Web-R: a tool to record & replay personal web navigation

Jean-Daniel Kant, Alain Lifchitz

Web-R: a tool to record & replay personal web navigation

Jean-Daniel Kant
LIP6 (Laboratoire d'Informatique - Université Paris 6)
8, rue du capitaine Scott
75015 Paris, France
33 1 44 27 88 05
jean-daniel.kant@lip6.fr

Alain Lifchitz
CNRS-LIP6
8, rue du capitaine Scott
75015 Paris, France
33 1 44 27 43 32
alain.lifchitz@lip6.fr

ABSTRACT

This poster presents a useful tool to capture the content of browsing sessions. *Web-R* saves systematically all the components sufficient and necessary to visualize offline the page, the same way it was displayed online. In order to avoid the saturation of local disk, *Web-R* integrates a page comparison mechanism to avoid unnecessary redundancy and, in addition, provides a way to manage the maximum storage space. We give some insights than, in typical cases, this storage is compatible with today's casual hard disk capacities and costs, along years. Moreover, since the full content of visited pages is stored, the *Web-R* user can access and process (*e.g.* sort, extract) several information including path and visit statistics, therefore giving him/her a global view on his/her personal navigation.

Keywords

Personal Web navigation, Personal Web archive, Web history search, Web accessibility, Offline browsers, Tools and techniques for Web personalization, Web visualization, Information management tool.

1. INTRODUCTION

In this work, we want to design a companion recording tool of the browser, thus integrating navigation, browsing and recording, whether they occur online or offline. We propose to save automatically the pages the user already browsed in order to help this user to organize his/her own Web. In order to be effective, the recording tool has to be non-intrusive (Figure **1 below**), fast and should provide space management services in order to avoid possible local disk saturation. It should also provide an exact and complete storage of user's web pages and web navigation, by saving all the components sufficient and necessary to visualize offline the page the same way it was displayed online.

Figure 1. *Web-R* root interface (actual scale)

Most of existing tools (online and/or offline browsers [**5**], site rippers [**6**], recorders [**1**]) do not fulfill our requirements since they are usually based on URL recording that cannot capture the dynamic and sometimes transient nature of web pages. Moreover, these approaches lack automation and user-centric (no adaptation, works only at request since the user has to ask to record the page each time he/she browses it).

2. DESIGN

Web-R is an application that will take control of its child running instance of *Microsoft Internet Explorer*® (*IE*) using *OLE Automation*®. Only the instances that have been created through *Web-R* will be monitored.

2.1 Page recording

The main architecture is depicted in Figure **2 below**.

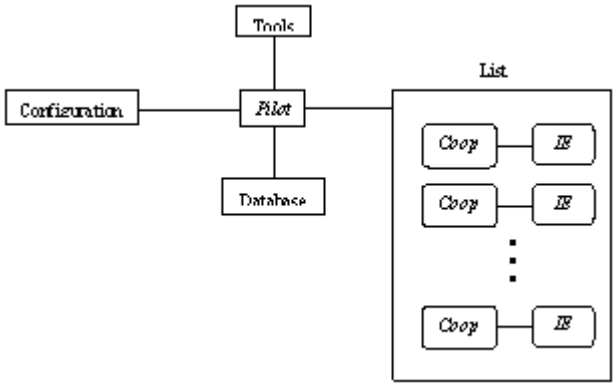

Figure 2. Overall *Web-R* architecture

The *Pilot* module is the heart of *Web-R*. It controls the configuration, tools and database modules, and a list of *IE* instances. Each *IE* instance in this list is coupled with a Coop process, which is used to communicate between this instance and *Pilot*. *IE* instances are independent, so the user could browse several pages simultaneously.

The page recording proceeds as depicted in Figure **3 below**. The *IE* instance retrieves the page from the Web (1), and interprets its code for display. While the page is downloaded by *IE*, the *Coop* module monitors this download process, and waits until *IE* has finished its processing (2). *Coop* recuperates the result of this processing (*i.e.* the processed page) and sends it to the *Page Source Processor* (PSP) module (3). The aim of this module is to perform all the additional processing that is necessary to ensure a perfect storage of the current page. Therefore, the PSP might collect additional files (*e.g.* images, sounds, *Flash Macromedia* code, etc.) from the Web (4). Once the page processing is fully completed, the page and its components are sent to the *Pilot* module (5) to store all the files on the database (6).

