Usability Engineering for the Web

Keith Instone

Abstract

As ease of use becomes more important than being "cool" on the World Wide Web, usability engineering techniques will let developers create more usable Web sites by helping them know their audience, valuating their user interfaces, and redesigning their sites based on user feedback. Understanding the tradeoffs between standards and new Web technologies is essential in creating usable sites.

Introduction

The ability to be "cool" on the World Wide Web --that is, taking advantage of the latest and greatest features--has been an important factor for those who want to stand out from the crowd. Lately, however, users have made it clear that it's not enough to simply use the newest technologies, such as frames, JavaScript, animated GIFs, or Shockwave. Many sites have been forced to provide alternatives to their frame designs because visitors have found them too hard to use [1].

Usability engineering techniques can create more usable Web sites by helping developers focus on the users rather than the technology. Understanding users' needs and the tasks they are trying to accomplish is steadily becoming more important than adopting the latest whiz-bang feature. Of course, if the latest technology fulfills a need, it should be used; but incorporating this technology just to be cool is becoming less and less acceptable.

In the very competitive world of the Web, usability will become crucial to the survival of sites (and their corresponding businesses). The value of usability engineering was put in perspective when Robert Hertzberg, WebWeek editor, advised that the observation of users "may be more valuable than what most businesses get from a $10,000-a-day consultant" [2].

Improving usability is not a black art: it can be accomplished by heeding common sense advice and applying techniques that help us understand and meet the needs of users. A good introduction to this topic is Jakob Nielsen's Usability Engineering [5]. Applying usability engineering to the Web is not that difficult, although the Web does introduce a few interesting problems--and solutions. Darrell Sano's Designing Large-Scale Web Sites [10] discusses many Web usability issues, such as limitations and constraints for Web publishing and structuring the information space.

What follows is a review of usability techniques and methods, seen in light of the constraints and tradeoffs of the World Wide Web. The best place to see usability engineering applied to the Web is the Sun-on-Net project [6], where usability testing, iterative design, and other activities improve the usability of Sun's Web site.

Web Usability Engineering Life Cycle
The usability engineering life cycle (see the following list [5]) represents a menu of choices that can be worked into the broader development context in order to increase usability. The techniques focus on understanding users, promoting usable designs, finding usability problems, and understanding other constraints in development.

1. Know the user
   1. Individual user characteristics
   2. The user's current and desired tasks
   3. Functional analysis
   4. The evolution of the user and the job

2. Competitive analysis
3. Setting usability goals
   1. Financial impact analysis

4. Parallel design
5. Participatory design
6. Coordinated design of the total interface
7. Applying guidelines and heuristic analysis
8. Prototyping
9. Empirical testing
10. Iterative design
   1. Capturing design rationale

11. Collecting feedback from field use

Steps 1, 7, 10, and 11 (Know the user, Applying guidelines and heuristic analysis, Iterative design, and Collecting feedback from field use) are most pertinent to Web usability and are elaborated upon in the sections that follow.

**Know the User**

The most basic of usability guidelines, "know the user," can be very difficult on the Web, where international access, exponential growth, and wide-ranging demographics (from school-age children to elderly citizens) are the norm. It is also difficult to understand what people really want from this new medium, in which so much is possible. The many surveys of general Internet demographics will provide some information on "who is out there," but talking to users is still the best way to get a handle on user-
specific usability problems.

On the Web, "know the user" also means knowing the speed of their Internet connection, which browser version they are using, what plug-ins they have, and so on. Understanding these factors plays a big part in choosing advanced Web technologies and determining when they can be safely introduced.

**Applying Guidelines and Heuristic Analysis**

Many usability problems can be avoided by following the published guidelines [6] and by adhering to some general rules of thumb (see the Heuristics section below). Since the guidelines and heuristics are very general, additional usability techniques, such as empirical testing, are needed to determine the best solution for each particular problem.

**Iterative Design**

Iterative design involves a four-step process:

- Creating something.
- Testing it out with users.
- Understand the strengths and weaknesses.
- Designing a better version.

This fits very well into the Web "culture," in which a site does not have to be perfect the first time and the technology is constantly changing. Iterative design works best if it is part of the overall Web management process rather than an extra chore performed every year or so. Between iterations, one should be gathering user feedback, analyzing log data, creating rapid prototypes, and testing users.

Part of the iterative design should also be an evaluation of new technologies for the next version. Such a technology should not drive the change but rather be seen as a solution to known usability problems.

**Collecting Feedback from Field Use**

One of the easiest and most effective ways to improve the usability of a Web site is to see what real users do with it.

A strength of the Internet is the ease with which the users can provide feedback. A common practice is to place mailtos and feedback buttons on every page, encouraging users to take a moment to send the developers a note if they have a problem, a complaint, or some praise. The best user feedback can be designed into the site transparently.

Good logging of Web site usage is important for those who have to justify the cost of developing and supporting a site. As far as improving usability, logs can also indicate how people access the site, what problems they are having, and how a redesign affects usage.

