

# Capturing Window Attributes for Extending Web Browsing History Records

Motoki Miura<sup>1</sup>, Susumu Kunifuji<sup>1</sup>, Shogo Sato<sup>2</sup>, and Jiro Tanaka<sup>3</sup>

<sup>1</sup> School of Knowledge Science, Japan Advanced Institute of Science and Technology

<sup>2</sup> Master's Program in Science and Engineering, University of Tsukuba

<sup>3</sup> Graduate School of Systems and Information Engineering, University of Tsukuba  
{miuramo,kuni}@jaist.ac.jp, {satosho,jiro}@iplab.cs.tsukuba.ac.jp

**Abstract.** We propose a method to enhance web browsing history records by considering browsing window attributes. The browsing window attributes means properties of web browser window such as size, location, z-order, activated status, etc. We often open multiple web documents for each window and browse the contents through handling these windows at the same time. Since a series of the window attributes describe the user's browsing activities, we exploit the attributes to enrich the browsing records to help the user find previously visited web pages.

## 1 Introduction

Most web browsers have a built-in history mechanism to help users find previously visited web pages. Cockburn et al. pointed out that approximately 81% of pages are previously visited by the user[3]. Therefore, there is a need to develop techniques and tools to efficiently revisit those web pages.

Basically, bookmarks are managed by the user, whereas the built-in history mechanism retrieves data without any explicit operations. Consequently, many built-in history mechanisms only use limited attributes such as the title, visited time, and visit counts. Furthermore, the history contains a large number of entries, even those made several weeks before, making the user struggle a lot to narrow down his/her search to a certain target page.

We consider the lack of attributes of entries to be one of the main reasons for struggling with many history entries. A history mechanism should provide richer attributes by attaching more contexts to the history entry. A richer attribute will help the user to efficiently filter the entries and find the target URLs.

## 2 Display-Oriented Attributes for Extending Web History

In this paper, we propose a method to enhance web browsing history records by considering browsing window attributes. By browsing window attributes, we mean the display-oriented properties of web browser window such as size, location, z-order and status of the window. We often open multiple web documents

and handle several windows while browsing the page contents. The browsing window attributes describe how the web documents were displayed and managed on a desktop screen. Therefore those attributes can be used to precisely infer the user’s browsing activities.

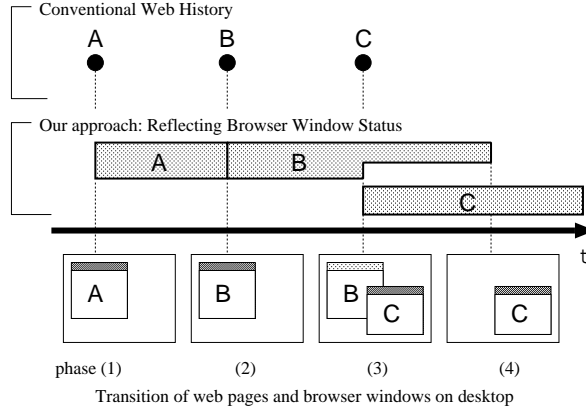


Fig. 1. Basic concept of our approach

## 2.1 Basic concepts

Our approach is originally motivated by a hypothesis that “the appearance of web documents shown on a desktop screen describes the significance of the document.” Figure 1 illustrates a basic concept of our approach comparing it with the conventional web history model. The four boxes below the time-line represent transition of desktop screen, and the inner box represents a browser window. In this scenario, each phase of the transition corresponds to the following actions: (1) the user opened page [A], (2) the user opened page [B] in the same window, (3) the user opened page [C] as a new window (the first window is deactivated), and (4) the user dismissed the first window.

The conventional web history only retrieves opened time (shown on the top of Figure 1). Thus the opened time can be illustrated as a “point.” Our model retrieves both the “opened time” and the “closed time.” The duration of appearing of a web document on the computer screen can be represented as a “line.” Our model represents browsing activity with high accuracy by introducing the notion of “duration.”

In addition to the concept of duration, our model also considers how a web document is displayed on the screen. The size and location of the browser window can be captured to add the importance of URL. When a window overlaps other windows (see phase (3) of Figure 1), the importance of overlapped web document ([B]) is reduced. By employing our model, the importance of web document can be calculated by a definite integral of document appearing area.

