Interaction Design

78 Chilton Street Belmont Massachusetts 02478

User models as a basis for Web design

A position paper for *Organizing Web Site Information*
24 February 1999

Hal Shubin
Interaction Design
www.user.com, hshubin@user.com
617 489 6595

Introduction and background

I have been working as a consultant on Web sites for about four years, doing expert reviews, usability testing and original design. This includes information sites (where the focus is on finding and reading information) as well as applications (where the focus is on transactions). Most of my work has been with applications, including printer and general network management, document management and e-commerce.

The same issues of organizing and categorizing information arise in all types of Web sites. In fact, many of them arise in desktop applications. Even though some types of applications exist on the Web and the desktop, the Web presents new reasons to be concerned about organizing information. The primary reason for this is that desktop applications typically use a form metaphor while Web browsing uses the less-familiar page metaphor [Fellenz 1998]. Web applications are somewhere in the middle.

The Web does not provide the same standard organization as found on the desktop, where each application has its own menus and toolbars. This requires each designer to do more work to achieve the same level of organization. (The menus and toolbars in a browser relate to the browser, not the application running in it. See [Shubin 97] for more details.)

Another reason for focusing on information architecture on the Web is the cost of waiting after each click. Part of that wait is due to graphics, part to the delays imposed by servers. As described in [Rosenfeld 98], the graphics and page layouts available on the Web allow people to "enjoy the overall atmosphere and experience created by the site." We can't reduce the enjoyment by causing delays and confusion due to poorly organized sites.

voice 617 489 6595

fax 617 489 7395

hshubin@user.com

http://www.user.com

Organizing to support appropriate user models

My focus in this paper is organizing a Web application based on an understanding of the users' model of the tasks involved. A user model is a reflection of users' understanding of a product. It includes how they perceive the objects and activities involved in the operation of the product. When I write a user model document, I include descriptions of a few typical users and some scenarios representing what tasks they will do. It also describes from the users' point of view the concepts and objects involved.

The points covered in this paper include focusing on typical tasks, understanding tasks from the users' point of view, helping people remember where they are in a task and knowing which user model to choose. I will discuss three products that I've worked on recently:

> Rezz, a hotel reservation system. This application works as a sub-site associated with Web sites for conferences and other events. As with many Web applications, Rezz has many types of users. Experience in the hotel industry, use of a previous prototype, information from call centers and usability testing helped us understand the user model for the product. (The client agreed to my writing this paper, but the product is under development, so I removed identifying information, including client and product names.)

> PrintWizard, an application that I designed for Digital Equipment Corporation. It allows system administrators to manage printers on a network. The application's pages presented printer parameters to be set or viewed. The user group here is more cohesive: system administrators, who all share a reasonable understanding of the task.

> Boston.com, a large information site that I reviewed. This online magazine provides information about Boston and is produced by a subsidiary of the Boston Globe newspaper. This site has the most variety in its user base. In the brief review project I did, I used general ideas about design and usability on the Web.

Speeding up access to typical tasks

A type of organization common to both browsing and transactional sites is making sure that people can find things quickly. By creating and analyzing scenarios based on typical users and their tasks, we can make it easier for people to do what they do most often. We had a good understanding of which tasks people would be doing most often in Rezz.

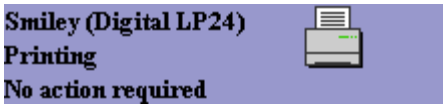
The original prototype required users to start by picking from a menu of three choices: new reservation, modify or cancel. We knew that 50% to 60% of users would be making new reservations, so I made that the default choice – starting the application takes users right to the New Reservation page.

The top of the first page in Rezz, showing event information and the toolbar

The toolbar offers a choice to modify or cancel a reservation, so there is no extra work imposed for those tasks. Because the majority of users will be to make new reservations, this organization saves one page transition in the typical case.

In PrintWizard, we had the challenge of presenting potentially hundreds of attribute/value pairs on a page in an understandable way. Examples include the current versions of software on a printer, and the list of SNMP network settings. Different people will use the product in different ways, but one of the most common tasks is to find out the status of a printer.

We optimized access to this information by placing basic printer status information at the top of every page, whether it was displaying TCP/IP parameters or toner levels. Included in this area is a status icon providing a quick graphical indication of the printer status.



Printer-status information is on every page. This icon shows that the printer is printing.

Designing tasks from the users' perspective

In the original prototype of Rezz, users who wanted to make certain changes to a reservation had to cancel it and make it over again. This seemed to violate a basic part of the user model of hotel reservations. Making changes through a live reservation clerk would not require a cancellation, so I didn't think the online version should.

This seemed to be a classic case of letting the architecture show through to the UI. The underlying system did sometimes have to treat a modification as a new reservation, but users did not have to know all the gory details. The final design allows users to make modifications and handles making and breaking reservations in the background.

Another aspect of supporting different user models in Rezz was task order. Some conference attendees start out the hotel reservation task knowing when they will be travelling, and want to shop for hotels based on those dates. Others work the opposite way, for example knowing that they want to stay at the event's headquarters hotel, but are not sure which dates they want their rooms. Still other attendees know both items when they start.

The path from the first page is any of these:

1. Select a hotel, then on the next page select dates for any of the available rooms.
2. Select stay dates, then select from the list of hotels and rooms available on those dates.
3. Select dates and the hotel, and then pick from the list of rooms available at the hotel on the specified dates.

This is more than a superficial product requirement, because it allows people to work with the product the way they think about the task, rather than having to learn to work the way the developers and designers think about it. The program was designed to work for all three groups. Usability testing has not yet shown any positive or negative responses to this; I hope to do more when implementation is complete.

