
Enabling Cross-Device Interaction With Web History

Timothy Sohn

Nokia Research Center
955 Page Mill Road
Palo Alto, CA 94304 USA
tim.sohn@nokia.com

Koichi Mori

Nokia Research Center
955 Page Mill Road
Palo Alto, CA 94304 USA
koichi.mori@nokia.com

Vidya Setlur

Nokia Research Center
955 Page Mill Road
Palo Alto, CA 94304
vidya.setlur@nokia.com

Abstract

Internet-enabled personal devices are growing in number. As people own and use more devices, sharing information between devices becomes increasingly important. Web browsing is one of the most common tasks, thus sharing web history is a first step in supporting cross-device interaction. Current methods of sharing web history involve manual, cumbersome methods. This paper explores a system to automatically synchronize web information among a user's personal devices, and optimize the interface to support mobile users. We describe a system that enables users to quickly find directions on their mobile phone based on past web searches, and seamlessly share favorite web pages between their personal devices.

Keywords

Web history, maps, cross-device, mobile phone

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous

General Terms

Human Factors

Copyright is held by the author/owner(s).

CHI 2010, April 10–15, 2010, Atlanta, Georgia, USA.

ACM 978-1-60558-930-5/10/04.

Introduction

Computing devices are becoming more prevalent throughout people's lives. Mobile phones, netbooks, laptops, and media servers are increasing in popularity. As the number of heterogeneous devices increases, synchronizing a user's information across these devices becomes increasingly difficult. In addition to the data itself, interacting with multiple devices has only loosely been supported. For example, starting a task on a mobile phone and finishing the task on another devices such as a laptop/desktop.

Cross-device synchronization of data has widely been widely explored in distributed systems. Many of the problems of maintaining consistency and optimizing for performance were evident in early distributed systems such as Coda [7]. Today's devices have a variety of use contexts, which means that users need a way to interact with them differently based on their mode of use. A mobile phone would be used differently while on the move as opposed to a desktop machine.

This paper explores enabling cross-device interaction between a mobile phone and a laptop/desktop machine. In particular, we describe a system that shares web history and map data between a user's devices enabling them to access information in different contexts of use.

Related Work

There has been a large body of work exploring mobile web use and sharing of information across devices. We discuss some of the related work in these areas.

In a diary study of mobile information needs, Sohn *et al.* found that 30% of the time when users choose to address a need, they seek out web access (either via

their phone or at a location) [10]. This paper builds upon that work exploring how making mobile web use easier based on cross-device interaction can ease the burden of addressing mobile information needs. Cui and Roto conducted a study of mobile web use and found that users search for information on the mobile web in addition to putting personal content on the web for personal use [1]. Kane *et al.* found that smartphone users want to share web information between devices, but are often inhibited by cumbersome sharing methods [4]. Based on a study of 14 participants, their results suggest that sharing web information across devices can be beneficial for the user when accessing the mobile web.

Dearman and Pierce studied how users use multiple devices [2]. Their interviews showed that users do not use all their devices in the same way and need better tools to access information across devices. The authors suggest lighter-weight methods to transfer information between a user's devices. The PIE system takes some of these findings and creates an infrastructure that supports cross-device functionality [6]. Companies such as Google, Amazon, and Dropbox offer ways to share information across devices by storing data in the cloud. Google even offers a web search history¹ feature that synchronizes one's search patterns when logged in to a Google account. Our work explores how a user would benefit when all of their web history is shared across devices and also the interface/presentation of this information.

Sharing information across devices is a first-step towards improving the mobile experience. This

¹ <http://www.google.com/history>

information needs to be presented in a way that helps users quickly access the information they want. Kaasten *et al.* explored the presentation of web thumbnails to help users quickly find web pages that have been seen before [4]. Teevan *et al.* defined a visual snippet that uses a combination of salient images and text to provide compelling summaries of web pages for revisitation [10]. We believe the success of these visual cues in the desktop domain could also be successful in the mobile space when sharing web information across devices.

Scenarios

The following scenarios outline how we envision our cross-device system to be used. In particular, we want to support tasks that are common to the mobile user, such as saving items for use later and obtaining just-in-time directions to events [9].

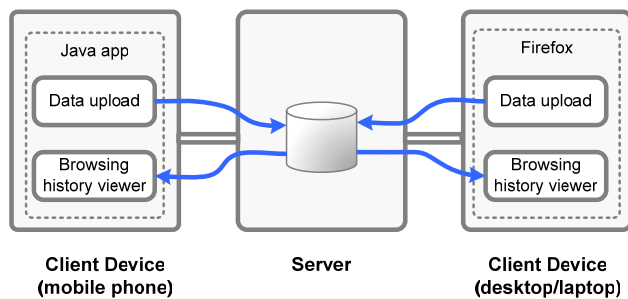


Figure 1. System design to share web history across a user's personal devices. A java application uploads mobile browser data to a central server. A firefox plugin uploads browser data from desktop/laptop machines to the server. Both client devices have a viewer to visualize the information.