## 2.2 Extracting user's browsing activity from window properties

In addition to the above basic model, we try extracting user's browsing activities from window events and status. We have currently focused on the following window attributes; (1) window ID, (2) window class, (3) location and size of the window, (4) z-order, (5) title, (6) URL, and (7) combination of events, and those events are: opened, closed, iconified, de-iconified, maximized, de-maximized, activated, de-activated, moved, resized, and reordered. We utilize "Window ID" to track a window attributes. "Window class" represents the type of application. The "Window class" property can be used to distinguish web browser window from other applications.

**Rating records** As we described in 2.1, the precise records in both temporal and spatial factors based on window appearance can be retrieved by considering the location, size, and z-order. These factors can be used to evaluate importance of web pages. Other attributes such as active/inactive status and iconified/deiconified events can supplement these factors for ratings. For example, the page which the user activates the window for a long time will be noted. Also the numbers of events such as activated, maximized and resized are informative factors because they indicate positive stance for the page.

We can infer some browsing situations and intentions by tracking the attributes. Suppose that the user dismissed a page. By considering the attributes, we can distinguish the following cases, (1) when, the window remains, the page was altered by following link but the browsing session continued. (2) When the window is destroyed, the user wanted not only to leave the page but also to quit browsing session. (3) When the window is iconified (stored in a task bar), the user had some reasons to interrupt browsing but he/she was still keeping attention to the page, and he/she will continue browsing.

**Relating records** Tools such as Footprints[8] and Browsing Icons[6] visualize page trails as graphs. These tools consider "explicit" relationship describing link structure among pages. The explicit relationship based on the "link" is promising. However, we focus on "implicit" relationship to vary the patterns of web revisit. We consider the window attributes as one of the implicit relationships.

The implicit relationship can be extracted by browsing actions, which include interaction between pages. Suppose that the two pages are displayed in mutually-adjointed windows. If the user moves a window near the other one, and if the edges of windows snap, the user intends to browse these pages concurrently. If the user activates these windows alternately, these pages can be related.

This relationship can be used to provide an associative search function. Suppose that a user, who cannot recall a keyword of target page directly but does remember some keywords of the page, opened either temporally or spatially adjoined windows. The user can reach the target page by following the implicit relationship.

The window attributes can be used to extract semantic aspects of web browsing activities. We consider the effect of this approach to become higher as the

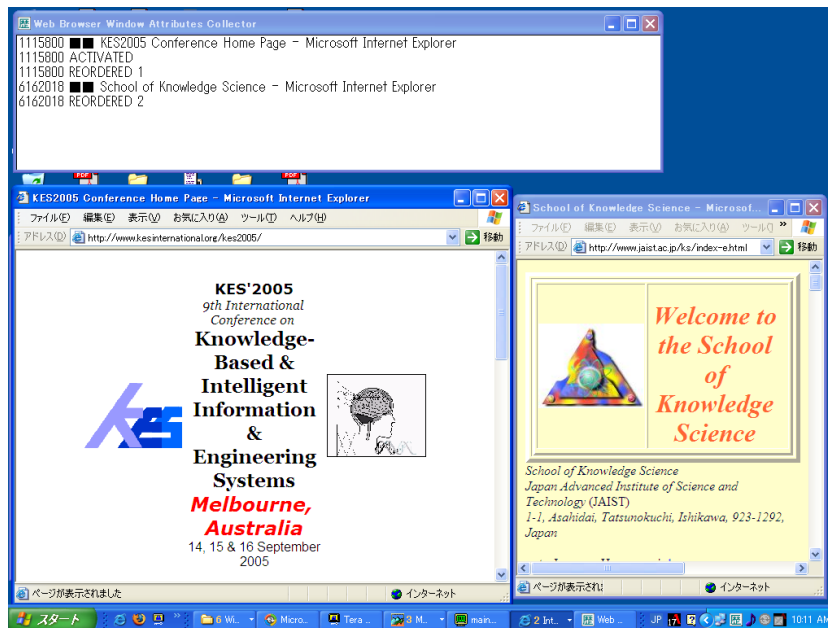


Fig. 2. The collector status window

display size becomes wider. A large display enables the user to freely layout the web windows; thus SXGA (1280×1024) or a larger display is appropriate for this approach.