**Heuristics**
Heuristic evaluation is one of the most important aspects of usability engineering: it is easy, fast, and inexpensive. It involves study of a user interface by a small set of evaluators for violations of common usability principles (rules of thumb, heuristics) so that the next iterative design can try to solve the usability problems. See the list that follows for some of the common usability heuristics [5].

- Simple and natural dialog
- Speak the users' language
- Minimize the users' memory load
- Consistency
- Feedback
- Clearly marked exits
- Shortcuts
- Good error messages
- Prevent errors
- Help and documentation

Heuristic evaluation is well suited to the Web, where everyone is in a hurry and the site can be evaluated from half-way around the world. When doing a heuristic evaluation for the Web, the common heuristics of consistency, feedback, and so on apply. But a few other rules of thumb are also useful when evaluating a Web user interface.

User Control

Because of the Web's architecture, in which browsers are responsible for interpreting HTML, developers never really know who--or what--will be processing their pages. Forcing users into particular fonts, sizes, colors, screen widths, or browser versions will generally reduce usability. Of course, designers need to have some control. But the more general the implementations and designs, the more likely we will reach satisfactory levels of usability for the widest range of users and over the longest periods of time.

Emerging Web technologies should often be avoided because they fail on this user control heuristic early in their life spans. One example is animation. To accommodate a diverse set of situations, users should be allowed to start and stop animation on their own. On one hand, animations that repeat forever are distracting; on the other, those that cannot be repeated lose their value. Until users gain adequate control over this technology on the Web, animations should be used very carefully. (Even once user control is in place, animation should be used only when appropriate [7]).

A new Web feature that will enable greater user control is cascading style sheets[1]. Style sheets force the developer to separate the presentation aspects of the page from the content, making it easier for a user to “intercept” the presentation information and insert his or her own requirements for display. Beyond selecting link color, for example, style sheets will extend the amount of control users have to
include many other items of personal taste. The largest gain, however, will be that users can specify style attributes in a standard way to ensure their own personal requirements for usable access. For example, users with poor eyesight could specify large font sizes for headers and text, overriding the font sizes specified by the authors.

When evaluating a site for user control, the following questions can be asked to determine what usability problem might arise:

- Can users override this feature?
- Can they customize to suit their tastes or needs?
- Will giving users control of this feature reduce the usability of the site?

Structure

In all hypermedia systems, some underlying structure is important to help users figure out where they are and where they can go next. On the Web, where search engines and links from other places can throw users into the middle of any site, showing this structure is even more important. Since Web browsers lack substantial navigation aids to help users discover this structure, Web authors have to do a lot of this work themselves. To evaluate structure, one can look at every major page of a site and ask the following questions:

- If a user were taken directly to this page from an outside site, what could they figure out about the rest of the site from this one page?
- Is the site "brand" present?
- Is it clear which part of the site they are in?
- Is it clear how to navigate to other parts of the site?

Figure 1 provides an example of a page that indicates its position within the overall site structure.

Design for Change

The Web is about change, so a user interface had better be able to deal with both continually changing content and design. The evaluator can ask the following questions:

- How is old content archived?
- How is new content added?
- Can this design withstand the addition of 20 times the current content?

If everything has to be introduced through a "What's new" page, the site was not properly designed for changing content. If iterative design is going to work well, the design has to be flexible enough to withstand small changes without having to be thrown away. For example, if user feedback indicates a need for an additional major section to a Web site, can the design handle it, or would the site have to be redone from scratch? Figure 2 shows the design of a page on the Netscape Web site. How easy would it
be to add an eighth section to this page? If improving the usability is too difficult, it simply will not happen.

**Usability Tradeoffs**

Usable interface design involves tradeoffs: weighing the costs and benefits to develop the best solution under the current conditions. The Web requires some very serious tradeoff decisions because of its low bandwidth, the role of browsers, and the existence of standards.

It is important to note that these tradeoffs are not meant to mandate one decision over another. What is important is to understand the pros and cons in order to make an informed decision. Sometimes usability has to be sacrificed because of cost, time, or management issues, but blindly adopting a new Web technology rather than studying the usability tradeoffs is not appropriate.

For example, third generation site design [4] improves usability by providing better page layout: paragraphs are indented, headers are close to their associated paragraphs, and there are left margins. (This is done with such "tricks" as single-bit GIFs, as shown in Figures 3 and 4, to place the text in certain places.) One of the costs, however, is that access with graphic loading turned off (to speed the process) presents a significantly less usable site; the precisely aligned text is no more, and "unloaded graphics" icons litter the screen. So if one is willing to eliminate use by one segment of the Internet population, a third-generation site might be appropriate. (Note also that several other populations of users are disenfranchised by third-generation site design such as those who have disabilities [8]). Part of the tradeoff is found in the answers to the following questions: is this a very usable site for a few users, a fairly usable site for the majority of users, or a "good enough" site for all users?

**Bandwidth**

Access speed is currently one of the major constraints on design. The best design, no matter how good, will not be seen if it takes too long to download. Until access speed is no longer a major problem,
tradeoffs involving the quality of the graphics, the complexity of the design, and the use of higher bandwidth advanced technologies will be important to understand in creating usable Web sites. In addition, developers would benefit from observing users at very slow speeds.

