

Web User Interface Design: Forgotten Lessons

Uttara Nerurkar, Infosys Technologies

A number of prescriptions are in vogue for Web user interface design (let's call it WUI, in line with GUI). However, Web site usability continues to be a serious issue.¹ WUI design methods are quite different from their GUI precursors. The latter have matured considerably over some time and have been proven to yield highly usable interfaces.² While the Web paradigm is apparently quite different from the GUI one, are there similarities that we can exploit to improve WUI

design techniques? I try to show here how I believe this can be achieved.

Let us take a quick overview of one representative methodology each for WUI and GUI design.

WUI design methodology

A popular Web design technique³ is based on building an information architecture, following these basic steps:

1. Define the site's mission and vision. Also, envision your intended audience.
2. Determine the site's content and functionality.
3. Define the organization of information on the site, including navigation, labeling, and search systems. Specify these in terms of Web page hierarchy diagrams, or *information architecture blueprints*.
4. Build and test. Preferably, test the site's usability by observing members of your intended audience perform specific tasks on the site.

Analyzing the Web design technique a bit more closely, we can see a few gaps in the scheme. At the second step, the link between

audience and design breaks. While the design is *guided* by the expected audience's characteristics, there is no *direct* way of defining user requirements, especially in terms of content. This invariably leads to an almost imperceptible shift in the focus from what the user would like to find on the site to what the owner would like to display. Similarly, at the third step, the organization of content has no explicit reference to user requirements or expectations. It usually involves a logical partitioning of content and a drill-down architecture for details. For example, on an entertainment site, music and movies typically appear on separate pages because they are different *types* of items. However, would it be better if the movie's soundtrack appeared on the movie page? We need to understand the user to answer this question. Indeed, an expert site designer recognizes this and often flouts the "logical partitioning" rule. But this leaves us to grapple with the question, "If not logical, then what?" GUI design gives us some pointers, as I shall reveal later. At the fourth step, the user focus explicitly returns to the design cycle—a bit late to avoid major design flaws in the earlier steps.

The methods used to design Web sites and graphical user interfaces differ considerably, but the author argues that it is possible to improve Web design methods by applying well-established GUI design principles.

The GUI methodology's main strength lies in the mapping between users and tasks and that between tasks and presentation elements.

GUI design methodology

For GUIs, a proven method for improved interface usability is user-centered design.³ It broadly involves the following steps:

1. Set up user types.
2. Define tasks, or operational user scenarios, for each user type.⁴
3. Design the user interface by specifying presentation elements to complete these tasks.

The GUI methodology's main strength lies in the mapping between users and tasks and that between tasks and presentation elements. These two mappings provide not only a direct aid in focusing the designer's attention on the user, but also help in discovering the application's usability, such as by revealing the number of steps required to complete a task.

How different are these paradigms?

Let's step back a bit from the interface design techniques and analyze the differences between the WUI and GUI application paradigms themselves. Major differences between them include the following:

- User characterization is more difficult on the Web because it typically addresses an unknown audience. For applications, however, the user community is usually well defined.
- Applications are typically task-centric, whereas the Web is still largely information-centric.
- Customers are not stuck with a Web site the way they might be with an application or product they have purchased. At the slightest difficulty, the surfer tends to move on to another site, maybe never to return again.
- Navigation between Web pages is much more flexible than that between forms (or screens) of an application. Thus, you can reach a particular Web page by any number of navigation paths provided on the site. In contrast, in the GUI application, the routes to a particular screen are extremely limited. Indeed, most screens have only one path to them.

The first distinction reveals a hurdle in defining the Web paradigm's user types. Web design experts have suggested the creation of personas or detailed characteriza-

tions of users belonging to different demographic profiles to predict the surfer's behavior on the site. Other design experts define relevant user characteristics and predict user behavior based on various combinations of these characteristics, such as expert versus novice.

The second issue pertains to a problem in defining tasks for a Web site. In the information-centric paradigm, search is the key operation. It does not seem to fit the definition of a task because it is too flexible and vague. For this reason, the Web techniques assume that search will be made simpler if the content is organized logically and proper labeling and navigation schemes are provided. So why do so many content organizations fail on this account? I contend that organizations focus on the content they want to share with the world and not on what the world would like to know about them. I suggest that we take each of the user types defined earlier and set up tasks for them such as "Find product information" and "Find company profile" based on our guesstimate of user interest. Not only will such a scheme create a mapping between users and their tasks, but it will also provide direct guidance to how to organize content on the site. If users cannot accomplish the "Find product information" task within three page visits, the site's design must be improved. Evidence that we can enumerate the search operation in this way comes from the WUI design method's fourth step. We can define the same tasks given to the tester to perform on the site *before* we design the site.

