



## Navigation in Web Applications

Hal Shubin, Interaction Design, hshubin@user.com  
Margaret M. Meehan, Text Matters, margaret@emdash.com

Originally published by the ACM in *interactions* magazine, November 1997, issue IV.6  
© Copyright ACM, Inc., 1997

15 November 1997

The World Wide Web is a new application development platform. Applications on the Web range from having users complete simple tasks, such as filling in a form, to more complex undertakings, such as collaborating on a project.

The tools available for creating Web pages allow anyone to design a Web application. But the Web – unlike Windows, Macintosh, and Unix – has no standards for designing applications. People come to the Web with expectations based on using the other platforms, and these expectations make up their conceptual model of using an application on the Web.

Frequently in Web applications, users get lost because the model of navigation on the Web differs from that in other applications. This difference causes confusion among users, but application developers can use a few techniques to help people successfully use the application.

This paper discusses several common navigational problems and techniques for avoiding them in designing Web applications. Although the focus is on applications rather than on purely informational sites, you can use these guidelines for designing anything on the Web.

## Knowing Where You Are and Where You Are Going

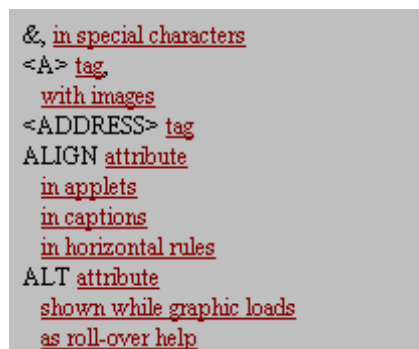
It is easy for users to get lost in Web applications because they have less structure than other applications. On the Web, designers can create pages of any length, with any number of links to other pages. If users get buried in information or lost on a side trip with no signposts or landmarks, they may find it hard to get back to what they were doing.

You can solve this problem by arranging information thoughtfully. On an informational Web site, where surfing is the main activity, it may be beneficial to link all pages on the site to all others, or even to provide links to relevant information on another site. In an application, however, you need to help users either complete a task, by minimizing distractions, or make a clear choice to quit the task.

Page design also helps users keep track of where they are. Use clear and consistent navigational aids – page names, logos, banners, icons, background colors – as visual clues for users. Arrange choices in a clear structure, such as a hierarchy or star pattern, so that users can understand at a glance where they are and where they can go from here.

URLs are important, too. Some applications do not have unique URLs for all pages because of dynamically-generated HTML or a bad use of frames. In this case, clicking the Reload button loads something other than the current page, losing unsaved data and causing confusion for users. To prevent this, write your application so that each page has a unique URL.

Hypertext indexes and help systems provide quick access to information. With the proper use of frames, a help system can show the help text for a task in one frame while the user performs that task in another. Likewise, an application can display an index in one frame, as shown here, and show the target page in another frame when a user follows the link.



A Web application index has page links instead of page numbers

## Losing Users and Losing Work

Sometimes designers make it too easy for people to leave a site. For instance, having users click on advertisements may provide revenue for your site, but consider the tradeoff of losing users and losing work. Similarly, a site that offers attractive links like *Increased performance* in the context of performing a task, such as registering for a mailing list, may take users away from the task. The result may be that users never sign up and the site owner loses a reader or a customer. External links are more appropriate for information sites, where surfing is the primary activity, than for applications, where users have a task to complete.

In the following example, toolbar commands could distract the user while performing the main task of configuring a new printer, thereby creating the possibility that the user may not complete the task.

<a href="#">Check alerts</a>	<a href="#">Printer status</a>	<a href="#">Check supplies</a>	<a href="#">Send email</a>
<b>NEW PRINTER SETTINGS</b>			
<b>Name:</b>	<input type="text" value="Jake's printer"/>		
<b>Password:</b>	<input type="password" value="*****"/>		
<b>Default tray:</b>	Tray 1 ▾		
<b>Stacking:</b>	On ▾		
<b>IP address:</b>	<input type="text"/>		
<b>IP name:</b>	<input type="text"/>		

Links outside of an important task can send users on detours

A better way to structure an important task is to have only the relevant commands on the page and to require users to either complete or quit the task before doing anything else. In this case, the toolbar would only have once choice labeled "Cancel".