### 3 System Implementation

In order to investigate the effects of extended web browsing records, we have developed two applications; a web-browsing window attributes collector for logging and a web browsing history viewer for searching.

#### 3.1 Collecting the attributes

We have designed the collector to run on Microsoft Windows environment, which is one of the most commonly distributed platform for web browsing tasks. The collector watches browser window attributes every second, and records the attributes if the value changes. The record is stored as a csv file. The current implementation of the collector supports Internet Explorer, Opera, and Mozilla.

The collector application initially appears as a tray icon. When the user clicks the icon, status window (Figure 2 top) appears, and lists current logging attributes. Figure 2 shows the situation when the user activates the left browser window. As a result, the left window (ID:1115800) becomes active and is displayed in the front. Thus the z-order of the right window (ID:6162018) changes

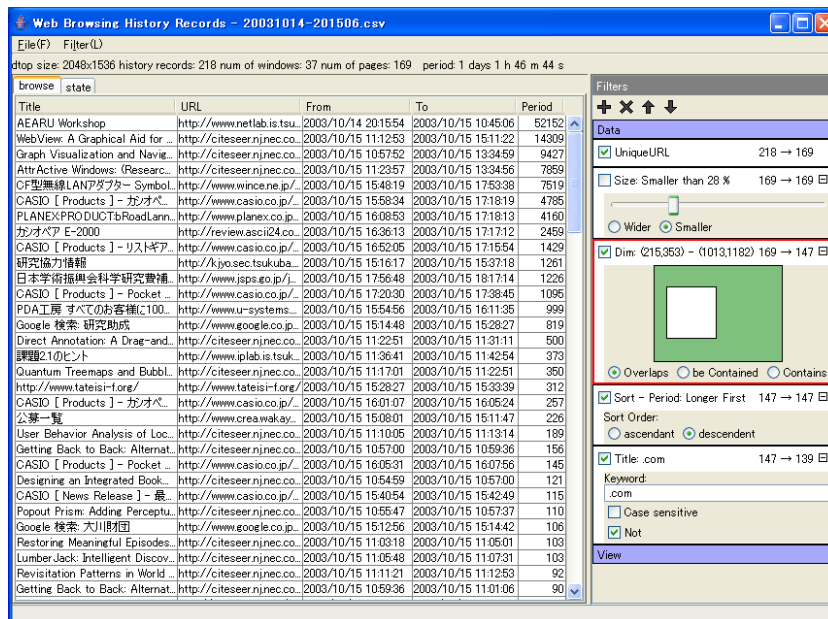


Fig. 3. Web History Records Viewer

to two. The status window is helpful in understanding how the collector works, but users are free from the window in daily use.

The additional merit of the collector is that the system can integrate web history records performed by several web browser applications. Typical web history records are closed by each browser. Of course a situation of handling several browsers at the same time is not popular. However, there are cases where people have to change their browsers to check a page, which recommends a particular browser environment. The collector unites multiple browser attributes held on the user's machine and associates them for a wider use.

### 3.2 Searching the records to re-visit sites

We have implemented a prototype of the web browsing history viewer as a Java application (Figure 3). The viewer loads a log file generated by the collector, and lists the records in a table view. The initial table view shows the following five attributes; page title, URL, open time, close time, and duration. The user can revisit a page by double-clicking on the URL.

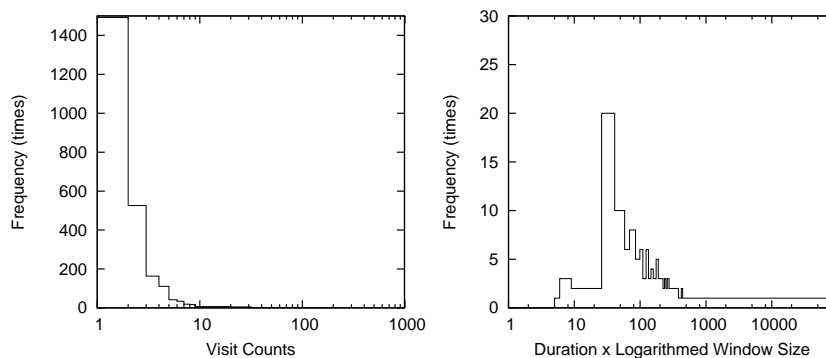
The user can sieve the records by specifying filters. The primary filters are; left unique by title/url, and can be searched by title/url, and sorts by title/url/date/duration/etc. When the user chooses a filter from a filter list, the filter appears in a filter dock (Figure 3 right-pane) and updates the list. The filters in the dock are applied in a top-down order. The user can easily mark a checkbox to validate/invalidate the filters.

