The user experience

The iceberg analogy of usability

Dick Berry (reberry@us.ibm.com)  
User Experience Design  
IBM Ease of Use Team  

In his first column for developers looking for insights into better application design, Dick Berry explains why look and feel is only the tip of the iceberg. Find out why starting with the user experience leads to better application design, whether for Web users or unplugged users.

Developers sometimes ask which aspects of look and feel contribute most to the overall usability of an application or Web site. They are typically surprised when I answer that the "look and feel" aspects aren't the major contributors at all. Look and feel have been popular discussion topics for many years, and some developers have proposed various schemes purporting to allow an easy swap of one look and feel for another. They were perhaps compelled to this thinking to compensate for an inadequate understanding of their users. Around 1990, I became alarmed by the popularity of design architectures advocating paradigms like the User Interface Management Systems (UIMS) that enable a pluggable look and feel. Many of my colleagues and I felt that look and feel represented only the tip of the iceberg. We felt that the set of concepts users must learn and understand to use a product or Web site effectively is actually the most important factor. I began using the iceberg analogy in presentations, articles, and books whenever possible. Since then it has been endorsed and reused by many practitioners in the field.

While this analogy was developed through analysis and years of experience in designing graphical user interfaces (GUIs), the basic premise concerning the importance of concepts applies equally well to the design of Web sites. While some interface aspects may be weighted differently, the evolution of Web-based applications is reestablishing the applicability of this view.

Aspects of interface usability

Let's examine the interface aspects that contribute to usability, and the role of look and feel. The usability of an interface can be analyzed by identifying three sets of aspects: look, feel, and the user model, as shown in Figure 1. The look includes aspects such as visual cues, feedback, and esthetics. Visual cues convey the ability for the user to perform an action. For example, the three-dimensional look of push buttons in traditional interfaces is a cue that says "push me."
Feedback provides acknowledgement to the user that a request is being acted upon, and esthetics includes all of the aspects that create subjective appeal, such as the use of color, line styles, and typography. The feel aspects include keyboard mappings, mouse button mappings, menu structures, shortcuts, and interface navigation conventions. Taken together, the look and feel of an interface is analogous to the syntax of a language. They describe how a user can interact with a system but not what the user can ask the system to do.

Figure 1. The user model outweighs look and feel in the usability of an interface

The user model

The user model consists of aspects related to what the user is trying to accomplish, or in other words, the user's task goals. Continuing the language analogy, these goal-related aspects represent the semantics. They convey meaning in the conversation between the user and the system. The user model provides an understandable and cohesive framework of concepts that users can relate to, and that enables users to accomplish their tasks.

A user model is typically described in terms of user objects, the behaviors and properties of those objects, and their interrelationships. Those of you familiar with object-oriented programming (OOP) may think this sounds the same, and you're right, to a degree. The difference is that OOP deals primarily with implementation aspects, while user models deal only with aspects that users are expected to experience through learning and use, such as the objects, or things, users employ to do their work.

In using a system, an application, or a Web site to accomplish a task, the first hurdle any user must overcome is to map his or her goals to the available capabilities. These capabilities are best understood in terms of the "things" that are provided to enable users to work, and all of the characteristics of those things, their properties (settings), behaviors (what they can do and what
the user can do to them), and their relationships (how they work together and how the user can combine them).

A user model describes all of these aspects in a cohesive manner, and ideally in a way that users find intuitive. Metaphors are sometimes employed in designs to help users bridge from familiar models to new ones. Users develop a mental model of any system they interact with. Although users' mental models may be subconscious, the models will direct their interactions and influence their success or failure. A user's mental model is sometimes referred to as the user's conceptual model. Ideally, a user's mental model matches the user model the designer creates. I'll say more about this relationship in a future column on the role of models in interface design and model-based design methodologies.

The user model versus the look and feel

How important is the user model in comparison with the look and feel? Over the years my colleagues and I have come to believe that the aspects embodied in the look of an interface contribute about 10% and those that are driven by the feel contribute about 30%. The user model plays the major role, contributing about 60% to overall usability.

