SFDC 1007

A Zooming Web Browser

Benjamin B. Bederson, James D. Hollan,
Jason Stewart, David Rogers, Allison Druin, David Vick
Computer Science Department
University of New Mexico
Albuquerque, NM 87131
{bederson, hollan, jasons, drogers, allisond, dvick}@cs.unm.edu
http://www.cs.unm.edu/pad++

ABSTRACT

The World Wide Web (WWW) is becoming increasingly important for business, education, and entertainment. Popular web browsers make access to Internet information resources relatively easy for novice users. Simply by clicking on a link, a new page of information replaces the current one on the screen. Unfortunately however, after following a number of links, people can have difficulty remembering where they've been and navigating links they have followed. As one's collection of web pages grows and as more information of interest populates the web, effective navigation becomes an issue of fundamental importance.

We are developing a prototype zooming browser to explore alternative mechanisms for navigating the WWW. Instead of having a single page visible at a time, multiple pages and the links between them are depicted on a large zoomable information surface. Pages are scaled so that the page in focus is clearly readable with connected pages shown at smaller scales to provide context. As a link is followed the new page becomes the focus and existing pages are dynamically repositioned and scaled. Layout changes are animated so that the focus page moves smoothly to the center of the display surface while contextual information provided by linked pages scales down.

While our browser supports multiscale representations of existing HTML pages, we have also extended HTML to support multi-scale layout *within* a page. This extension, *Multi-Scale Markup Language* (MSML), is at an early stage of development. It currently supports inclusion within a page of variable-sized dynamic objects, graphics, and other interface mechanisms from our underlying Pad++ substrate. This provides sophisticated client-side interactions, permits annotations to be added to pages, and allows page constituents to be used as independent graphical objects.

In this paper, we describe our prototype web browser and authoring facilities. We show how simple extensions to HTML can support sophisticated client-side interactions. Finally, we discuss the results of preliminary user-interface testing and evaluation.

Keywords: world-wide web, browser, information navigation, zooming, information visualization, multiscale information, animated user interface, Pad++.

1. INTRODUCTION

In 1945 Vannevar Bush [8] envisioned "a future device for individual use, which is a sort of mechanized private file and library." He termed this device a *memex* and proposed a form of associative indexing in which arbitrary pieces of information could be linked together such that "when one of these items is in view, the other can be instantly recalled by tapping a button." He further conjectured that "wholly new forms of encyclopedias will appear, ready made with a mesh of associative trails running through them, ready to be dropped into the memex and there amplified." Today, fifty years later, we have the World Wide Web and a memex in the form of web browsers. See [4] for an overview of the WWW.

The increasing number of users and the ever-growing quantity of information available on the web present challenging interface and navigation problems. There are a variety of human factors [19] issues that need to be addressed. A larger number of users means that people with diverse talents, interests, and experiences will be on-line via the web. Many will be novices with little prior experience with computers. A simple click of the mouse can bring a user from their friend's home page to unknown destinations across the world. Traditionally, following a cross reference meant shuffling across the library to find another volume. While time-consuming, this reinforced the transition that was taking place. The difficulties that novice users confront can be instructional to developers. While experts may not have as much difficulty, they experience the same cognitive burdens, and may just have a higher threshold before they experience similar difficulties.

While the immediacy of traversing information links offers many advantages, it can also make it difficult to maintain an intuitive sense of where one is, and how one got there - leading to the frequently described sense of being *lost*. This is a classic problem of hypertext systems. Part of the problem can be attributed to windows-based interfaces. Current window systems don't readily support showing more than a few pages at a time. In addition, each page is usually in a separate window with no depiction of relationships to other windows. Popular WWW browsers, like other applications built according to current tiled or overlapping windows philosophies, also have this same problem, although they do offer limited methods to aid navigation by keeping track of interesting sites - usually in hierarchical sets of *hotlists* or *bookmarks*.

Several groups have proposed alternatives and extensions to browsers to address some aspects of this problem. Oostendorp describes the PAINT system (Personalized Adaptive Internet Navigation Tool) [25]. It provides an interface for accessing hierarchies of bookmarks in a style similar to the NEXTStep interface. WebMap is a browser extension that shows a graphical relationship between web pages [11]. Each page is represented by a small circle that can be selected to display the actual page. The links between pages are colored to indicate information about the links, such as whether it is a link to a different server or whether the destination page has already been read. These graphs may be saved and used by others.

