keywords), to *very high* (positive matches about 80% to 100%).

The results of the comparison analysis is shown in Figure 8. Only the commercial KMTG tool exhibits a better performance than our prototype implementation. The xMem algorithm is particularly effective, when analyzing pages with a prevalence of images (see Figure 8), this is due to the high weights assigned to the `alt` attribute of the `<IMAGE>` tag and to the `<TITLE>` tag.

4.1.4 Clustering Keywords

Keyword definition by means of the *Filter Definer* and automatic extraction by means of the *Analyzer* are followed by a clustering activity, which allows grouping keywords into clusters representing high-level classes.

A cluster $C = \{p_1, ..., p_n\}$ is defined as a collection of pages, which have similar keywords. A page vector over all known keywords is computed for each page with a TF-IDF ranking, a common transformation, which reflects both the frequency of words within a specific page, as well as their overall frequency with respect to the whole page set. The classical TF-IDF formula has been modified by considering our *within-tag frequency* of keywords, as defined in Section *4.1.2*. Similarity is computed by means of cosine similarity. Clustering proceeds as follows:

- Each cluster is initialized with a *seed page*, i.e., a Web page visited by a user, and is then expanded using similarity among page keywords.
- During expansion, new documents are added to clusters, if they have the same keywords of documents already in the cluster, or if their keywords are synonyms or hyponyms. Synonyms and hyponyms are determined by means of the WordNet dictionary [5].
- During cluster reduction, documents showing a low similarity with respect to the other cluster documents are removed. Expansion and reduction iterate until the clusters of step *n* are the same as those of step *n-1*.
- After the clusters have been defined, cluster names are derived from their most frequent keywords. Even in this case, WordNet is used for solving synonyms and hypernyms.

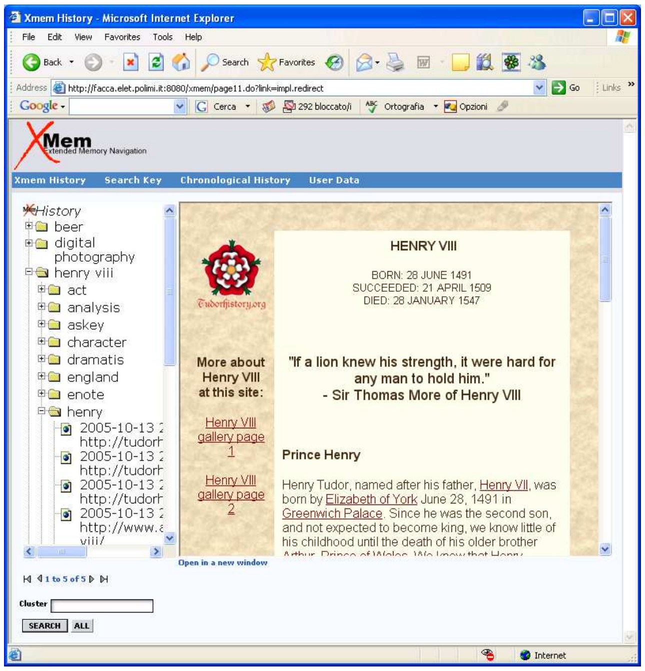
5. IMPLEMENTATION

Figure 9 shows a screenshot of the xMem Navigator's Web interface that allows browsing the user's navigation history, according to the enriched history organization discussed in Section 3. This front-end allows registered xMem users to access history data by chronological order, as well as by means of a cluster and keywords hierarchy. Also, a keyword-based search is possible.

The xMem *Tracker* is implemented as HTTP sniffer at the client side and as Java servlet at the server side. The sniffer application has to be installed on the client PC, and can be controlled by means of a small tray icon in the operating system's taskbar. The URL tracking process can be started and stopped at will; starting it requires the authentication of the respective user through *username* and *password*. At the server side, the Tracker server module, which feeds the tracked URLs into the application's data source, works and in parallel to the Navigator application.

The *Classifier* module is programmed in Java, and makes use of regular expressions for filter evaluation. The *Analyzer* module (also implemented in Java) operates independently in parallel to the Web application and analyzes all those pages that are not yet covered by filters.