Helping users know where they are in a task

Users always want to know what they have accomplished so far. I worked with a graphic designer to lay out pages in Rezz to support this need in two ways. The bottom part of the tool-

bar shows the subtasks in the current task, with the current one highlighted. (See the example above.) Another technique was to carry along a summary of the work done so far.

For example, in the Payment Information page, the summary area at the top of the page showed this information:

Hotel information	
Hotel	Segue Motel
Guests	Fred J. Fish, September 30, 1998 – October 4, 1998 Joan Jett, October 1, 1998 – October 4, 1998
Rates	Sep 30, 1998 $89 (1 guest) Oct 1, 1998 $120 (2 guests) Oct 2, 1998 $130 (2 guests) Oct 3, 1998 $130 (2 guests) Oct 4, 1998 $120 (2 guests) Daily rate includes $x/day for 2nd occupant. Additional charges may apply for additional occupants in the same room. Taxes and fees are not included in rate. Rates are shown in $US.
Total	$589

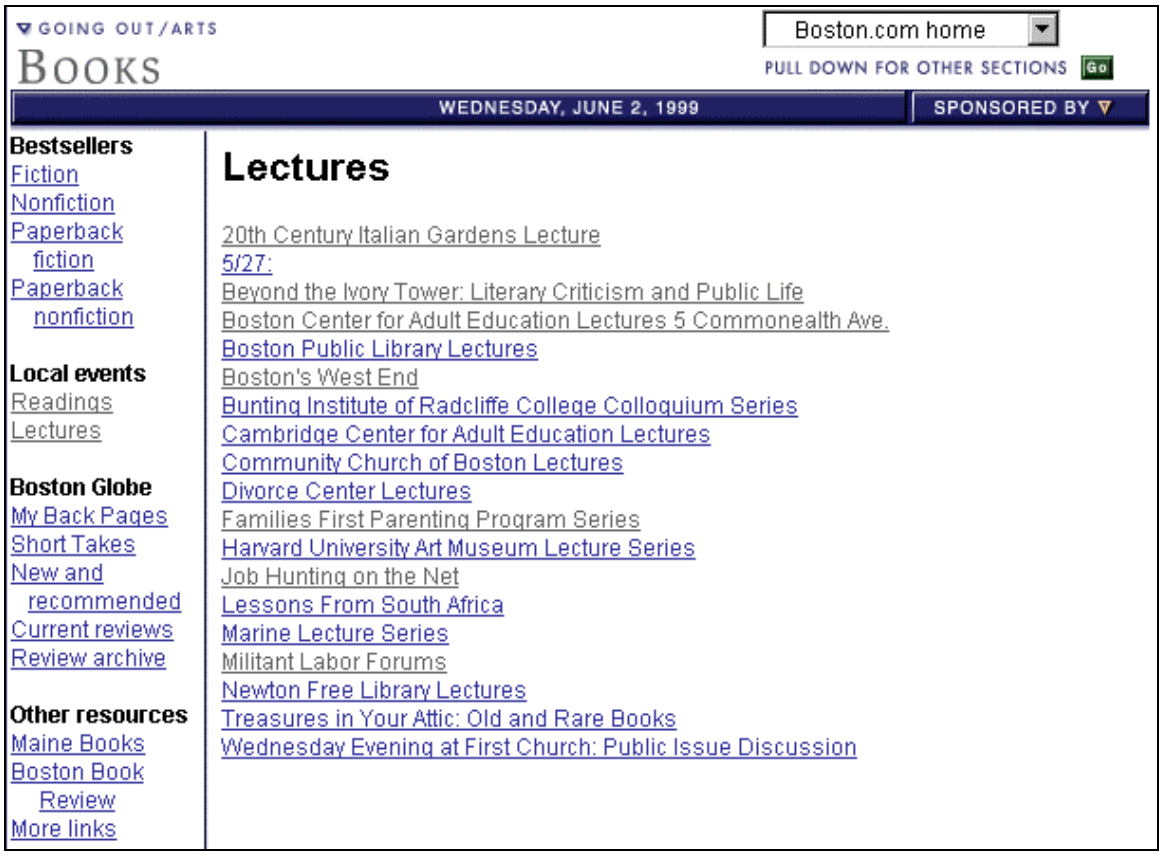
Each page summarizes what the user has done so far. This is a worst-case scenario where the rate and number of occupants both change during the stay.

Two participants in the Rezz usability study said that they "got the rhythm of the page" because of the consistent page layout. One person said that it is "keeping you connected to the process, building information with the computer."

Choosing the right user model: page layout

Recent research has shown that people read Web information differently from how they read on paper. Most people scan rather than read word-by-word. [Neilsen 97]

The Boston.com site seemed to be using a newspaper user model rather than a Web model in laying out their pages. Recent redesigns are now more Web-like, with lists of links instead of long columns of undifferentiated text:



The new Web-oriented layout for listings at Boston.com

Summary

An analysis of who will use a Web application and what they will do with it provides important information for designers. It allows us to optimize an application for typical tasks and help them get their work done.

References

Fellenz, C., Parkkinen, J., and Shubin, H. *Web Navigation: Resolving Conflicts between the Desktop and the Web, a CHI 98 Workshop Report.* http://www.user.com/chi98/web-and-desktop.htm. Accessed Feb 99.

Nielsen, J. *Alertbox.* http://www.useit.com/alertbox/9710a.html. Accessed Feb 99.

Rosenfeld, L., and Morville, P. *Information Architecture for the World Wide Web.* O'Reilly & Associates, Sebastopol, CA, 1998.

Shubin, H. and Meehan, M. "Navigation in Web Applications," *interactions,* ACM, NY, NY, Nov+Dec 1997, IV.6.