While the web is inherently cyclic, it is easier to visualize hierarchies, and so many web visualizations are based on hierarchies extracted from the graph of the web. Some interesting work focuses on alternative visualizations [24]. Furnas [14] shows how *multitrees* can be used to represent a collection of hierarchies sharing parts of the underlying data. One application of multitrees is visualization of bookmarks from multiple individuals[34]. Furnas [16] also describes a framework for characterizing how different structures influence effective view traversal, the mechanical process of moving between information items, and view navigation, finding good paths to information items.

Another approach to visualizing large information spaces that can be applied to web browsing and navigation involves techniques to show detail at particular nodes while maintaining context. One general approach, fisheye views [13], has been extended with graphics [30], three dimensions [9][22], hyperbolic representations[20], animation [10], and zooming [2][3][28]. Other techniques include exploiting a large virtual space [12], using lenses or filters [5][23][31], and visualizing two dimensional layouts [1][21].

In addition to the difficulty of finding information, it becomes ever more important to tailor information for one's own needs. Also rather than searching oneself can be sensible to go by other's recommendations. This is the basis for commercial services such as Yahoo [36], and follows the often effective strategy of exploiting recommendations from those one knows and trusts [35].

Annotations are another important information tailoring facility. Annotations are personal markings that can be used to highlight and comment on information for oneself and others. One interesting approach to annotation on the web separates the annotations from the original documents and stores them in a special annotation server [29]. Used with an enhanced browser, displaying a new page automatically brings in the annotations of others and integrates them into the page.

In the sections that follow, we describe our zooming web browser and the attempt to use animation and multiscale representation of context to support more effective web navigation. In addition to visualization of standard HTML pages, we introduce extensions to HTML that allow more sophisticated presentations and client-side interactions. We demonstrate the beginnings of direct manipulation graphical authoring tools and show how annotation can be supported as a form of authoring. Finally, we present the results of initial user testing and envision a scenario for future web use in the classroom.

2. A ZOOMING WEB BROWSER

Navigating the WWW presents a struggle between focus and context. As one browses or searches the web the need for detailed views of specific items conflicts with the need to maintain a global view of context and history of traversal. This struggle is made more difficult by the haphazard organization of the WWW. Information items closely related by links are not necessarily closely related by content nor in terms of the user's information needs. At times, one seems more likely to find something of interest when not looking for it than when specifically searching. This serendipitous contact with information, though at times frustrating, can also be an advantage. The challenge is how to best support both incidental and intentional access while organizing useful information so that it can be effectively retrieved again in the future.

We are exploring dynamic multiscale techniques to support focus and context during navigation of large information spaces. To accomplish this we are building a zoomable web browser using Pad++, a substrate for building multiscale dynamic user interfaces [2][3][27][28]. Pad++ provides an extensive graphical workspace where dynamic objects can be placed at any position and at any scale. Pad++ supports panning and zooming. Zooming can involve simple geometric scaling or what we term *semantic zooming*, in which rendering of objects can vary based on factors in addition to scale, such as context of the task or complexity of the information being displayed. Pad++ is built as a widget for Tcl/Tk, a scripting language and user-interface library [26][33].

Pad++ allows WWW pages to remain visible at varying scales while they are not specifically being visited, so the viewer may examine many pages at once. In addition, Pad++ allows the user to zoom in and out of pages, enabling explicit control of how much context is viewed at any time. To orient themselves, users can simply zoom back to view a number of web pages. To get more detailed views of a particular page they can zoom in. We think this variable scale contextual display of web pages can provide important support for navigation. We are currently exploring a tree layout system that permits users to dynamically add to and reorganize a tree of web pages. Using our Pad++ web browser, users navigate a space filled with familiar objects, not iconified representations of those objects.

Our dynamic Pad++ tree browser combines a basic focus-driven layout with automatic zooming and panning to support navigation. The software allows the user to select a focus page. That selection animates the page to occupy a larger section of the display. Pages farther from the focus page get increasingly smaller, resulting in a graphical fisheye view [30]. See Figures 1 and 2 for snapshots of the Pad++ web browser during reorganization.

