

ICONS: Much Ado About Something

— published in *The X Journal*, January/February, 1996



Eric Schaffer, Ph.D., CUA, CPE, is Founder and CEO of Human Factors International, Inc. (HFI). He teaches, consults, and speaks on corporate and governmental Web and GUI interface design issues.



John Sorflaten, Ph.D., CUA, CPE, teaches and consults as a Project Director at HFI. With Eric, he initiated a usability curriculum at a local university in his home town of Fairfield, IA.

©1996, Human Factors International, Inc.

Every battle has a psy-ops component, otherwise known as psychological operations. Each side attempts to demoralize the other and remoralize it's own troops. In UI design, the battle against GUIs from hell is no different. Recall the evil influence of cryptodesign—design ideas that work for certain situations but get misapplied in other, quite different circumstances. We've seen a lot of developer trauma associated with icon design: cryptohyperinconitis. But hang on. This article gives you, the troops in the field, some psycho-inoculation against the cryptic IMFAP syndrome (Icon Mania, Fetish, and Phobia)!

ICONS WERE MEANT TO BE HANDLED According to the dictionary, an icon is either:

1. an image; esp. a religious image painted on a wood panel, or
2. a small picture on a computer display that suggests the purpose of an available function.

Think about a folder icon. You can drag a document icon to it and the document pops into the folder. You can drag the folder and pop it into another folder. Ergo: icons were meant to be handled. The folder and the document icons suggest the purpose of their “available function”: the folder stores the document. Stretch the folder metaphor, and a folder can also store a folder. However, stretch the metaphor even more, you can click on a folder and it turns into a window! Oops! (or OOPS for our object-oriented friends). Thus, there are many uses of icons, although some may require considerable suspension of disbelief. But that's okay if it works. In summary, icons provide the benefit of being “directly manipulated.” They can be dragged, they can be dropped—all activities that are natural to our kinesthetic, nonverbal intellect. This augments the analytic, verbal intellect. We can do both kinds of thinking at once. It's called “dual task processing.” What better satisfaction than dragging a document to a shredder (or trash can) while talking to someone on the phone? We can do neat things in a few motions that would otherwise require prohibitive amounts of attention-diverting verbal commands. “Soul design” brings this satisfactory outcome from that small iconic picture on your computer display.

CRYPTO ICONS SNEAK ATTACK In our GUI design seminar, we survey participants for their ideas on what icons do for users. Here's one list:

1. Save time (users doesn't need to “read” a label)
2. Save space (a picture is worth a thousand words)

3. Cross cultural (pictures transcend language)
4. Usable by illiterates (again, the picture thing)
5. “Easy,” and
6. “Flashy.”

Here’s a test: Enter the function of each of the icons in Figure 1.

A hint: They are used for text formatting and printing functions.

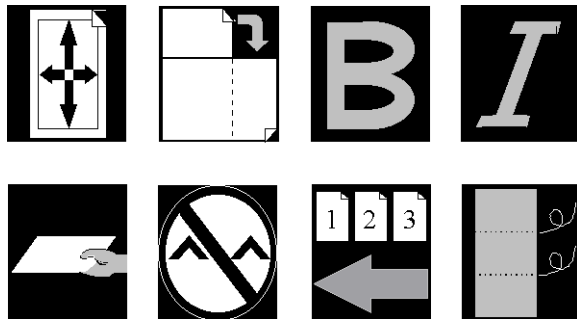


Figure 1. Write down the function of each of these icons

What did you write? (Check further into the column). Typically, only two out of the eight get more than about 15% correct answers (Bold and Italics). What went wrong? Six of the eight icons were not easy. They were not cross-cultural. They did not solve the problem of illiteracy. And they didn’t save any time. Yet, these icons could appear in any average application.

We have just diagnosed a case of cryptic IMFAP syndrome (Icon Mania, Fetish, and Phobia)! We fervently believed icons were groovy. We focused on the image. And now, we might be afraid of using them any more. How beat the ensuing trauma of crypto-hypericonitis?