The application has been deployed as Web application on top of a J2EE/Struts platform, and can be accessed by authenticated users by means of standard Web browsers. The implemented tracking process is completely decoupled from the execution of the Web front-end and implements an asynchronous communication mechanism with respect to the rendering of Web pages. This allows a flexible runtime management of the single xMem modules and reduces possible overheads during page computation. Together with the modular architecture of the xMem system, the asynchronous communication mechanism allows for highly scalable configurations of the overall system, so as to cope with a high number of user sessions.


Xmem History - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://facca.elet.polimi.it:8080/xmem/page11.do?link=impl.redirect
Go Links
Google Cerca 292 bloccato/i Ortografia Opzioni
XMem Extended Memory Navigation
Xmem History Search Key Chronological History User Data
History
beer
digital photography
henry viii
act
analysis
askey
character
dramatis
england
enote
henry
2005-10-13 2 http://tudorh
2005-10-13 2 http://tudorh
2005-10-13 2 http://tudorh
2005-10-13 2 http://www.a viii/
Tudorhistory.org
Open in a new window
1 to 5 of 5
Cluster
SEARCH ALL
Internet

HENRY VIII

BORN: 28 JUNE 1491
SUCCEEDED: 21 APRIL 1509
DIED: 28 JANUARY 1547

More about Henry VIII at this site:

Henry VIII gallery page 1

Henry VIII gallery page 2

"If a lion knew his strength, it were hard for any man to hold him."
- Sir Thomas More of Henry VIII

Prince Henry

Henry Tudor, named after his father, Henry VII, was born by Elizabeth of York June 28, 1491 in Greenwich Palace. Since he was the second son, and not expected to become king, we know little of his childhood until the death of his older brother

Figure 9: Screenshot of the xMem Navigator tool. At the left hand side users can navigate their personalized history, while at the right hand side they have the possibility to show a preview.

6. RELATED WORK

Few researchers have considered new history mechanisms and most of them in the research context of new browser metaphores.

IBM's Web Browser Intelligence (WBI) tool [1] is a browser with some personal history functions based on the annotation of hyperlinks on all Web pages with traffic signals, and this performs well for remembering visited pages, providing a keyword search through the text of pages already visited, noticing patterns in the Web browsing behavior and suggesting shortcuts, and automatically checking favorite Web pages for change.

WebTOC [15] automatically creates TOC (table of contents) frames for sets of Web pages. The TOC frame can be quite useful, and enables a more dynamic presentation of the desired information. Its main drawback is that it occupies a large portion of the screen.

WebNet [4] is a browser extension that displays a graphical representation of the hyperspace being explored. This is performed dynamically and independently from the content provider, across many sites. It is a challenge, however, to present the graph in such a way that the contextual information is highlighted.

DeckScape [2] is an experimental browser based on the concept of *deck*. Each deck is a linear stack of pages that the user can leaf through. As with history mechanisms, if a user starts from page A and goes to B, B is added to that stack or deck. Unlike history mechanisms, if the user goes back to A and then traverses a link to C, B is not lost; it remains in the deck of pages. However, users are cognitively loaded with the responsibility of maintaining pages logically in different stacks unless decks are pruned regularly.

Recently, [6] the Microsoft research center has worked on *Stuff I've Seen* (SIS), which is able to support users in retrieving and reuse information already seen locally on the PC. The system aims at facilitating information seeking by providing an index of information that a user has seen (email, Web page, document, appointment) and, in addition, a set of rich contextual cues about the searched information made available from the previous accesses.

Personalized Search is an improvement to Google, currently under experimentation. In particular, Google offers a *personalized history search* that enables users to view and manipulate their history of searches. Users might search in their past interactions with Google; they may search history *by Web* and/or *by images*; they may pause the history (this means that the services will not collect any history until users choose to resume); they may bookmark the search results displayed in the history list.

All solutions described above respond to the need to record users' navigational history for allowing the successive re-access of visited pages. xMem works beyond these mechanisms by analyzing and interpreting the structure of recorded URLs and by making available the results of this process to end users.

7. CONCLUSIONS

We have argued that current browsing functionalities do not adequately support retrieving information from a user's navigation history. xMem improves history mechanisms by making use of new criteria to organize and show the navigational history instead of simply exploiting time-sorted history mechanisms that prevail today. As also demonstrated by a controlled experiment, this retrieving strategy makes history navigation easier and more effective, because it provides a characterization of the context in which information has been seen.