Scenario 1: Kate is waiting in line to checkout at the grocery store. She passes the time by browsing through her Twitter/Facebook feeds. Her friends have been posting a link to an interesting news story. Just as she's about to click the link, she is called up next to checkout in line. Kate clicks the link knowing that her web history will automatically sync to her other devices. When Kate gets home, she opens her laptop and sees a link to the story she was about to read in line. She opens the webpage and reads the opinion piece on several recent hot topics.

Scenario 2: Joe just received an invitation to a party in a couple weeks. He looks up directions to get an idea of where it is before confirming his attendance. On the day of the party, Joe is out and about and doesn't remember how to get to the party location. Instead of searching through mounds of emails, he looks at his web synchronization application that shows recent map and direction searches he's made on all his personal devices. In one click he gets directions to the party because he searched for it earlier on his desktop device.

System Design

We provide a client-server architecture that facilitates synchronizing web history across multiple personal devices (e.g., desktop, laptop, mobile phone). An application resides on each device and uploads web history to a central server. The server encrypts all the web history using the user's personal key prior to it being saved in the database. This protects the data from being viewed by anyone else on the server. Figure 1 illustrates overview of the system with data flow.

The server stores every web history item with a timestamp, url, title, visit count, and thumbnail of the webpage. These thumbnails are cached to quickly provide client interfaces with an image preview of the webpage. A client device can query the server to get a list of web history items sorted by most recent, most popular, favorites, or map history. We chose these four sorting criteria based on the history items that a mobile user may want to explore. Location and directions are a key part of being mobile, thus we explicitly support this through maps in our system. Favorites correspond to bookmarks and items quickly saved such as those described with Kate in Scenario 1. Users could also

search through their web history as a last resort if they are looking for a particular page. We have chosen to try and offer a quick way for mobile users to access pages they may want a single click.

The current desktop implementation uses a Firefox plugin to capture web browsing history. Every time a user visits a webpage, the history item is uploaded to the server and stored in the database associated with the user. In the future, we believe that it would be easy to create plugins for other browsers to capture web data.

Due to the variety of browsers on the mobile platform, there is a greater challenge in acquiring web history data. We created an application that runs on certain Java-enabled phones to capture mobile web history and upload it to the central server. The mobile application also captures web items that a user may want to tag as a favorite for later viewing.

Web History Snippets

One of the key goals in designing our interface for presenting web history data to a user was to allow it to be easily browsable and identifiable. We created a visual web snippet algorithm that creates summarized thumbnail optimized for viewing on mobile platforms, but is also useful for desktop interfaces. The following section describes our algorithm in detail.

Creating web page snippets

One of the main goals of the mobile interface is to facilitate the user to search and re-finding previously visited web pages. Thumbnails can serve as an effective visual rhetoric for supporting the task of re-finding [3]. However, web pages are typically designed

with the desktop screen real estate in mind, and often use multi-column layouts and preformatted page widths. Merely scaling down such pages into thumbnails tends to be hard to read on small screens. We therefore generate snippets of web pages containing a small amount of image and textual content that are designed to contain the essential elements of the web pages in a smaller real estate and are also visually distinctive.

Our snippet generation algorithm, inspired by the Visual Snippets work [10], extracts three main elements from each web page: the title from the web page, the most salient image, and a portion of text surrounding the salient image. The title can be easily extracted by parsing the HTML content of the web page. The three pieces (title, salient image and salient text) are composited together into a 70 X 70 snippet based on the design template shown in Figure 2.

Salient Image Extraction

The most salient image is often placed in attentive areas of the web page. Typically on a web page, there are several images, but many of them are often advertisements and logos. We hence use the following heuristics for determining the salient image:

- *Aspect ratio.* The aspect ratio of the image is computed as the ratio of its height and width. Salient images have larger aspect ratios than less salient images.
- *Image format.* We check if the image is an animation or not as a Boolean valued feature. Salient images tend to be static. Advertisements, logos and banners tend to be animated.

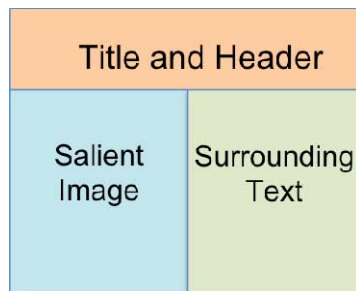


Figure 2. Design template for creating visual snippets

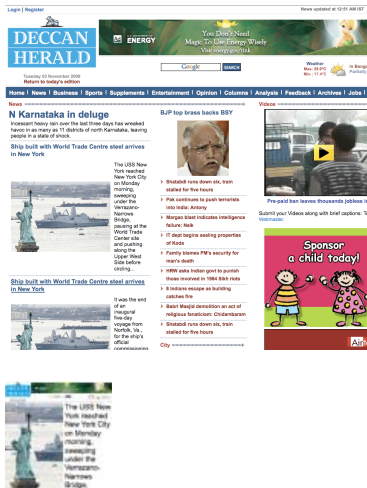


Figure 3. Webpage and corresponding snippet using salient text and images

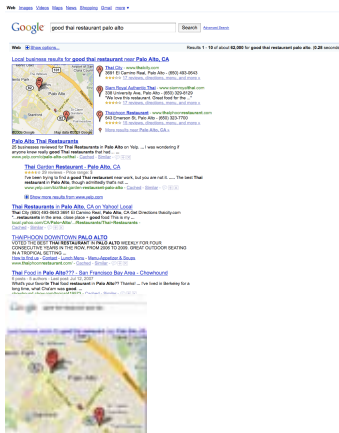


Figure 4. Map page and corresponding visual snippet

- **Position.** These are the x and y coordinates of the image. Salient images tend to be located at the top or center of the web page.
- **Area ratio.** The area ratio is the ratio of the image size to the web page size. Salient images tend to have a higher area ratio than the other images.
- **External source.** This is a Boolean value to denote whether an image URL is provided by another website or not. Typically advertisements have different hosts than their hosting web pages.