Figure 2. Look and feel make up the tip of the iceberg

Of course these are educated estimates, and they vary between traditional graphical user interfaces (GUIs) and Web interfaces. This is where Web interfaces introduce a slightly different weighting. Traditional Web interfaces have been very heavy on visuals, and relatively light on interaction and user model concepts. That's because the traditional Web site was intended primarily to convey information to the user. With the exception of an occasional form, the only interaction was clicking through links. Web pages have been designed to entice and attract, always with a major emphasis on the (sometimes-flamboyant) visuals. With the advent of e-commerce, forms have become more prevalent, and transaction flows have started to introduce more complex
user model concepts. As the Web matures and fulfills the promises of the business-to-business (B2B) paradigm by enabling applications with customers, employees, and suppliers, we will have come full-circle, returning to relative contributions in line with those described by the iceberg analogy.

Designers need to understand not only the impacts that these aspects have on user learning and productivity, but how you can affect, control, and hopefully design these aspects to meet users' needs. Let's consider the leverage points that affect these aspects.

The look and feel aspects of traditional graphical user interfaces are typically dictated by the platform on which an application is provided. The combination of a platform's UI toolkit and style guide typically prescribe most of the look and feel aspects. Similarly, for Web applications the toolkit is prescribed by the implementation platform to a greater or lesser degree, where that platform might be Java, ActiveX, or some other. For GUI applications, the user model is almost entirely dictated by the operating system and application design. For Web-based applications, the user model of the operating system has become less relevant, if it exists at all. In both environments the major source of user model concepts is the application design itself.

For example, the Windows toolkit provides entry fields, radio buttons, checkboxes, a variety of message boxes, and standard dialogs, along with a style guide documenting good usage practices. The common user model consists of data and directories, represented as files and folders on a desktop, a clipboard along with cut/copy/paste actions, and programs, represented by Shortcuts and through data type associations. Applications contribute concepts such as paragraphs, sections, tables, headers and footers, slides and slide shows, spreadsheet rows and columns, formula, and so on.

**Figure 3. User interface toolkits address only the tip of the iceberg**

What can you learn from the preceding analysis? First, you need to have users represented from the very early stages of design. You need to understand who they are in terms of their
skills and motivations, and the tasks they perform. Second, you can take advantage of user-task knowledge and software engineering methods to model the concepts as well as the look and feel. User models are the foundation of design. They serve not only as frameworks through which we can assess and validate designs, but also as a source for the crucial documentation needed to coordinate and direct an entire development team, including programmers, visual designers, writers, and testers. Ultimately, user models can help you greatly improve your odds of providing designs that hit the mark the first time, and that users find intuitive or even compelling.

The human-computer interface for Web applications

Designers learned a great deal through 15 years of evolving the graphical user interface. It certainly isn't perfect, but its evolution has all but stopped. In many cases Web designers are repeating past sins when they blindly adopt GUI approaches for Web-based applications. The Web offers new opportunities for creative solutions that match users' needs, wants, and abilities. Whether we as designers can take advantage of this opportunity to improve the human-computer interface is still a question. We think we understand many of the relevant factors that enable user-driven design - the iceberg analogy conveys a few.

In a future column, I'll explore further the fundamental differences between GUI and Web-based design, and the impacts on ease-of-use. In coming months I will introduce a methodology for explicitly designing user models to ensure that users' needs and expectations are met in the very early phases of product design and development.
Resources

- Visit the [IBM ease-of-use](https://ibm.com/developerWorks/) site for the latest in design guidelines, from designing for the Web to out-of-box-experience. Also, find out how you can participate in IBM's annual worldwide conference on ease-of-use: "make it easy 2001."
- Explore cross-industry initiatives to improve the ease of use of PCs in general. The PC Ease of Use Roundtable consists of leading PC, communications, and peripheral manufacturers who meet monthly to share data and collaborate on ways to improve the user's experience.
- David Travis at System Concepts, Ltd. has published an article proposing similar degrees of contribution to usability based on the roles of visual design, consistency, and user's tasks.
About the author

Dick Berry

Dick Berry is a Distinguished Engineer with the IBM Ease of Use Architecture and Design team in Austin, Texas. His focus since 1980 has been the design of human-computer interfaces, user object models, and methodologies for ease of use design and development. He was the lead architect for several generations of IBM's Common User Access (CUA), a user interface style initially published in 1987. Dick has also been an innovator in designs for visual programming, and the use of virtual-reality techniques to create a productive 3D user environment for office tasks (IBM RealPlaces). Dick has 45 US patents in the field of user interface design. He is an elected member of the IBM Academy of Technology and an author of IBM's Ease of Use Web site. You can contact him at reberry@us.ibm.com.

© Copyright IBM Corporation 2000
Trademarks
(www.ibm.com/developerworks/ibm/trademarks/)