**Browsers**

Another tradeoff occurs because of the crucial role that browsers play on the Web. The browser does certain things that are out of the author's control--history, bookmarks, and exact page layout, for example. The fact that the developer has to rely on the browser to provide certain functionality may limit the creation of usable sites. One need only look at the sophisticated and easy-to-use interface on CD-ROMs today to realize the limits that browsers impose on the Web. Browsers often have bugs that make certain features unusable. Netscape Navigator's early implementation of frames had serious problems with Back and bookmarking, which made frames quite unusable. (Now those bugs are fixed and frames are more usable--but not necessarily usable enough for certain applications.) Since not everyone has the most current version of browsers, these bugs are still an issue when designing a site today. Finally, there are a lot of browsers, each acting slightly different. Developers need to test their sites on as many platforms as possible in order to ensure usability for the widest range of visitors.

**Standards**

Standards make the World Wide Web interoperable. The most important standard for developers is HTML (even though other standards, such as HTTP, affect usability). The latest version of HTML is 3.2 [9][2] While HTML 3.2 is not a great technological leap from 2.0 (several new features that were planned for 3.0 never made it into the oddly numbered 3.2), this latest version does represent a very important political advance for the Web. HTML 3.2 has brought the major Web vendors together for the time being, making the Web more interoperable and slowing the "tag wars" that were impeding the creation of usable Web sites across the different platforms. Knowing the HTML standard and following it when appropriate are crucial in making informed design and implementation decisions.

In general, the more meticulous we are in following the standards, the more usable a site is for the largest audience. Any variation from the standard, even if it follows "Netscape's standard," will have an effect on some part of the user population. It would help to consider the following questions:

- Is that new tag useful enough for those who can take advantage of it?
- Does it significantly affect those who cannot take advantage of it?

The best extensions are those that give added functionality for the "haves" while having no negative effect on the "have nots." A good example of this is the BGCOLOR attribute for table cells. Because they are
not part of the HTML 3.2 specification, table cell background colors are ignored by those browsers that follow the standard and have not implemented this extension. Developers need to be aware that this is indeed not part of the standard and should ensure that their sites are still usable for those who do not have access to this added feature.

Developers should also be aware of cases in which browsers are "forgiving"--i.e., those that create usable layouts of nonstandard HTML. For example, images are not allowed in PRE sections according to the HTML specification, but most browsers will generate reasonable interpretations of such markup. Designs that rely on such nonstandard use of HTML should be carefully considered.

"Following the standards" also means adhering to the spirit of the standards. For example, because it gives the desired appearance, an H6 header or other formatting is often used to "sign" a page. However, the ADDRESS tag was specifically created for this purpose ("specifies information such as authorship and contact details for the current document" [9]). More sophisticated search engines and other indexers might focus on the ADDRESS tag to improve searching. (See Figure 5.) Being able to search for pages authored by a certain person or group could be a very useful way to find information on the Web; if developers do not follow the standard as it was intended to be used, such advances cannot be made.

Designers of a corporate Intranet have it much easier than those with an Internet audience. If all users are known to have a particular browser and version, one can more safely take advantage of nonstandard aspects. Still, any nonstandard usage could tie that corporation to one particular vendor, so some intelligent decisions need to be made.

**Conclusion**

Usability engineering techniques, such as user testing, iterative design, heuristic evaluation, and user feedback, can make Web sites easier to use. Usability evaluations of new and emerging Web technologies, such as frames, JavaScript, Java and animated GIFs, are particularly important so that the tradeoffs of incorporating them into a site can be understood. Web standards, such as HTML 3.2, provide the basis of interoperability and help to ensure that designs will remain usable over time. Although excitement about the Web is still motivating users to play with new technologies, the rapidly expanding base of noncomputer-expert users, along with the increased use of the Web for "everyday" tasks and not just for fun, indicates a trend in which "what's useful" will become more important than "what's cool." Those sites that incorporate usability engineering into their development process will find that they are better able to survive in the highly competitive marketplace of the World Wide Web.


http://www.webweek.com/96Sep09/opinion/editor.html


**About the Author**

Keith Instone

Research Associate

Computer Science Department

Bowling Green State University

Bowling Green, OH 43403

instone@cs.bgsu.edu

http://web.cs.bgsu.edu/instone/

Keith Instone is the Research Associate for the Computer Science Department at Bowling Green State University. His research currently focuses on usability aspects of the World Wide Web. He maintains the Human-Computer Interaction Virtual Library and WebHCI, ACM SIGCHI’s forum on user interface aspects of the Web. His volunteer efforts include Information Director for SIGCHI and Newsletter Editor for SIGLINK.

[1] This issue of the *World Wide Web Journal* features the latest CSS1 (Cascading Style Sheets) spec

http://www.w3j.com/5/s3.instone.html 06/09/2001
from the W3C, as well as an introduction to Cascading Style Sheets by Norman Walsh.