

Multibrowsing: Moving Web Content across Multiple Displays

Brad Johanson, Shankar Ponnekanti, Caesar Sengupta, and Armando Fox

Stanford University

Abstract. Although ubiquitous computing hardware technology is widely available today, we believe one key factor in making ubiquitous computing useful is a framework for exploiting multiple heterogeneous displays, whether fixed or on mobile computing devices, to view and browse information. To address this issue, we propose *multibrowsing*. Multibrowsing is a framework that extends the information browsing metaphor of the Web across multiple displays. It does so by providing the machinery for coordinating control among a collection of Web browsers running on separate displays in a ubiquitous computing environment. The displays may be "public" (e.g. wall-sized fixed screens) or "private" (e.g. the screens of individuals' laptops or handhelds). The resulting system extends browser functionality for existing content by allowing users to move existing pages or linked information among multiple displays, and also enables the creation of new content targeted specifically for multi-display environments. Since it uses Web standards, it accommodates any device or platform already supported by the Web and leverages the vast existing body of Web content and services. We describe the design and implementation of multibrowsing and a variety of scenarios in which we have found it useful in our test bed ubiquitous computing environment.

1 Introduction

Although the Internet has been active for over thirty years, it was the simple-yet-powerful information sharing mechanism provided by the Web, that significantly enhanced the usefulness of Internet-connected computers and led to the wide and rapid adoption and development of the Internet. In the last few years, we have seen the beginnings of a similar trend: ubiquitous computing hardware has become increasingly affordable and mainstream. However, we are still awaiting the catalyst applications that will help ubiquitous computing "take off", just as the Web did for Internet-based computing.

We believe one such application is flexible information movement. In this paper, we describe one such information movement system, *multibrowsing*, a framework for movement of web content across the displays in an ubiquitous computing environment. The presence of multiple display surfaces is a common feature of ubiquitous computing environments. For example, the iRoom is an experimental conference/workroom we have built as part of the Stanford Interactive Workspaces project [5]. As envisioned by Weiser [12], the iRoom includes

three whiteboard-sized touch-sensitive screens, a bottom projected table-top display, and multiple laptops and handhelds. Multibrowsing allows web content to be freely moved across these displays. In particular:

1. A web page currently displayed in a browser on an enabled machine can be redirected to any other display in the environment without direct interaction with the target display.
2. A web page currently displayed on any display in the environment can be “pulled onto” a local browser without the need for copying the URL and typing it in the local browser.
3. Web pages can be purpose-designed for multi-display viewing by embedding links which open content on other displays in the environment.

We have found these capabilities to be useful in the iRoom, especially for meetings and presentations. Multibrowsing has also proven useful in offices with multiple PCs. Several other systems such as iLand’s Passage [10], pick-and-drop [8], augmented surfaces [9], and Greenberg et al. [4] have also addressed the information movement problem. Multibrowsing is however unique in that it heavily leverages web technologies, thus resulting in simplicity and ease of deployment. In addition, the system strives for robustness and minimal configuration. Multibrowsing also provides a simple technique for easy authoring of content for multi-display viewing. This paper presents the multibrowsing system architecture and implementation, and gives some examples of its application in the iRoom, where it is in day-to-day use.

2 Architecture and Implementation

The following section describes the overall architecture of the system, and details of its use. This includes the butler service, which runs on machines that are multibrowse targets, the MB2Go browser plugin, and details of how embedded multibrowsing links function.

2.1 Framework

In the multibrowsing framework, displays can assume *one or more* of the three different roles described below:

1. Regular Client: A *regular client* can direct special links from web pages purpose-designed for multi-display viewing to other displays in the environment.
2. Enhanced Client: An *enhanced client* can direct any regular web page (or link on a web page) to any enabled display in the environment. Similarly, an enhanced client can “pull to itself” a web page from any enabled display in the environment.
3. Target: A *target* allows web pages to be directed to it from clients. Also, a target allows web pages displayed on itself to be “pulled off” by enhanced clients.

Typically, private devices such as laptops and handhelds function as clients and public devices such as wall-mounted and table-top displays function as targets. The system itself does not, however, prevent other configurations.

A regular client only needs an off-the-shelf web browser. Thus, handhelds, laptops, wall-mounted, and table-top displays can all function as regular clients with *no* custom multibrowse software installed. Enhanced clients are required to install a custom browser plug-in called *MB2Go*. Targets run a special service called the *butler service*. The overall architecture is shown in figure 1.