## 4 Informal Observation

We conducted an informal observation to reveal the characteristics of our method. We collected several history records for two month, which included 11,476 items. Out of that the number of unique title items was 2,505. We calculated two different kinds of measures such as visited times and exposures for every items. Incidentally, we adopted only foremost and activated windows for accumulation of the exposures.

The top two results were common in measures; a start (portal) page and a search engine site like Google. According to our data, Wiki editable pages held the top of the exposures indexed list. The visited times tends to rise up a major “hub” page and scheduler page.

We assessed the distribution of these measures in Figure 4. The visited times was biased. More than half (59.6%, 1,492) pages were visited once, and 96.7% of pages were visited less than 10 times. The distribution of exposures were flatten than the visited times. The main purpose of re-visitation is to find a page, which was infrequently visited. Therefore the measures of exposures have advantages in finding lost pages.



**Fig. 4.** Distributions of Visit Counts and Proposed Measure

## 5 Related Works

Many researches have been carried out to visualize the Web in a zoomable two-dimensional pane [4] or a three-dimensional space [1, 2]. Systems which visualize web history such as WWW3D, The HyperVIS system, and WebPath are also summarized in different research papers. These systems can help a user to understand the path and its structure by mapping history records in a flexible virtual space. Our system aims to add extra factors which are extracted from window attributes to history records.

Regarding usability of web history, JasonSmith et al. [5] compared a linear temporally ordered list typified Netscape Navigator's history with a hierarchical structured list typified by Internet Explorer's history. Nadeem et al. [7] also compared a simple listed history with a visual graph history which is similar to PadPrints. These experiments mainly focus on the effectiveness of the interface with traditional history framework. In this paper we endeavor to promote a framework to enhance typical history records by capturing window attributes.

## 6 Concluding Remarks

We propose a method to enrich web history records by capturing attributes of web browser windows. The method focuses on "appearance" of web documents on the desktop screen. We believe that the attributes naturally reflect the user's intention of web browsing activities.

To realize the proposed method, we have developed tools for collecting window attributes, and for revisiting via history records. No operation is necessary for collecting and enhancing the history records; the window attributes are automatically captured, and are associated with URLs. The history viewer enables the user to search the records with a flexible filtering function. These tools can be helpful in finding valuable pages from huge history records. As a future work, we assess the effectiveness of this method using our system.

## References

1. S. Benford, I. Taylor, D. Brailsford, B. Koleva, M. Craven, M. Fraser, G. Reynard, and C. Greenhalgh. Three dimensional visualization of the world wide web. *ACM Computing Surveys*, 31(4es):25, Dec. 1999.
2. E. H. Chi. Improving Web Usability Through Visualization. *IEEE Internet Computing*, 6(2):64–71, 2002.
3. A. Cockburn and B. McKenzie. What Do Web Users Do? An Empirical Analysis of Web Use. *Int. J. Human-Computer Studies*, 54(6):903–922, June 2001.
4. R. R. Hightower, L. T. Ring, J. I. Helfman, B. B. Bederson, and J. D. Hollan. Graphical Multiscale Web Histories: A Study of Padprints. In *Proceedings of the ninth ACM conference on Hypertext and hypermedia*, pages 58–65, 1998.
5. M. JasonSmith and A. Cockburn. Get a Way Back: Evaluating Retrieval from History Lists. In *Proceedings of the Fourth Australian User Interface Conference (AUIC2003)*, pages 33–38. Australian Computer Society, Inc., 2003.
6. M. Mayer and B. B. Bederson. Browsing Icons: A Task-Based Approach for a Visual Web History. Technical Report CS-TR-4308, Computer Science Department, University of Maryland, 2001.
7. T. Nadeem and B. Killam. A Study of Three Browser History Mechanisms for Web Navigation. In *Proceedings of the Fifth International Conference on Information Visualisation (IV'01)*, pages 13–21, 2001.
8. A. Wexelblat and P. Maes. Footprints: History-Rich Tools for Information Foraging. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 270–277, May 1999.