### Quitting a Web Application

While non-Web platforms provide clearly marked exit paths in applications, usually a Quit or Exit command on the File menu, Web applications do not have a way to stop "running" the application without closing the browser. Yet closing the browser contains no offer to save work in progress. Commonly, Web users leave an application by following a link to another page or site. Using the browser's Back button returns users to the application's pages, but not necessarily in the same state as when they left.

Seeing a clearly defined Exit link might help users remember to finish what they are doing before leaving an application. It could check that a task is complete and ask the user to confirm exiting. The benefit to the user increases by using such a link if you offer a reward, like a review of the changes made during a session or a receipt.

Here is a simple guideline to keep from losing users and losing work: Do not let your users leave before they are done with your site or before you are done with them. If it is important to provide links out of the product, try one of these techniques:

- > Have a special exit page showing external links and other useful information.
- > Put links in safe places so users do not lose work if they click and leave. Remind them to bookmark the page and return.
- > Use frames when users click an external link. One frame shows the target page, the other is a reminder to return.

### Separating the Browser and the Application

Navigation is an essential characteristic of the Web, much more so than on other platforms. Web authors can link any page to any other, and users can pursue information at will. The importance

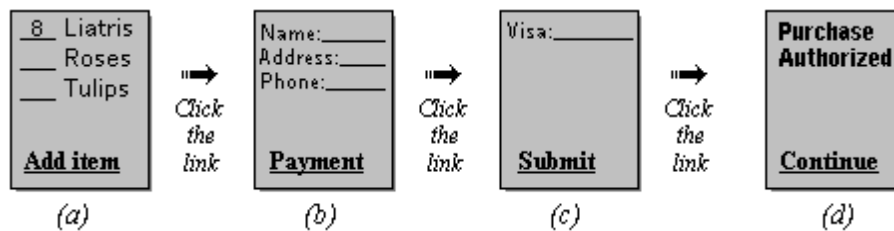
of navigation on the Web serves to demonstrate the separation between an application (such as a search engine or online store) and the browser (which simply shows the data on each page).

This split is perhaps clearest with the Back button, which is part of the browser, not the application. If the user clicks Back three times, the application does not "know" it. Even though it might seem that backing up over those pages is undoing the task they represent, the user is simply seeing the pages fly by in reverse order. The button redisplay the pages, but does not typically undo the state of the application. Similarly, clicking the Forward button is not a Redo command.

Thinking of the data as separate from the controls used to display it is new, because applications on other platforms combine data and controls in a seamless whole. Users of the Web, especially novices, do not recognize where the browser ends and the application begins.

### Online Commerce Example

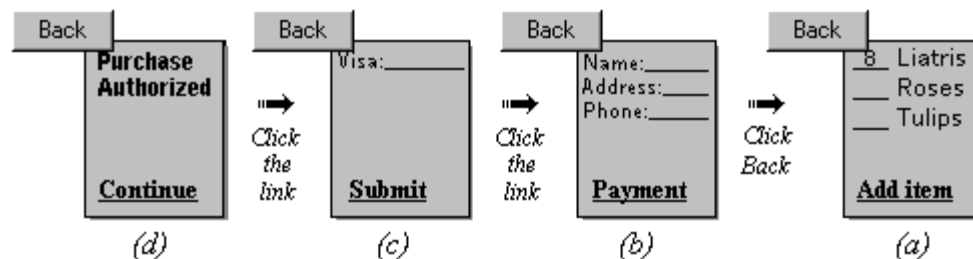
Fred is ordering plants from an online nursery. (The arrow  $\Rightarrow$  indicates clicking on a link and moving to the next page.)



