How to Choose Good Metaphors

Ever wonder why some icons don’t seem remotely related to their functions, while others are wonderfully intuitive? The reason may very well be the lack of user involvement in choosing metaphors and defining their behaviors. John Laugren, a consultant with 18 years experience in designing graphical user interfaces, passes on his company’s techniques for ensuring that good metaphors are included as part of interface design.

— Bill Curtis, Editor

Making Metaphors An Explicit Part of Interface Design Can Help Further the Client’s Understanding of Interface Design in General. When clients realize that the roots of the design process lie in a metaphor, they are more able to grasp what designers do and why. Metaphors create a common understanding, a way of discussing interface design that is understandable to anyone, regardless of their technical background and experience.

But although they seem straightforward on the screen (well, sometimes), metaphors are not always easy to implement. They require an understanding of what a metaphor is and what it is supposed to accomplish, what behaviors you can add to make it even more powerful, and what form it should take on the screen.

What Is a Metaphor? Most English classes teach that a metaphor is another way to think of something. In the interface world, it gets more complicated than that. I remember beginning a project about 10 years ago at Microelectronics and Computer Technology Corp. to investigate the kinds of interface metaphors that would be appropriate in 10 years, given the rate of computer technology advancement. I and the rest of the project team assumed that people would have very fast, desktop PCs and that these machines would have high-resolution color displays and could display dynamic 3D graphics. But we didn’t get much further than that. In a very short while, we were asking ourselves, “What is an interface metaphor, anyway?”

Fortunately, a few years earlier George Lakoff and Mark Johnson had written Metaphors We Live By (University of Chicago Press, 1980), which helped clear up some of our confusion. As they explain it, we understand and communicate about our world through metaphors. For example, when someone says, “I’m really feeling down today,” they are actually using a spatial metaphor to convey bad or sad feelings. The book is full of such examples, and I recommend it for those who are struggling with the concept.

Metaphor As Map. After many discussions about metaphor and interface, we began to realize that all computer interfaces are metaphorical. All the bits on the screen represent something else. There are no real files or folders, for example, no clock with numbers, and no document. These are all associations created by the designer. The words chosen tap into our metaphorical memory, causing us to immediately associate a file with the contents of a file cabinet and documents with books. Because long-term memory is thought to be largely associative and visual, having visual metaphors is a powerful way to support the work we do with computers.

This understanding led us to the working definition of metaphor as “a partial map between two concepts.” When we call the Macintosh a desktop interface, for example, we don’t mean it has all the characteristics of a desktop or even a subset of them. It is similar, but not identical. The difference is that because the computer is the ultimate elastic medium, we can actually extend desktop behavior to suit our purposes. We can put anything on these screens, from random patterns of bits to buttons and menus. Users certainly don’t expect real folders to be able
to have other folders inside them, which have other folders inside them, and so on, but they are willing to accept that behavior for the metaphoric folder in the desktop interface.

These extensions are a major source of power. The extensions that we add, such as having folders within folders, being able to search for a folder by name, and displaying what is in a folder in many different ways, provide users with capabilities they just don’t have outside their computers.

**Finding Good Metaphors.** How do you find good metaphors? To be usable a metaphor must be in the user’s sphere of knowledge. The lack of understanding of this requirement is what frequently causes problems in interface design. Many of the interface metaphors I have seen are metaphors within the knowledge of the developer, but not within the knowledge of the user. For example, my father, a retired physician, bought his first computer about five years ago. Disk drives just weren’t within his sphere of knowledge, so while the folder and document metaphors work fine, he has never fully understood that little disk icon in the upper corner of the Macintosh screen.

What this says is that you have to talk to the users, not just to other developers. In my consultancy, we use a technique called contextual interviews to gather this information. I’ll explain it by walking you through a session with a hypothetical potential user I’ll call Sally.

The first thing we do is phone Sally and schedule a two-hour meeting at her place of employment. We always try to meet the person where he or she works because the results are likely to be more accurate. When we interview users outside their work environment, they give us an overly organized picture of their work, apparently forgetting that real work tends to be less structured and is often interrupted. Also, we have found that two hours is just about the right time. Longer sessions tend to distort results because the session is exhausting for both interviewer and the interviewee.

Before beginning the interview, we ask Sally if she has any comments, questions, or issues of a general nature. Our intent is to put her at her ease and dispel any excitement or nervousness she may have. Once we forgot to do this and the subsequent interview was going so badly that the interviewer felt compelled to ask what was wrong. It turned out that the interviewee was color-blind and concerned that he couldn’t help us at all because he knew the interface was going to be in color. We were able to allay his fears by telling him that the interface would use color only as redundant coding and that his needs were important input. The rest of the interview went smoothly.

As part of the preinterview, we inform Sally that the interview will be audiotaped. We tried videotaping, but the interviewees seemed too self-conscious about how they would look and were worried about who would see the tape.