We are currently experimenting and improving the xMem project along several directions. First of all, we are further verifying the effectiveness of our approach by investigating whether chronological access is more effective for the short-term history (e.g., this week's pages), thus restricting the semantic dimension only to the long-term history. We are also planning a further improvement of the user experience by means of collaborative filtering techniques for semantics sharing among xMem users and by means of a cooperative interaction paradigm, in which users can also refine the automatically derived indexing and clustering terms.

The automatic approach that we have adopted for achieving history enrichment is based on syntactic criteria for extracting keywords from page contents. As such, our description of pages by means of keywords has some limitations with respect to the real semantics of page contents, that would be better described by means of higher-level concepts not always achievable when using pure syntactic extraction techniques. For this purpose, we are planning to investigate the use of ontologies for augmenting the level of abstraction of our keywords and clusters and for highlighting also possible relationships existing among keywords. We are also experimenting the possibility of interfacing search engines for the provisioning of keywords.

Finally, we are further validating the efficiency of the clustering algorithm, by applying it to large data sets and by considering the adoption of incremental techniques.

8. REFERENCES

[1] R. Barrett, P. Maglio, and D. Kellem. Web Browser Intelligence: Opening up the Web. In *Proc. of COMPCON'97*, 1997.

[2] M. Brown and R. Shillner. Deckscape: An experimental Web Browser. Proc. of WWW'95: Technology, Tools and Applications, April 1995.

[3] L. Catledge and J. Pitkow. Characterizing Browsing Strategies in the World-Wide Web. Proc. of WWW'95: Technology, Tools and Applications, April 1995.

[4] A. Cockburn and S. Jones. Which way now? Analysing and easing Inadequacies in WWW Navigation. *Int. J. of Human-Computer Studies*, 45(1):105–129, July 1996.

[5] Cognitive Science Laboratory, Princeton University. WordNet - A lexical Database for the English Language. `http://wordnet.princeton.edu/`, 2005.

[6] S. Dumais, E. Cutrell, J. Cadiz, G. Jancke, R. Sarin, and D. Robbins. Stuff Ive seen: A System for Personal Information Retrieval and Re-Use. In *Proc. of SIGIR03, July 28 August 1, 2003, Toronto, Canada*, 2003.

[7] GRSoftware. Grsoftware Keyword Density Analyzer. `http://www.grsoftware.net/search_engines/software/grkda.html`, 2005.

[8] GVU's WWW Surveying Team. GVU's 10th WWW User Survey. `www.cc.gatech.edu/gvu/user_surveys/survey_1998_10/graphs/use/q62.htm`, 1998.

[9] E. Kandogan and B. Shnaiderman. Elastic Window: A hierarchical multi-window World Wide Web Browser. In *Proceedings of UIST'97*. ACM, 1997.

[10] KAT. Keyword Analysis Tool. `http://www.webmaster-toolkit.com/keyword-analysis-tool.shtml`, 2005.

[11] KDA. Keyword Density Analyzer. `http://www.keyworddensity.com/`, 2005.

[12] KDP. Keyword Density & Prominence. `http://www.ranks.nl/tools/spider.html`, 2005.

[13] KMTG. Keywords Meta Tag Generator. `http://www.hermetic.ch/kwg/kwg.htm`, 2005.

[14] B. Krovetz. *Word Sense Disambiguation for large Text Databases*. PhD thesis, Department of Computer Science, University of Massachusetts Amherst, 1995.

[15] D. Nation, C. Plaisant, G. Marchionini, and A. Komlodi. Visualizing Websites using a hierarchical Table of Contents Browser: WebTOC. Proc. of Human Factors and the Web'97, Denver, June 1997.

[16] Nehuen Multimedia. KDAPilot Professional for Windows. `http://www.nehuenmultimedia.com.ar/html/kdapilot.html`, 2005.

[17] D. Newfield, B. Sethi, and K. Rayll. Scratchpad: Mechanisms for better Navigation in directed Neb Searching. In *Proc. of UIST'98*. ACM, 1998.

[18] F. Rizzo, F. Daniel, M. Matera, S. Albertario, and A. Nibioli. Evaluating the Semantic Memory of Web Interactions in the xMem Project. In *Proc. of AVI'06*. ACM, 2006.

[19] L. Tauscher and S. Greenberg. How People revisit Web Pages: Empirical Findings and Implications for the Design of History Systems. *Int. J. Human Computer Studies*, 1997.