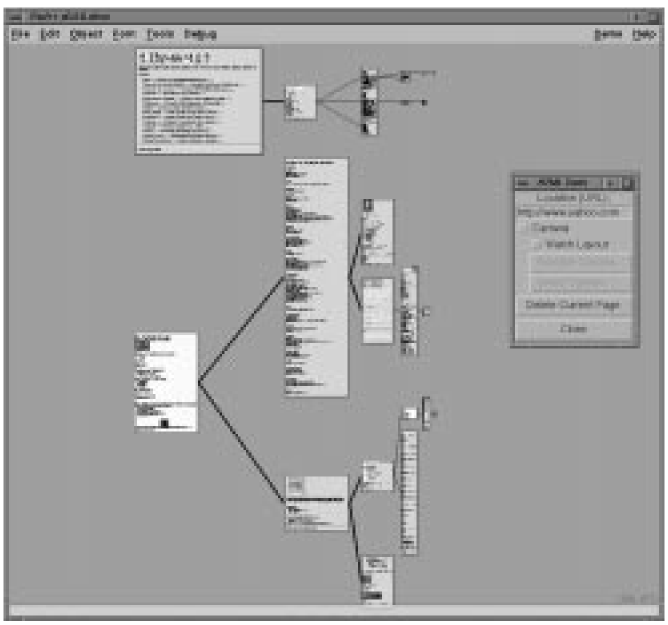

Figure 1: Snapshot of Pad++ Web Browser.

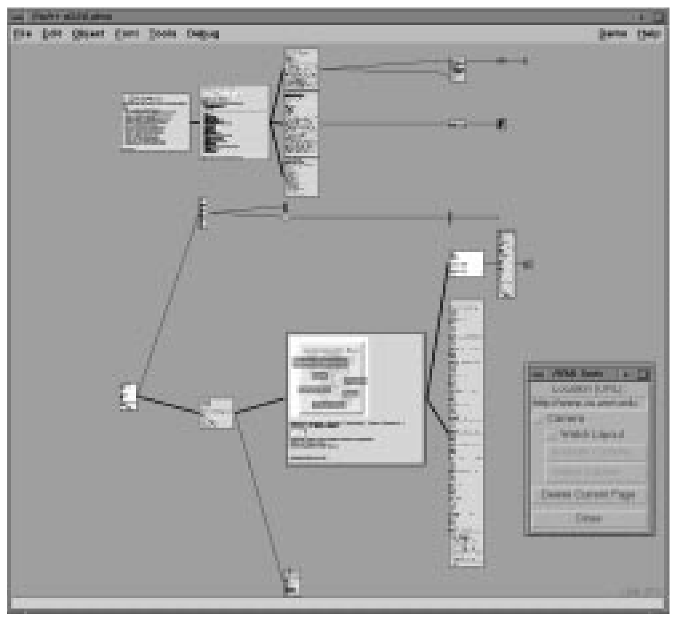

Figure 2: Another view of same web pages.

The Pad++ WWW browser combines Pad++'s interactive multiscale display with dynamic objects that can restructure themselves in response to user actions. Clicking on a link brings up a new page, adds it to the tree of pages, and causes the tree to restructure itself. Unlike other web browsers that immediately replace the current page with a new page, the restructuring process is animated so that users can understand how the tree is being reorganized. The animation helps maintain object constancy and the graphical depiction of links highlights relationships between pages. The new page becomes the current focus and is moved to the center of the screen, at a size suited for viewing. The user may designate any existing page to be the current focus by clicking on it.

As in earlier fisheye displays, our basic layout function assigns a *degree-of-interest value* to each node in the tree based on its distance from the focus page. We define the distance to be the shortest path between two pages[13]. This value is then used to determine the size of each node. See [11] for a description of other hierarchical layout techniques not based on fisheye views.

The layout described above provides a sense of context while following links. We have also implemented an alternative *camera mode* of navigation. It shows the web of links on one side of the screen with a zoomed in view of the focus page on the other side of the screen. A camera is depicted along with the web of pages. The camera can be dragged around or automatically animated through the web. The zoomed in view shows the page the camera is currently looking at (Figure 3). This mode also supports automated tours. For example, one type of camera can take you on a tour of all the parts of your saved web pages that have changed since you last looked at them.

We are currently experimenting with more flexible mechanisms for dynamic tree layout and interaction. These include exploring alternative visualizations and better methods for managing and interacting with large dynamic trees. New tree layout methods will work with any kind of item on the Pad++ surface. Thus, in addition to HTML pages, users will be able to create spaces using any Pad++ object, including drawings, interactive maps, and text.