For every image on the web page, we compute the individual feature values based on the above features. In order to compare the images with each other, we normalize the non-Boolean feature values of the images. Normalization is achieved by utilizing a linear function to map the minimum value of a feature to zero, and map the maximum value of the feature to one:

$$f(x) = \begin{cases} 0 & x = \min \\ \frac{x - \min}{\max - \min} & \min < x < \max \\ 1 & x = \max \end{cases}$$

where x represents a feature, and \min , \max represent the minimum and maximum values of this feature of all the images in the web page. The images are subsequently ranked such that the highest ranked image has the most number of maximum values of all the features.

Salient text extraction

While it may be sufficient to merely utilize the most salient images in the web page snippet, in practice we have found that often the highest ranked images may not completely represent the context of the web page. The image heuristics are intrinsically specific to the image properties themselves and not the textual content to which they are contextually related. Given the limited real estate of the mobile interface, it is pertinent to maximize the relevance of the snippets used for search tasks. We hence use standard HTML parsing and information retrieval techniques to retrieve the surrounding text around the salient image [8]. Figures 3 and 4 show two examples of our snippet generation system for a typical webpage and a direction listing.

Visualizing Cross-Device Web History

Creating an intuitive user interface to view one's browsing history can be challenging depending on the device. Our goal was that the interface should be easily viewable and be a launching pad to access the web history items that a user needs in the moment. The design of our interface is in the spirit of Top Sites enabled by modern browsers such as Safari² or Google Chrome³. We created both a mobile and general web interface that can be accessed from any browser, either desktop or mobile. The mobile interface in particular is designed to have a native look and feel to the platform it runs on. Both interfaces take advantage of our snippet system that is optimized for small screens in order to support finding previously seen web pages.

² <http://www.apple.com/safari/>

³ <http://www.google.com/chrome>



Figure 5. Web interface showing four tabs of different web history sorting mechanisms. The map interface shows recent location and direction searches

Unlike typical web history views that show a list of urls, our web browsing history view provides several sorted views for quick access. Raw browsing history can be difficult to sort through and could be difficult for a user to what he is looking for. As seen on the screenshot in Figure 5 the view has a tabbed list view with four tabs: most viewed web history, recently viewed web history, map activity history, and favorites. The list view shows a small thumbnail image of the Web page along with the title of the page. Each of these tabs

corresponds to the sorting mechanisms that are enabled in our backend server. Each tab shows a list view with the history items title, url, and a thumbnail generated by the visual snippet system.

Conclusions and Future Work

We have presented a system that enables cross-device interaction through web sharing. Our system seamlessly shares web history data across a user's personal devices and presents the information in an intuitive manner to assist the mobile user. Two particular scenarios that our system is designed around is helping mobile users capture information that they are unable to address in the moment, and also accessing previously seen map and direction information that would be useful while on the move. In the future we would like to explore other ways to view

this information and also conduct a pilot study to understand the benefits of sharing web history among a user's personal devices.

References

- [1] Cui, Y. and Roto, V. 2008. How people use the web on mobile devices. In *WWW '08*, pp. 905-914.
- [2] Dearman, D. and Pierce, J. S. It's on my other computer!: computing with multiple devices. *CHI '08*, pp. 767-776.
- [3] Kaasten, S. et. al. How people recognize previously seen Web pages from titles, URLs and thumbnails. In *HCI '02*, pp. 247--265.
- [4] Kane, S. K et. al. Exploring Cross-Device Web Use on PCs and Mobile Devices. *Interact 2009*, pp. 722-735.
- [5] Karlson, A.K. et. al. Working Overtime: Patterns of Smartphone and PC Usage in the Day of an Information Worker. In *Pervasive 2009*, pp. 398-405
- [6] Pierce, J. S. and Nichols, J. 2008. An infrastructure for extending applications' user experiences across multiple personal devices. In *UIST '08*, pp. 101-110.
- [7] Satyanarayanan, M. Experience with Disconnected Operation in a Mobile Computing Environment. In *USENIX Symposium on Mobile & Location Computing 1993*.
- [8] Salton, G. et. al. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, 1975
- [9] Sohn, T. et. al. A Diary Study of Mobile Information Needs. In *CHI '08*, pp. 433-442.
- [10] Teevan, J. et. al. Visual snippets: summarizing web pages for search and revisitation. In *CHI '09*, pp. 2023-2032.