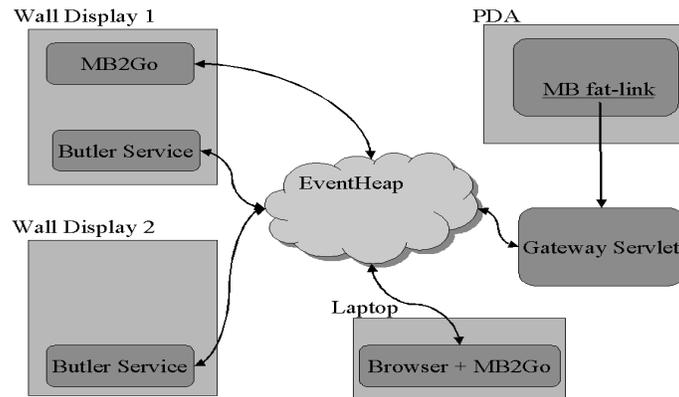


Fig. 1. Multibrowsing Architecture. Wall display 1 acts as both a target and an enhanced client. Wall display 2 acts only as a target. The laptop acts as an enhanced client while the PDA acts as a regular client. Regular clients use *multibrowse fat-links* to direct pages to other displays. The multibrowse fat-links are handled by the *gateway servlet*.

2.2 Butler Service on Targets

The butler service is written using the ICrafter [7] service framework (which itself utilizes an underlying tuplespace [3] based event system called the Event Heap [5]). The service framework provides a soft-state discovery system which allows dynamic detection of available target displays. Each butler service has a unique name which remains unchanged across crashes and restarts. The butler service provides the following methods which can be invoked by sending appropriately constructed events:

- `browse(String URL)` : Causes the passed URL to be displayed on a locally running browser.
- `String getDisplayedResource()` : Returns the URL currently displayed in the top-level browser window.

Both these methods were implemented using the COM/OLE APIs exported by Internet Explorer.

2.3 MB2Go

MB2Go is a browser plug-in that consists of a Java application dynamically detects the available butler services and allows redirection of web pages. MB2Go modifies the Windows registry to add additional items to the IE context menu. When the user right-clicks on a hyperlink in IE, the context menu that appears contains an additional option called “multibrowse link”. If the user selects this option, a pop-up window appears that allows the user to select the desired target from the list of discovered target displays. When the user selects a particular target, an appropriately formatted event is submitted that causes the butler service on the target to display the linked web page. The context menu that appears when a non-link area on the browser is clicked contains two additional items, “multibrowse to” and “multibrowse from”, which respectively allow the user to cause the web page locally displayed to be loaded on a remote display or vice-versa. Figure 2 shows the workings of “multibrowse from”. The other two cases (“multibrowse to” and “multibrowse link”) are similar.

2.4 Multi-display viewable web pages

Web pages can be purpose-designed for multi-display viewing using *multibrowse fat-links*. These are hyperlinks that encode a web-page display request. An example multibrowse fat-link (simplified for illustrative purposes) is shown below:

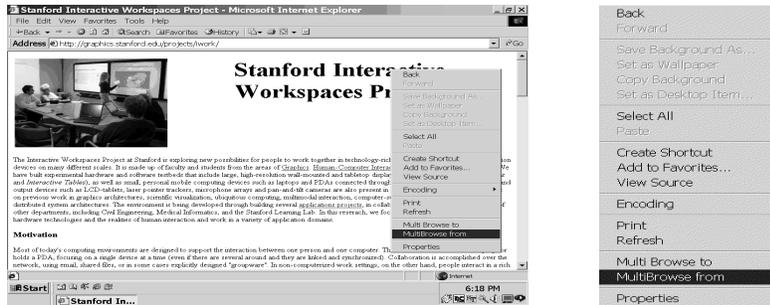
```
http://gateway/multibrowse?SvcName=Butler1&OpName=browse&param1=
www.stanford.edu
```

Clicking on this fat-link causes a form submission to the gateway servlet, which decodes the request and submits an appropriate event to the butler service instance `Butler1` to display the web page `http://www.stanford.edu`. A multibrowse fat-link can also encode multiple events thus causing several pages to be loaded on various displays with a single click. We also have a helper program that allows end-users to easily construct these fat-links.

