

Web Navigation: Resolving Conflicts between the Desktop and the Web

A CHI 98 Workshop

Carola Fellenz, Jarmo Parkkinen and Hal Shubin

This paper summarizes a workshop at CHI 98 that focused on navigational problems caused by differences in navigational models between the desktop and the Web. The goal of this workshop was to identify usability problems encountered when users move from the “traditional” desktop to the Web and to identify ways to minimize transfer-learning problems between the two platforms.

The major outcome of the workshop was the classification of Web usage. We developed three categories: browsing, performing transactions and running applications. Each type of usage has its own characteristics and design requirements. In addition, we identified a category of application that interacts directly with the Internet, without needing a browser as an intermediary. We called these Net applications.

Introduction

Metaphors can help users achieve their goals in using a system. The *desktop metaphor* has been very successful in making computer systems accessible to the general public. On the other hand, metaphors can lead users astray, especially when they are mixed together in one user interface. Users find it difficult to determine which controls apply to which part of the application.

Web browsers, the primary tool for accessing the World Wide Web, use the *page metaphor*, which is appropriate for browsing static text with hyperlinks. This is the task that browsers were designed for. As the Web expanded into transaction systems and applications, the page metaphor has been mixed with application metaphors. This has created confusing environments for users. Advances in technology, such as frames and dynamic HTML have further added to the complexity of the situation, creating potential pitfalls for designers and ultimately usability problems for users.

Classification of Web Usage

As we were discussing usability issues, we discovered that usage had a strong bearing on the navigational problems we set out to discuss. One of the first things we did in the workshop was to classify uses of the Web. We determined there were three general classifications into which all activities fell:

- > Browsing
- > Performing Transactions
- > Running Applications

Each category is described below and some of the usability problems are mentioned. Some design recommendations are also given. (The problems mentioned for browsing apply to the other categories as well, but they are not repeated.)

Browsing

Browsing is when a user clicks on a link to go from one page of static information on the Web to another. This is what browsers were designed for. Not surprisingly, this results in the least number of usability issues.

Orientation

The most common usability problem when browsing is that users don't always know where they are or how to get where they want to go. It is important to consider that the user may have entered a site via link or a bookmark that isn't the main entry point. It is useful to have information about the structure of the site and the user's relative location on every page in a site.

Frames

There are several problems with Web sites that use frames. The first shows up when users follow a link from a frame to another page that contains frames. The user can end up with frames inside of frames inside. This can be very disorienting.

Another problem with frames is that you cannot always precisely bookmark them or link to them. The issue is that there is currently no way to reference an anchor inside of a frame inside of a page. This makes it impossible to cross-reference in a document developed with frames. Additionally, these documents are difficult to print.

Frames are not inherently bad, but designers must be aware of their limitations and design carefully around them.

Dynamic HTML and the Page Metaphor

Dynamic HTML, such as Java applets, allows content to be generated dynamically for a single URL. The problem is that this looks like a series of Web pages to the user. The user expects to be able to use the browser navigation controls to navigate through the content, but cannot. When the user hits the Back button, the page prior to the one with Java is re-loaded, although the user might expect to just back up in the Java navigation.

This is an example of where the page metaphor breaks down because the content imposes a different model of navigation. Unfortunately, users may not recognize the difference.

Additional Browser Window for Viewing Linked Page

Bringing up additional browser windows for viewing linked pages can cause confusion for users. Typically, users don't realize that a new window has been launched. They only notice that the back button doesn't work any more. To avoid this confusion, carefully consider whether a new window is necessary. Also, ensuring that the new window is offset from the old one can help users understand what is going on.

Performing Transactions

Transaction systems involve getting information from the user and providing information to the user, typically interacting with a database on the server. We discussed two types of transaction systems:

- > those that change the database in significant ways (such as e-commerce sites for purchasing goods and services online), and
- > those that make inconsequential changes to the database (such as search engines).

Transaction systems typically have state information that must be maintained.

Orientation

The orientation issues that apply to browsing apply here as well. In addition, users are often confused about where they are in the transaction. It is important to provide clear information to the user on how they can correct errors, cancel the transaction and when they have completed their transaction.

The Page Metaphor

The page metaphor begins to breakdown with this type of system. An important design consideration is that not all transaction systems are good candidates for being in an HTML page in a browser. Our rule is: “When in a browser, behave like a page.” If the page metaphor really isn’t working, get the transaction system out of the browser by launching a separate window.