We then ask Sally what she would be doing if we weren’t here and ask her to start that task. While she is doing it, we ask her to describe what she is doing, why, and for whom. We also ask what kinds of difficulties she might run into that we don’t see.

When she describes any of this, we are careful not to direct her word choices. If anything is unclear, we ask for clarification by asking her how she might describe it to a new coworker. We also note what items she has in her work space and ask her to describe what they are for. As we usually do, we make sketches of the work space as she talks. At the end of the session, we briefly discuss with Sally what we saw and ask her if there is anything else she would like to tell us.

As follow-up, we prepare a complete report of the interview, including an annotated transcript, sketches, and copies of any forms, calendars, and other supplemental material. In annotating the transcript, I mark the nouns Sally used, the actions she mentioned, and any relationships among the nouns.

After repeating this process for several other potential users, we generally
get a sense of the metaphors that might be appropriate. For example, if several people in addition to Sally refer to a "red book," and the system is to manage information in this book, then the red book becomes a likely metaphor. It works because whenever anybody in the organization mentions this term, everyone knows what they are talking about, and what is in the book.

CHOOSING AND EXTENDING BEHAVIORS. The second step in this process is to determine which behaviors in the object chosen to be the metaphor are worth preserving. In investigating the red book, for example, we find that everyone who uses the book knows how it is organized. This information tells us that there is one basic organizational view of the book. If the actual red book has a table of contents, we should probably include one in the metaphoric red book. People generally open a book, so that is probably the action the user will invoke to see what is inside the metaphoric book. Other operations might be to add and delete forms or change information.

During this step, we bring in users and talk to them about the metaphors we are planning to use and what functions they might have.

Having selected the object for the metaphor and the behaviors to include, we now begin to look at what additional functions we can give the metaphor to make the interface even more powerful. The functions we add must be productive and easily understood. At this point it is critical to have discussions with potential users to determine what functionality is appropriate.

This is the step that trips up many interface designers. They frequently adopt the "kitchen sink" approach, which leads to a system with incredible amounts of inaccessible power. I can't think of many things that frustrate users more than playing hide and seek with system functions. When we talk to users about functionality, we ask them at least three questions:

- Where would you use this function?
- How much would you use it?
- Is the function necessary for you to effectively get your work done?

The answers usually give us some idea of how to rate the function being suggested.

GETTING VISUAL. About this time, we start working with a graphic artist. We are fortunate to have an extraordinarily talented artist and designer who sketches the metaphor on paper. We begin this way for two reasons: it keeps down the cost of exploration and it helps users understand that the only expense of changing appearance or look is the time it takes to resketch. Our artist can easily rough-sketch a new screen in less than a minute. We also retain multiple versions of screens and metaphors so that users can see several ideas that have already come up and so that we aren't re-creating ideas. It doesn't take very long before consensus starts to emerge on the metaphor, its functions, and how it should look.

There are, of course, alternatives to using a graphic artist; you can sketch the metaphors yourself, for example, but I don't recommend it. The artist we use is exceptional because she has all three of the qualities you need: She is a trained graphic artist. She can work on a computer. She has used interfaces and understands what they should do. Those who have only the first quality tend to get too arty. Those with only the second quality tend to get carried away with the details of producing the image. Both of those tendencies can cost you a lot of time.

FAILURES AND SUCCESSES. Although failure and success are subjective terms, I can offer some advice on what doesn't work and what does.

The biggest source of failed metaphors is developers who impose their metaphors on the user. Many developers we have worked with believe that if they like the metaphor, the user will too. We worked with a client recently who wanted to have detailed descriptions of hardware communications appear on the screen all the time. That way, he reasoned, the user would be able to tell an engineer what was going on if an error occurred. Because the client was an engineer, he was naturally thinking like one, and to him, the information would be valuable. The user, on the other hand, found that much detail distracting.

Another source of failure is developers who try to improvise on an accepted metaphor. One client we worked with decided to use the cut-and-paste metaphor but to change it to delete-and-paste. The reasoning went something like this: We use the same key ("c") to do cut and copy, so wouldn't it be better to change cut to delete and eliminate the potential confusion of cut and copy?

This reasoning ignores what I call metaphor coherency. Cut and paste work together. To change cut to delete is to violate the notion that when you cut something, you still have it available to paste. Predictably, users were continually confused about what delete meant.

It is somewhat harder to pinpoint successes, perhaps because users tend to know more about what doesn't work. The success of a metaphor is likely to translate into no complaints or general statements rather than specific praise. Approaches that contribute to success are to actively involve the user in the choice of metaphor and to consider overall organizational metaphors, such as the desktop, spreadsheet, and paint canvas. Each of these metaphors immediately creates some expectations about what kinds of things a user might be able to do and provides an anchor for activity.

The real success, however, is the more positive attitude of the user. When people tell you they can now do something more easily or faster than they did before, that's a clue that the computer has disappeared and they are better able to focus on their work.

---

John Luscyn's address is Luscyn & Luscyn Assoc., 11615 Joger Rd., Suite 104, Austin, TX 78759

MAY 1994