Wolf Naegeli

Designing More Effective Webmap Viewers

This presentation will offer principles and advice for effective user-interface and functional design. It will discuss pros and cons of different implementation approaches and technologies. It will illustrate the principles and their implementation, with many examples, highlighting and critiquing good and bad aspects of mapviewers seen on the Internet, including some designed by the presenter.

USER-CENTRIC DESIGN

Careful analysis of user needs and attention to user-friendly system design are hallmarks of successful information systems. Out-of-the-box mapviewers may be adequate for simple tasks. Greater user satisfaction, higher productivity, and competitive advantages can be obtained by putting to good use the various levels of customization possible within ArcIMS mapviewers, or by building streamlined special-purpose mapviewers from scratch.

One size may not fit all

Understanding your intended users is an important starting point for application design. Are you able to answer these questions about your users?

- What equipment will they use to access your application?
- What is their level of computer literacy?
- What is their educational or professional background?
- How often will they use the application?
- Why will they use the application?

Your work will be easier if you have a clear and simple answer to each of these questions. That may be the case, if you are about to develop a mapviewer for internal use only by a private enterprise. But don't skimp on the thoroughness of your user-needs assessment. Appearances can be deceptive. In real life things tend to be more complex than anticipated.

Your work will be more challenging if you do not have a well-defined user community and when you have multiple answers to any of the questions. Don't try to shoehorn incompatible user needs into a single solution. It may not only be more appropriate but also easier and less expensive to develop more than one

mapviewer to present the same content.

Performance

Knowing what equipment and methods your users will use to access the Internet is one of the most important design considerations from a technical perspective. Mapping applications tend to be very data intensive. Therefore, the speed of the Internet connection often is the dominant performance bottleneck. Poor responsiveness of application still is the most common complaint of computer users.

If you are developing a corporate Intranet application that may not appear to be an important issue. But, what if some of the employees will need to use the application from their home or from a laptop computer while on travel?

If your application is intended for the general public, you are dealing with private homes, rural libraries, and grade schools where it is not uncommon to find computers that are 7 to 10 years old. Many users have not upgraded their software in years. In less affluent areas, we find many second and third-hand computers and hand-downs that are rarely configured optimally for their new users, and which perform even less well than they could according to their outdated specifications. In rural areas, deteriorating phone lines that won't permit Internet connections at more than 20 or 30 kbps may hamper even users who own modern equipment.

Test your applications with multiple webbrowser brands and versions, on multiple platforms and operating system version combinations, and with the help of tools such as Line Speed Tester at www.optiview.com/ and Screen Size Tester at www.anybrowser.com/ScreenSizeTest.html.

Accessibility

If you develop for some of the types of users mentioned above, you also may have to design for compatibility with old versions of web browser software and operating systems, older computer hardware, and slower Internet connections. You should not rely on anything that needs to be downloaded and installed. Don't count on plug-ins or applets that must be installed on the user's computer before your system can be used!

The more advanced the Web features you use, the fewer the visitors who will be able to use your site! The principal reasons are

- browser and platform incompatibilities
- browser misconfiguration
- missing browser plugins.

The last problem cannot be averted effectively by including a link for downloading the necessary plugin.

Many plugins work only with the most common recent webbrowsers and operating systems. Most visitors don't have the patience to wait for them to download. Many don't succeed installing them. Many don't dare, and many simply don't want to do it. Moreover, most public libraries and schools and many corporations don't allow, or prevent their users from, downloading and installing new software.

At the Southern Appalachian Information Node of the National Biological Information Infrastructure, we have experimented extensively with the ArcIMS Java Connector and have prototyped a few applications with it. In our experience more than half of the users who tried had significant difficulties with the installation of the Java Viewer or the Java Plugin or both. Personally, I have experienced major problems after successfully using the Java Viewer when I installed unrelated software that included a newer version of the Java Runtime Environment. In one case, it took me more than one day and extensive manual editing of the Windows Registry to get the ArcIMS Java Viewer to work again.

Appropriateness

Minimize the complexity of your applications. The most appropriate design is one that works well to perform its intended tasks within the users' capabilities and the constraints of their computing and communication environment.

Most users prefer to browse through a series of ready-made maps rather than having to manipulate multiple layers themselves. Only a relatively small number of sophisticated users will appreciate a full-featured webmap viewer. For most Web surfers, that is overkill.

Try to communicate the most important information with simple static maps. You can generate these on the fly with ArcIMS in a context sensitive fashion. Although such map images are generated dynamically, they are static in the send that you do not provide tools for the user to further manipulate them. In other words, you may design fairly sophisticated applications wile hiding much of their complexity from the user.

Such simple maps are effective at providing an overview or introduction to a topic, and may entice a user to spend more time learning to use a more richly featured mapviewer than he or she might have been willing to do in the first place.

If static maps are inadequate only because they can't be zoomed, consider using PDF files. Or, use HTML image maps to make static maps clickable, such that they will link to more detailed maps or other types of related information. An outstanding example of the potential of this technique is the Hanford DMS by Christina Drew at nalu.geog.washington.edu/dms/hanford.html.

KEY PRINCIPLES FOR MAPVIEWER DESIGN

Many webmapping applications found on the Internet today have a poorly designed user interface. They

are non-intuitive and difficult to grasp by first-time users, often cumbersome to operate, and rarely memorable enough to make tasks easier for infrequent users. Many out-of-the-box webmap viewers are unnecessarily complex. It is not rare to find an overabundance of buttons. Some may be mostly redundant with each other. Some are unnecessary or even useless for the specific application. Others may be useful only to a small fraction of users. Attempts to come up with an innovative new look for a webmapping application within the scope of a single viewer project are usually misguided.