The Back Button

The back button can be especially problematic for transaction systems. Hitting the Back button loads the previously loaded page from the cache. The user can then re-submit the page accidentally, thereby ordering more products than intended or viewing information that is out of date. Since the server and client are loosely coupled, the server is unaware that the client has reloaded a cached page via the back button and can therefore not warn the user of potential problems.

Also, novice users may perceive the Back button as implementing undo when it is strictly navigation.

When building transaction sites, the functionality of the Back button should be taken into consideration with regard to users’ expectations and required functionality

Running Applications

In discussing applications in our workshop, we defined two types of applications: *Web applications* and *Net applications*. *Web applications* are essentially desktop style applications whose primary reason for being viewed in or via a browser is ease of deployment and are discussed here. By contrast *Net applications* use the Internet, but not necessarily the Web. Net Applications are discussed in more detail in the next section.

Page Metaphor

The complexity of these Web applications often results in a conflict between the page metaphor and the application navigation. When this occurs, the Web application should be run in a browser window that has been stripped of its browser controls. This requires opening a second browser window from the first one that was used to navigate to the application. A word of warning: doing this can result in the usability problems described in the next section.

Browser Dependency

Launching applications from a main browser window into a second browser without browser controls as recommended in the previous section is currently the best technique of making applications available via the web without mixing metaphors. This technique, however, has an unfortunate side effect. From the perspective of the user, the second browser window isn't a browser window at all. It is the application window. The dependency between the two windows is not clear in the interface. The user may intentionally kill the browser and thereby inadvertently kill the application.

A technological improvement that was discussed in the workshop was that of creating a plug-in model where the browser could spin off an independent browser process or virtual machine for each application that was launched. At present, none of the browsers provide this functionality.

Net Applications

Net application is the name we gave in the workshop for applications that interact directly with the Internet without the browser as an intermediary. The application takes care of any backend protocol work. This type of application is interesting to this discussion because it avoids most of the usability issues discussed in the sections above by not using a browser. Since the browser page metaphor is gone, there is much less confusion for users.

In the future, the desktop-metaphor of computer use may move toward a "Webtop metaphor." This may provide users new ways of seeing, arranging and accessing information and applications on the Net.

Future Work

Ideas for future work on this topic include:

- > further refinement of the classification of web usage,
- > additional work on metaphors and how they interact,
- > refinement of design recommendations,
- > validation of design recommendations with usability testing, and
- > application deployment strategies.

Organizers

Special thanks go to our organizers, Hal Shubin and Ron Perkins, for their efforts in organizing and leading the group.

Position Papers

The position papers from the workshop are available at <http://www.user.com/chi98/workshop>.

Participants

Nancy Bayer, Parametric Technology Corporation
Ward Clark, Open Market
Carola Fellenz, Sun Microsystems

Tom McLaren, McLaren Associates
Jakob Nielsen, Nielsen Norman Group
Jarmo Parkkinen, TAI Research Centre, Helsinki University of Technology
Ron Perkins, Design Perspectives
Hal Shubin, Interaction Design

About the Authors

Carola Fellenz is a human interface designer with Sun Microsystems, Inc., working on the Sun's Web browser. She has previously designed web-based administration tools, Unix-based development and administration tools, informational Web sites and Web-based small financial applications.

Jarmo Parkkinen is researcher in the Usability Group of the Laboratory of Information Processing Science at the Helsinki University of Technology. He has worked with over 20 companies conducting usability tests on a wide range of products varying from embedded products, tools for specialists and Web solutions. He has also taught courses on usability and consulted on software development.

Hal Shubin is a founding principal and vice-president of Interaction Design, Inc. Hal has over twenty years experience in software, first as a software engineer in the field of Artificial Intelligence and later in user interface design and usability testing.

Authors' Addresses

Carola Fellenz
Sun Microsystems, Inc.
901 San Antonio Road, MS UCUP01-209
Palo Alto, CA 94303
carola.fellenz@sun.com

Jarmo Parkkinen
Helsinki University of Technology
PL 1100
02015 TKK
Otakaari 1
Espoo
jarmo.parkkinen@hut.fi

Hal Shubin
Interaction Design
78 Chilton Street
Belmont, MA 02178
hshubin@user.com