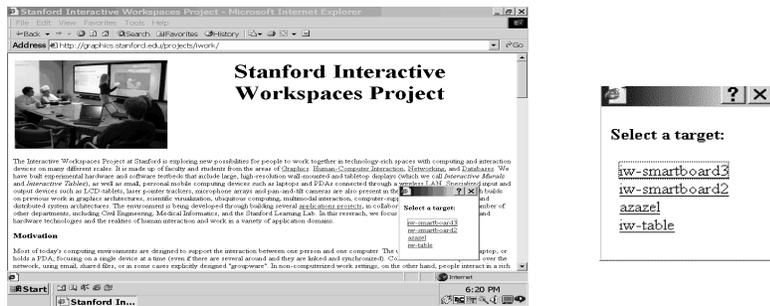
Unlike the butler service and the plug-in, the fat-link mechanism is currently not fully portable since the fat-links hardcode the path to the local gateway servlet and the names for the target service instances. To overcome this limitation, we are currently exploring a proxy mechanism to dynamically rewrite hyperlinks for the local environment.

3 Usage Scenarios

In this section, we present typical usage scenarios of multibrowsing in iRoom. The iRoom has an 802.11b wireless network (WaveLAN) which allows appropriately-



(a) Modified context menu. Note the two new items "Multibrowse From" and "Multibrowse To" in the context menu



(b) Target selection popup window. When the user selects "Multibrowse From", a window pops up that allows the user to select a target

Fig. 2. . MB2Go Illustrated. When the user wishes to load a page displayed on a remote display on her local browser, she simply right clicks on the browser, selects "multibrowse from" (part (a)) and then chooses the desired display from the pop-up window (part (b)).

equipped laptops or handheld devices to participate in multibrowsing in addition to the large wall-mounted and table-top displays.

3.1 Meetings

Ad-hoc meetings: We will step through the sequence of interactions in a typical iRoom research meeting to illustrate the use of multibrowsing: Susan mentions that she has heard of the new ABC wireless modem that could be useful for the handheld devices in the iRoom. Using the multibrowse plug-in from a browser on her laptop, she opens the web page of ABC modem on SmartBoard1. Meanwhile, Brad uses a search engine on his laptop and locates a site that summarizes the features of various commercially available wireless modems. Again using MB2Go, he multibrowses these webpages to the other two SmartBoards so that the group can compare the features of the modems. The ability to freely pull up information and move it to different displays is a key useful feature for such meetings.

Meeting Agenda: In iRoom meetings, meeting coordinators can prepare a “meeting-agenda” web page containing links to information relevant to the meeting. During the meeting, the coordinator opens this web page on a laptop or SmartBoard and uses MB2Go to multibrowse the linked pages to the other displays in the room. The coordinator can also pre-design the agenda web page with suitable multibrowse fat-links such that clicking on the agenda items causes them to appear on the other displays.

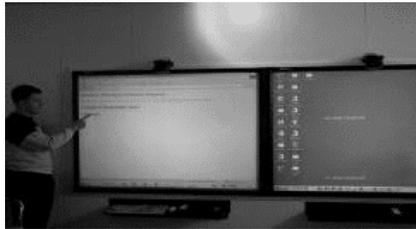
3.2 Presentations

One use of purpose-designed content is to enhance presentations in multi-display environments. To illustrate this, we describe a typical iRoom presentation. The “lead slide” of the presentation gives a broad overview of the project and contains several multibrowse fat-links to the other presentation slides. In a typical presentation session shown in figure 3, the lead slide is displayed on SmartBoard1. Clicking on any link on the lead slide causes detailed information about that category to be brought up on the other displays. For example, when the presenter clicks on the multibrowse fat-link “overface” on the lead slide, a web page showing the detailed block diagram of the overface system is displayed on SmartBoard2 and the primary goals of the overface system are displayed on SmartBoard3 (see figure 3). A similar chain of events occurs when “Graphical Presentation Tools” is clicked.

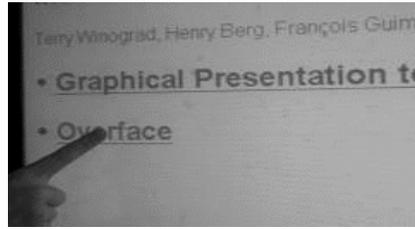
It is worth noting that this mechanism only needs a regular (i.e., non-enhanced) client for displaying the “controlling” slide. The presenter can also control the presentation with any other device with an off-the-shelf web browser.