Widely accepted user-interface standards for web maps do not yet exist, but users have become accustomed to user-interface elements that are implemented in operating-system tools and popular applications, such as the Windows Explorer, the Macintosh Finder, Netscape Navigator, Internet Explorer, or Adobe Reader (Acrobat).

Consistency and predictability

The greatest sin in user-interface design is inconsistency in the use of controls, cues, and clues. It makes an application more difficult to understand, harder to get proficient at using, and more prone to accidental mistakes by experienced users.

We can distinguish five different levels of consistency:

1. Internal consistency within an application
2. Consistency across a set of applications
3. Consistency between an application and the operating system
4. Consistency among an application and similar types of applications
5. Consistency between an application and the "real world"

A frequently observable internal inconsistency concerns the presentation of hyperlinks and static text. Take care not to use the same colors or types of underlining in static text as you use for hyperlinks. Keep in mind that subtle distinctions may be imperceptible on low-quality and poorly adjusted computer displays, and especially to users with visual impairments. The risk of violating this principle is particularly great when modifying or enhancing an "out-of-the-box" application, such as one generated with the visual ArcIMS development tools.

If you create multiple mapviewers intended for the same users or to be included in the same website, keep them as consistent as possible. Where they differ in functionalities, however, make sure to reflect those differences in the layout and appearance of controls.

Consistency with the operating system is difficult or impossible to achieve in web-based applications because in may cases you do not know or cannot predict the users' platform. It is likely that they will access your application with multiple operating systems and webbrowsers. However, you should at least avoid repurposing for your applications icons and other user-interface elements from an operating system or a webbrowser if the behavior of these elements in your application won't be identical to their original

behavior.

Out-of-the-box ArcIMS applications have become abundant on the Web. Many users now are at least somewhat familiar with their standard (out-of-the-box) look and feel. You may have great ideas for better-looking icons to use in the toolbar, but consider that at least for those users who have already learned how the standard ArcIMS tools look, you'll make your application less user-friendly and usable, at least initially. On the Web, first impressions are extremely important!

If you design a user interface that differs from the standard ArcIMS look and feel, study the user interfaces of existing webmapping applications that are likely to be of used by the same users that you intend to attract and those of the most popular webmapping applications such as the National Map, Geography Network, TopoZone, MapQuest, Maps On Us, etc. You will notice differences among them, but also many commonalities. Build on those commonalities, but avoid borrowing looks for different functionality in your own application.

Clarity

A well-organized layout and the absence of ambiguity will make your application more inviting to use and more productive. Give much thought to the labels and names you use for controls. Test mock-ups of alternate layouts and labelling with unbiased members of your intended user community.

Intuitiveness

Many of today's user interfaces are built on metaphors borrowing from real-life contexts, such as desktops, trash cans, recycling bins, file folders, etc. That use of familiar visual cues and mental concepts can help make user interfaces more intuitive and memorable. Be careful when trying to follow that common practice. There are not that many good matches between computer operations and real-life processes and objects. The best ones may already have been implemented. Pitfalls abound if your application will be used internationally. Cultural differences affect meanings and expectations that the user will associate with such visual clues. In the worst case, you might pick something that is highly objectionable in some part of the world.

Sociability, or clues and cues

A good user interface should strive to be the user's "best friend." One of the reasons the dog is considered "man's best friend" is its gregarious nature. To better get along with each other, social animals provide many clues and cues that inform group members about their state of mind and their intentions. Dogs are "user friendly" because many of their clues and cues are easily perceptible by people. Wagging tails, hair standing up on the back, exposed canine teeth, happy barks, and angry growls are just some of the more obvious aspects of the dog's user interface.

Provide visual information that helps the user understand the function of controls. Be careful not to use radio buttons and check boxes interchangeably. Use visual clues to distinguish icon buttons that invoke a different mode or state of the application and change its behavior from buttons that execute a single narrowly defined function.

For example, when providing text buttons to invoke an operation, use ellipses (...) to indicate whether the operation will be executed immediately or will first give the user an opportunity to set options or review the details.

A changing cursor is an effective cue what to do next and a clue of what to expect as the result from a mouse click.

Feedback

The user should always be able to see the status of the application and the task, what is expected, or what has gone wrong and how it can be corrected. If the application has multiple modes, make sure the current mode is indicated conspicuously. Deactivate, visually as well as functionally, controls that are unusable in the current context. Acknowledge user input and show the progress of tasks that take more than a couple of seconds. The user should never be left wondering whether the input was accepted, the last click made any change, or an operation has completed.

Feel

The "feel" component of a user interface's "look and feel" is the result of the visual and functional quality of the interface and of how well it harmonizes with and reflects the application's behavior. The user will feel comfortable with an application if she or he always feels in control. That is the case if the user interface makes it easy to get the desired results and keeps the user informed at all time what the computer is doing. It eliminates ambiguity, uncertainty, and surprises that make the user feel that it is really the application that is in control of the computer and the user.

Wolf Naegeli, PhD
Senior Research Scientist
Energy, Environment, and Resources Center
The University of Tennessee--Knoxville
E-mail: wolf@naegeli.net