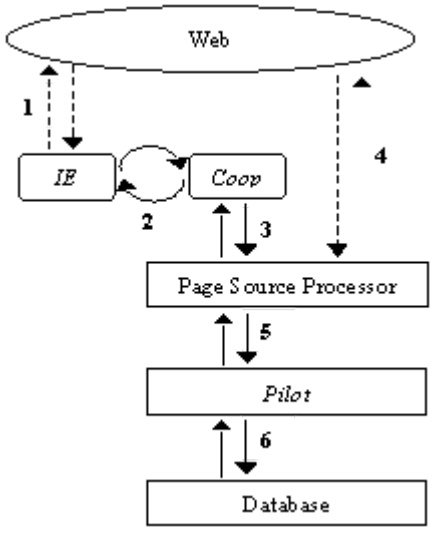

Figure 3. Page recording flowchart

The main task of the PSP is to handle properly file name redirection of live Web to local hard drive. It handles the complete storage of frames, images, secured pages, animations (*e.g. Flash Macromedia*®), scripts and dynamic pages

(*e.g.* ASP, PHP, CGI, etc.).

2.2 Storage

2.2.1 Page comparison algorithm

In order to avoid redundancy, we implemented a simple surface comparison algorithm to determine if the current page already exists in the database. This algorithm proceeds in 3 steps to determine if a page P needs to be stored:

1. **Let *SP* a set of stored page with the same URL as *P*. If *SP* is empty, store *P*;**
2. **Otherwise, look for a page $P' \in SP$ with the same HTML code as *P*. If none, store *P*;**
3. **Otherwise, compare respective individual size of the components in P' and *P*. If they differ, store *P*; otherwise do not store *P*.**

This algorithm is fast and will work for most of the situations encountered during realistic web navigation.

2.2.2 Storage space management

Since all visited pages will be stored on local disk during a *Web-R* session, one may be concerned about space on local disk. In fact, we believe that the total size of a personal web content will not exceed 5 GB / year for a typical user. Recent studies confirm this estimation ([4]). Anyway, we must provide a tool to manage storage space. Thus, the *Web-R* user can set a size limit of the storage space that could be none, a fixed size or a percent of the total disk size. When the limit is reached, a purge mechanism is started, which could be either manual or automatic.

2.3 Managing the personal web

In the current version, *Web-R* also offers a basic tool to manage the stored pages. When started, it displays a multi-column list window that contains all the stored pages. The page entries can be sorted / filtered by the following attributes: URL, title, number of access, date and time of first visit, date and time of last visit, logical / physical size. The managing tool could also be used to visualize a stored page. Its (offline) content will be displayed when the user double-click on one item of the page list.

3. CONCLUSION AND FUTURE DIRECTIONS

In this work, we have shown that a systematic storage of the personal web is technically feasible and realistic since it will not require too much disk space.

Two very recent systems have very similar features to *Web-R*. *Keepoint* [2] also provides an automatic and complete storage of web pages browsed by IE but does not integrate a page comparison mechanism hence requiring the user to decide whether a similar page should be overwritten. The *Microsoft WebScout* [3] project seems to be very ambitious and its *HistoryExplorer* component shares the same goals as *Web- R*. However, we do not have any details on this implementation, so it is difficult to compare it with our proposed solution.

Web-R is currently under development and several technical issues need to be addressed. In the short run, we need to consolidate the page processor in order to solve the cases where the storage was not completed (typically dynamical pages and scripts). Moreover, we will facilitate offline navigation, where links to already stored pages (local) and links to non-stored pages (exterior) will be distinguished. In the latter case, the system may ask the user if he wants to locally store the exterior page or navigate this page online. We also plan to make a better use of the browser resources, like using the cache to avoid multiple - and therefore unnecessary - fetches of components on the live Web. We should also compress the stored files to optimize space storage, and encrypt all the data to ensure a total privacy.

4. ACKNOWLEDGEMENTS

We would like to thank the LIP6 for its financial support on this project. We also thank Nicolas Limare for his participation to the development of the preliminary version of the *Web-R* software.

5. REFERENCES

1. **Anupam, V., Freire, J., Kumar, B. and Lieuwen, D. Automating Web Navigation with the WebVCR, in. Proceedings of WWW9 (Amsterdam (The Netherlands), 15-19 May 2000). http://www9.org/w9cdrom/208/208.html**
2. ***Keepoint*. http://www.keepoint.com**
3. **Milic-Frayling, N., Sommerer, R. and Tucker, R. MS WebScout: Web Navigation Aid and Personal Web History Explorer. In Proceedings of WWW2002 (Honolulu, (Hawaii, USA), 7-11 May 2002). http://www2002.org/CDROM/poster/170/index.html**
4. **Nielsen//NetRatings. http://www.nielsen-netratings.com/hot_off_the_net.jsp**
5. ***SurfSaver*. http://www.surfsaver.com.**
6. ***Wget*. http://www.gnu.org/directory/wget.html.**