4 Related Work

Multibrowsing was first discussed as an example application in [2], but details were not presented and MB2Go and dynamic discovery of targets was not op-



(a)



(b)



(c)

Fig. 3. Multibrowse presentation screenshots. The “lead slide” displayed on SmartBoard1 contains two multibrowse fat-links. When the user clicks on one of them (“overface”), a page describing overface automatically pops up on SmartBoard2. Note that for clarity, the figures do not show SmartBoard3.

erational at that time. Controlling other web browsers has been previously explored for collaborative browsing [1] [6]. The iLand project at GMD-IPSI in Darmstadt [10] has developed sophisticated groupware environments that coordinate information display using object synchronization. Pick-and-Drop [8] and augmented surfaces [9] present new metaphors for moving objects across computers. Greenberg et al. [4] present a system for moving information between PDAs and public displays. Multibrowsing is primarily distinguished by a larger reliance on off-the-shelf web technologies which results in simplicity and easier deployment. Unlike some of these projects, robustness and minimal configuration are first-class goals for multibrowsing. For example, dynamic discovery of displays in the vicinity enables the system to work transparently without any per-workspace configuration even when target displays (such as laptops) freely enter and leave the workspace. Similarly, due to soft-state service advertisement, if a butler service on a target display crashes, it is automatically removed from the list of targets. Finally, while multibrowsing is limited to web content, this restriction is ameliorated since emerging technologies like Web Folders and Web-DAV [11] allow many types of content to be made available on the web.

5 Conclusions and Future Work

We have designed and implemented multibrowsing, a system for movement of web information across displays which leverages web technologies for simplicity. Along with robustness, and minimal configuration requirements, this simplicity makes multibrowsing easy to deploy. We are now beginning to address the following challenges:

Security: We are exploring security and access control models appropriate for both public and private targets and clients.

Increased Portability: Apart from the proxy mentioned in section 2.4 which allows construction of more portable multi-display viewable web pages, we are also exploring building a proxy that automatically replaces links on standard web pages with new versions that allow for redirection of the linked pages to displays in the current environment. Such a proxy would allow even clients without MB2go to multibrowse regular web pages. Providing these new functions in a fully portable manner with *no* per-environment configuration is, however, challenging.

6 Acknowledgments

We thank Pat Hanrahan, Emre Kiciman, Brian Lee, Kathleen Liston, Susan Shepard, and Terry Winograd for their help with the design and implementation. Andy Huang and Ben Ling provided helpful suggestions for improving this draft.

References

1. CobWeb Collaborative Web Browsing System. <http://www.cs.unc.edu/~stotts/cobweb/doc/index.html>.
2. Fox, A., Johanson, B., Hanrahan, P., Winograd, T., "PDA's in Interactive Workspaces," *Computer Graphics and Animation*, May, 2000.
3. David Gelernter. Generative communication in LINDA. *ACM Transactions on Programming Languages and Systems*, 7(1), pages 80–112, January 1985.
4. S. Greenberg, M. Boyle and J. LaBerge. PDAs and Shared Public Displays: Making Personal Information Public, and Public Information Personal. *Personal Technologies*, Vol.3, No.1, 54-64, March. Elsevier.
5. Interactive Workspaces Group at Stanford University. <http://graphics.stanford.edu/projects/iwork>.
6. Netscape Conference. <http://home.netscape.com/communicator/conference/v4.0>.
7. Shankar Ponnekanti, Brian Lee, Armando Fox, Pat Hanrahan, and Terry Winograd. ICrafter: A service framework for ubiquitous computing environments. *Proceedings of Ubicomp '01*, October 2001.
8. Jun Rekimoto. Pick-and-Drop: A Direct Manipulation Interface for Multiple Computer Environments. *UIST '97*, pages 31-39, 1997.
9. J. Rekimoto and M. Saitoh. Augmented Surfaces: A Spatially Continuous Workspace for Hybrid Computing Environments. *CHI '99*, 1999.
10. N.A. Streitz et al. i-LAND: An interactive Landscape for Creativity and Innovation. *CHI '99*, pages 120-127, May 1999.
11. WebDAV. <http://www.webdav.org>.
12. M Weiser. The Computer for the 21st Century. *Scientific American*, 265(30), pages 94-104, 1991.