An Approach to Persistence of Web Resources

Joachim Feise
Department of Information and Computer Science
University of California, Irvine
Irvine, CA 92627-3425, USA
Tel: +1-949-824-7308
E-mail: jfeise@ics.uci.edu

ABSTRACT

The growth of the World Wide Web holds great promise for universal online information access. New information is constantly being made available for users. However, the information accessible on the Web changes constantly. These changes may occur as modifications to both the content and the location of previously existing Web resources. As these changes occur, the accessibility to past versions of such Web resources is often lost. This paper presents an approach to provide content persistence of Web resources by organizing collections of historical Web resources in a distributed configuration management system to allow online, read-only access to the versioned resources.

KEYWORDS: WWW, Persistence, Versioning

INTRODUCTION

The pervasive nature of the World Wide Web promises to bring immediate access to the world's knowledge to every user of the Web. This immediacy of access presents the Web as an information collection that exists in the present and hides aspects of the information collection that change over time.

Web resources can be described as temporal objects that come into existence, exist through time, and ultimately go out of existence. Access to the Web permits some of this temporality to be captured. For instance, URLs provide the ability to locate a resource that may not have come into existence, that may exist, or that may no longer exist. But this ability only addresses the location of Web resources, which has been well explored (see, e.g., [3], [4]) and is not covered here. I am interested in accessibility to the changing content of Web resources. Access to past content of Web resources can be of great value for a variety of historical and legal reasons, both in society at large and within organizational boundaries.

The approach described in this paper utilizes the capabilities of a configuration management system to capture relationships between the collected resources. An earlier report on this work was presented in the 2000 Open Hypermedia Systems Workshop[1].

COLLECTING WEB RESOURCES

Persistence of Web resource content provides a means to access collections of information that change over time. The first step to persistence is capturing the states of resources as they change over time. Collecting the content of Web resources over time allows a series of snapshots of the resources to be maintained and accessed.

Traditional Web Resource Collection Process

The process of collecting Web resources is well understood and is used on a daily basis by Web search sites like Google [2] as well as by organizations like the Internet Archive[6]. However, Web search sites only utilize the last resource revision collected. Access to previous revisions is not possible. The Internet Archive, on the other hand, permanently stores all collected resources. The size of this collection and the aggregation of unrelated Web resources in large files facilitate offline access and analysis, but makes online, real-time access impractical. To provide online access to collections of Web resources, a different approach is therefore necessary.

Resource Collection Organization for Online Access

Organization and management of resources is generally the task of configuration management systems (CMS). A wide variety of resource relationships can be modeled by these systems. For the purposes of providing Web resource content persistency, the versioning aspect of a CMS is of importance. Versioning provides the organization of resource changes over time, while allowing easy access to all resource revisions. CMS therefore seem to be a natural fit for the task of providing fast online access to multiple versions of resources collected from the Web.

I decided on using the Network Unified Configuration Management System (NUCM[7]) for the storage and organization of the versioned resources, since NUCM is designed to provide transparent distributed storage capabilities which facilitates scalability of the approach.

PERSISTENT WEB RESOURCE ACCESS

The access to the persistent Web resources, once collected, should be as transparent to the user as possible. The easiest way to accomplish this goal is to configure the user's Web browser to use a proxy server. The proxy server has the task of distinguishing between requests for normal, current Web resources and requests for historical revisions of Web resources. The proxy server can also be used to automatically store the latest version of the resources the users are viewing, thereby easing the task of keeping the resource storage up-to-date.

Client Implementation

To access Web resources maintained by the CMS, the user's browser has to send special requests to the proxy server. These requests include the resource URLs, the date and time of interest, and the revision number of the requested resource. These additional requirements should not be the cause for a large change in the user's browsing experience.

The approach I chose is based on the use of frames and JavaScript in the Web browser. In my implementation, the browser displays two frames, a navigation frame and a resource frame. The navigation frame contains several dialog elements to select the Web resource of interest, a date of interest, or a specific revision of the resource. The navigation frame allows easy navigation between revisions and dates.

Proxy Server Implementation

Once the user's Web browser is set up according to the discussion above, a proxy server is utilized to receive the user input and provide the user with the appropriate resources. I chose the Squid proxy server[5], because the availability of its source code allowed me to implement the functionality required to utilize the NUCM CMS to store and retrieve the requested revisions of Web resources.

As long as the user does not utilize the navigation frame, the proxy server performs its standard processing, i.e., for each incoming request, it delivers the requested resource either from its cache or forwards the request to the original Web site or another proxy server. Before the proxy server delivers the requested resource to the user's Web browser, however, my implementation intercepts the resource and stores it in the CMS with the current date and time. This storage is transparent to the user. It provides the advantage of building a collection of persistent Web resources that is useful to the user base utilizing that particular proxy server.

A different processing path is chosen if the user selects a specific date or revision in the navigation frame in the browser. The proxy server interprets the request to determine the user's selection. The proxy server then locates the requested revision of the resource in the NUCM CMS and delivers it to the browser.

LIMITATIONS OF THE APPROACH

Since the resource URLs are used to determine the storage locations within the NUCM CMS, movements of resources over their lifetime results in breaking the revision histories of the resources within the storage. Resources that are deleted at a certain point in time present another problem, since no knowledge about deletions exists.

Storage of arbitrary Web resources and creation of revision histories of these resources could run afoul of copyright laws. Ways have to be developed to restrict access to sensitive resources.

EXPERIENCES

I have implemented a prototype to show the feasibility of the approach and perform first scalability tests.

Tests with the prototype showed that access to a versioned resource is in the under 1 second range, even if the NUCM database storing the resources is accessed over a LAN on a machine different from the proxy server. Access of current Web resources through the proxy server doesn't incur any noticeable delay.

ACKNOWLEDGEMENTS

I am greatly indebted to Jim Whitehead, Yuzo Kanomata and André van der Hoek for numerous discussions helping to shape the ideas presented in this paper.

This work was supported in part by a financial scholarship from Eagle Creek Systems, Inc.

This work was sponsored in part by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-00-2-0599. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

REFERENCES

1. Feise, J., Accessing the History of the Web: A Web Way-Back Machine, OHS-6, San Antonio, Texas, USA, May 2000, LNCS Vol. 1903, Springer-Verlag, New York, Heidelberg, http://www.ics.uci.edu/~jfeise/papers/wwbm.pdf
2. Google, http://www.google.com/
3. Lawrence, S. et al, Persistence of Web References in Scientific Research, IEEE Computer, February 2001, http://www.neci.nec.com/~lawrence/papers/persistence-computer01/persistence-computer01.pdf
4. Shafer, K. et al, Introduction to Persistent Uniform Resource Locators, INET96 Proceedings, Montreal, Canada, June 1996, http://www.isoc.org/isoc/whatis/conferences/inet/96/proceedings/a4/a4_1.htm
5. Squid Web Proxy Cache, http://www.squid-cache.org/
6. The Internet Archive, http://www.archive.org/
7. van der Hoek, A., A Reusable, Distributed Repository for Configuration Management Policy Programming, Ph.D. Dissertation, University of Colorado at Boulder, 2000, http://www.ics.uci.edu/~andre/research/papers/DISSERTATION.ps