A complete purchase in a Web application

- > Step (a) Fred collects the products that he wants to buy by clicking any of the Add Item links in the catalog.
- > Step (b) Fred checks the order form for correctness and clicks on the Payment link when done with the form.
- > Step (c) Fred enters credit card information and clicks Submit
- > Step (d) Fred sees that his purchase was approved.

Before leaving the store, Fred realizes that he forgot to buy the dibble that he needs to plant the tulip bulbs, so he uses the Back button to return to the pages of the catalog. As he goes back from (d) to (c) to (b) to (a), he realizes that he's backed up over all of the ordering information.



The purchase pages unwinding as the Back button is pressed

Now he wonders about the state of his purchase:

- > Will his credit card be charged?

- > Will he get the items he has ordered?
- > Does he have to reorder?
- > When he clicks on the Add item link again, will he get an empty order form, or the one he has been using?

It is difficult for users to answer these questions, especially when they first encounter the situation. Designers should encourage users to kick the back habit. Instead of encouraging or requiring the use of the Back button, applications should have navigation buttons to take users directly to where they need to go.



Kick the back habit with clearly labeled buttons

### Lack of context

Unlike Macintosh and Windows programs, a Web application does not bring up a dialog box to ask questions or get information to complete a subtask. On the other platforms, the main window remains visible behind a dialog box and provides context. This model allows users to remember how this subtask fits into their work. Web applications need to find a way to display information in context rather than requiring users to remember it.

A simple technique for Web designers is to redisplay the page that caused an error along with the error message, instead of displaying the message on a page by itself. Even though they are looking at a new Web page, users perceive it as the same one. They do not have to go back and forth between pages to interpret the error message.

By showing new information in context rather than in isolation, users have less to remember and do not need to return to a previous page. This requirement is no different on the Web than on other platforms, but it can cause more problems on the Web because of the time it takes to go back to the original page and figure out the context.



An error message shown in context rather than on a separate page

## Delays caused by network connections

Navigating from one page to another in a Web application takes longer than moving from a main window to a dialog box in a traditional application. Delays inherent in navigation add to users' cognitive load because they have to keep context information in their minds while waiting. It also makes their work take longer than they expect.

Delays arise from two sources: network traffic and computation. If the Web server is on a network, especially if the server is distant, it simply takes more time to serve pages to the browser. Furthermore, Web applications typically depend on servers for computation, such as calculating shipping charges or re-sorting a list of names and messages, which adds to the wait for users.

While these delays may seem small, computer users have come to enjoy faster response time than is currently available on the Web. Moreover, users do not know what to expect because there is no way to provide feedback through a progress indicator when operations take a long time to complete.

To help minimize the delays caused by working on a public network, try these techniques:

- > Create an intranet, where you control the infrastructure, instead of using the Internet.
- > Use smaller graphics and fewer pages reduce the time spent waiting for pages to download.
- > Make navigation clear to reduces errors and overall task time.

## Summary

When a particular company owns a platform, they can set standards and publish guidelines for application development. The freedom and diversity inherent in the Web, however, break that model. The consequence is a wide variety of guidelines – some good and some bad. How do we decide which guidelines to follow in designing applications? By remembering the user's needs.

Web designers can help users navigate through their applications by applying traditional design guidelines with the particular requirements of the Web in mind:

- > Arrange information carefully so users always know where they are in the application.
- > Do not tempt users to interrupt their work by following links to other pages either in the application or on another site.
- > Hide the split between the browser and the application by including navigational controls in the application.
- > Preserve the context of an interaction so users to not need to recreate it in the middle of their work.
- > Minimize network delays through clear navigation and reduced download time.

\* Copyright © 1997 by the Association for Computing Machinery, Inc. Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Publications Dept, ACM Inc., fax +1 (212) 869-0481, or [permissions@acm.org](mailto:permissions@acm.org).