Difference three (customer loyalty) has two aspects. One, aesthetic appeal, is outside this discussion's scope. The other aspect is usability, which is the subject of this article.

Navigation flexibility is undoubtedly a nonissue in the traditional approach. Form navigation in older applications was normally so simple that it was unnecessary to model it explicitly using navigation maps. On the other hand, a way to model page navigation is critical for any Web design.

I suggest bringing together the best of both worlds to yield a new and improved Web design methodology, thus:

1. Define the site's mission and vision.
2. Envision the intended audience and model it as a finite number of user types.

3. Specify the tasks you expect each user type to perform. Include here the information they would be interested in viewing as "Find <content>" tasks.
4. Design the Web pages, including content and presentation elements, by building a mapping between the page and the task. The search or "Find <content>" tasks will then define the content required on the pages. Some tasks will span across a number of pages, thus automatically creating a page hierarchy and navigation path. Complete the page hierarchy and navigation paths for all remaining pages.
5. Design the labeling and search system so that it is optimized for the tasks.
6. Test the design.

The essentials of a user-centered interface design approach are definitely applicable for the Web. However, WUI gurus advocate a design path that

About the Author



Uttara Nerurkar is a senior research associate at Infosys Technologies Limited, a CMM Level 5 software services and consultancy company headquartered in Bangalore, India. She has worked in the software development area for the last nine years. Her research interests include software architecture, security, and user interfaces. She has a BTech in chemical engineering from Indian Institute of Technology. She is also a professional member of the ACM. Contact her at Software Concept Laboratory, Infosys Technologies Ltd., Electronics City, Hosur Rd., Bangalore 561 229, India; uttara@acm.org.

is totally divorced from the GUI approach. Applying some "ancient wisdom" from the GUI era could lead to more usable Web sites. ☞

References

1. K. Scoresby, "Win Consumers with Better Usability," *e-Business Advisor*, vol. 18, no. 6, June 2000.
2. A.R. Puerta et al., "MOBILE: User-Centered Interface Building," *Proc. Conf. Human Factors and Computing Systems*, ACM Press, New York, 1999.
3. L. Rosenfeld and P. Morville, *Information Architecture for the World Wide Web*, O'Reilly, New York, 1998.
4. B. Schneiderman, *Designing the User Interface*, Addison Wesley Longman, New York, 1998.

CALL FOR ARTICLES

Initiating Software Product Lines

The product line approach to software development is emerging as a technique for organizing the large-scale production of a number of related software products. Adopting the product line perspective on software development affects the usual way of doing business at the organizational and technical management levels as well as at the software engineering level.

Many companies are investigating the potential benefits, establishing pilot projects, and baselining these initial efforts. They face a number of questions:

- What types of planning are necessary prior to initiating the product line?
- How can existing products be incorporated into the product line?
- What levels of resources are required to support the early phases of a product line project?
- Are there lightweight approaches that can streamline the migration to product lines?
- How much of the requirements and the architecture must be determined initially?
- What must be done to existing development methods and processes to adapt them to the product line environment?
- How long does it take to realize the benefits of a product line approach?

We seek original articles on methods, tools, experiences, and techniques related to initiating software product lines. A full list of suggested topics is at computer.org/software/edcal.htm. Work should be presented in terms that are useful to the software community at large, emphasizing approaches taken and lessons learned.

Please submit two electronic copies, one in RTF or Microsoft Word and one in PDF or Postscript, by 1 March 2002 to IEEE Software at software@computer.org. Articles must not exceed 5,400 words including figures and tables, which count for 200 words each. The papers we deem within the theme's scope will be peer-reviewed and are subject to editing for magazine style, clarity, and space.

Please contact any of the special issue guest editors for more information about the focus or to discuss a potential submission; please contact *IEEE Software* (software@computer.org or computer.org/software/author.htm) for author guidelines and submission details.

Guest Editors:

John D. McGregor
Clemson University
johnmc@cs.clemson.edu

Linda Northrop
Software Engineering Institute
lmn@sei.cmu.edu

Salah Jarrad
Panasonic Wireless Design Center
sjarrad@email.com

Klaus Pohl
University of Essen
pohl@informatik.uni-essen.de

Submission Deadline: 1 March 2002
Publication Date: September/October 2002