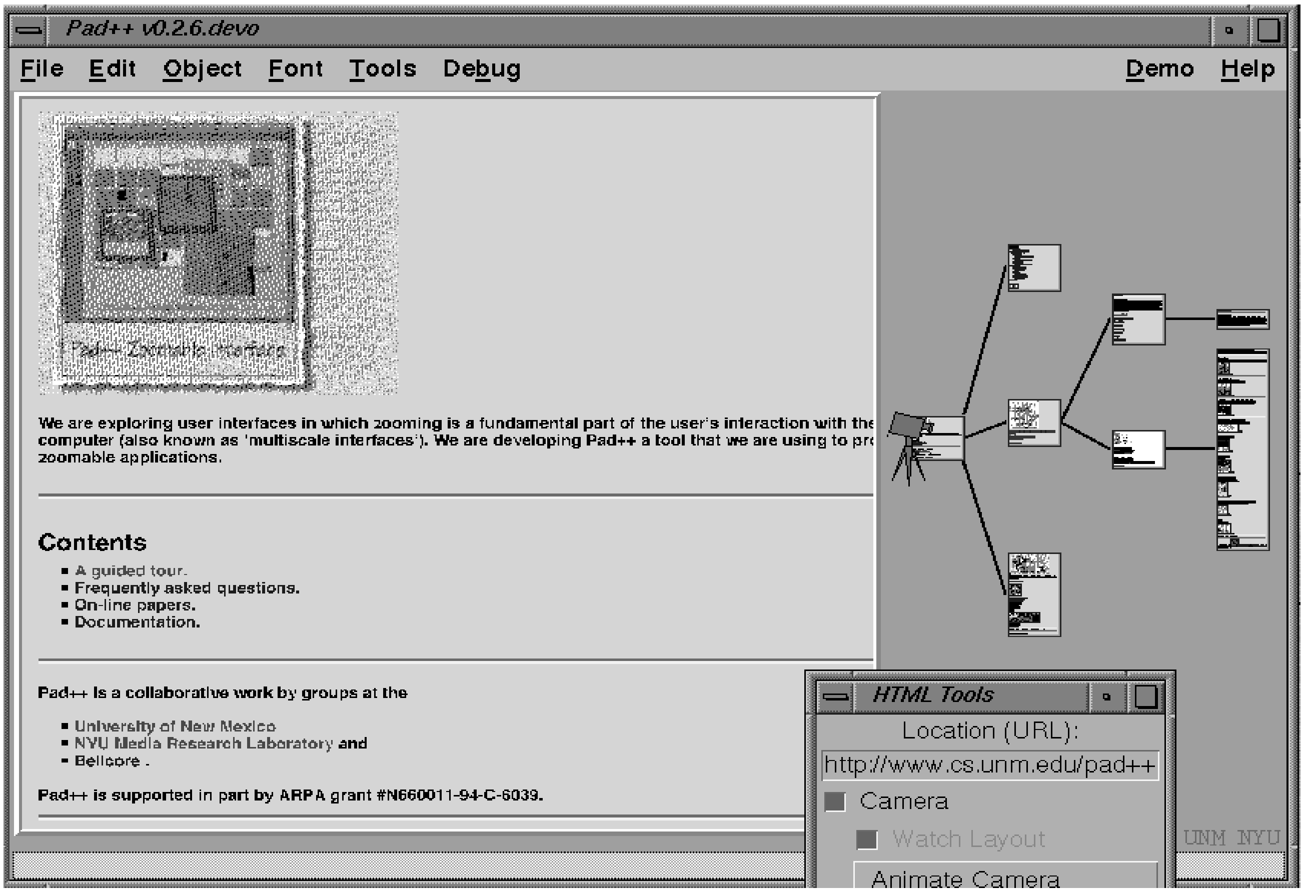

Figure 3: Camera view of web pages.

The new layout code is designed to be hierarchical, so that users may designate subtrees to have different layouts. This allows greater freedom in grouping and display. For example, a certain information tree may contain nodes with subtrees consisting of hundreds or thousands of nodes each. These nodes could exploit a hyperbolic layout to compress the information and the hyperbolic nodes themselves might be layed out radially [20].

Another topic we are exploring involves tradeoffs between maintaining pointers to information on the Web and making copies of the information locally. For example, a user might want to copy items from a remote page to prevent that information from being lost. At other times users may wish to maintain only pointers to information since it is being maintained elsewhere. There are interesting related issues of annotation, that we discuss later, as well as issues of maintaining annotation placement as pages change.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.