THE TASK IS THE THING As in each of our columns, we resurrect soul-design by analysis of the various kinds of work involved: Visual, Intellectual, Memory, and Motor (the VIMM model). Our problem arises when we forget that icons are best used for direct manipulation. Thus, the icons in our test served only marginal benefit as static “images” such as found painted on a “wood

panel”—or tool bar—or push button, as the case may be. Yet, we do not deny the beauty of religious icons. Similarly, in the context of elevating our UI out of the drab, dreary world of black and gray text, the “flash” value of subtly colored icons can provide some entertainment. Let’s see how we can make icons really work. Let’s get our IMFAP shots.

Reduce Visual Work

Issue: Users may fail to understand an icon if it doesn’t “fit” with accompanying icons. Consider these command sets associated with moving your right arm: Raise, Down; Left, Open; and In, Forward. Now compare with this command set: Up, Down; Left, Right; and In, Out.

Solution: Design icons so they share similar visual attributes. For example, make them all either 2D or 3D, not some of one and some another dimension. Draw from similar metaphors. Paint or draw programs make it clear that an aerosol can is not hair spray, by associating it with a bucket of pouring paint, a paint brush, and a pencil. We understand that the aerosol can creates a texture of a can a spray paint. This phenomenon can be called visual “coherence”. The icons should stylistically appear as a “family” of icons that go together.

Reduce Intellectual Work

Issue: While visual recognition of images is easy, abstract, action-oriented icons carry unexpected baggage. Try to “read” figure 2.



Figure 2. Icons that you can easily read.

It’s a “rhebus”, a combination of text and pictures that we use with our kids (or was given to us by our parents). It’s a book with icons as training wheels. Can you read it? I suspect you can figure it out as quickly as our course participants. But why was that so easy and the icons in Figure 1 so difficult? Notice that the rhebus pictures are all objects. They are nouns. However, most of the Figure 1 icons are actions. They are “verb forms.” Reading clockwise from the top left, we have “make the text fit in the margins,” “rotate to landscape,” “make bold,” “make italics,” “remove all page breaks,” “print in reverse order,” “remove all control characters”, and “hand feed the paper.” Actions are difficult to interpret graphically! Note that we recognize bold and italics only because we have memorized them based on our English language spelling.

Solution: Attach actions to objects. Make the action concrete by associating it with the object to which it applies. Many actions benefit from using arrows. See Figure 3.

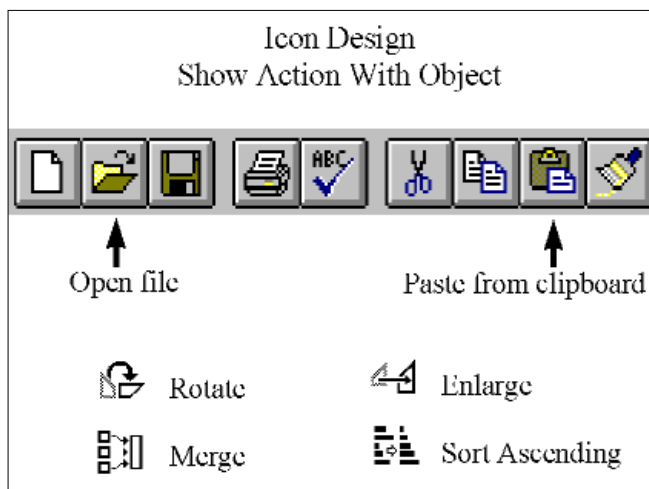


Figure 3: Incorporate an object to make an action icon clear.

Other interpretation problems remain. Do not expect similar interpretations of icons across cultures. The upheld hand may imply “stop” in North

America. However, the same gesture in Mediterranean cultures communicates an unprintable remark. Similarly, the trash can or mail box images common to us are not meaningful in Asia or Europe. Icons must be tested by sample users. The authority of respected authors may be useful as well. Apple’s interface design handbook provides an excellent start on issues of localization of international applications.

Reduce Memory Work

Issue: We didn’t do well on the icon test. We saw, but we didn’t understand. In fact, can you honestly affirm that you recall the meaning of all the icons on your word processor toolbar ribbon? Only about 5% of our course participants can. Cryptodesign says use pictures because we think we can remember them.

Solution: Soul design says use buttons with text, or add labels to icons you design. Accept that most icons are not instantly recognizable. Learning icon definitions costs your firm money since users take paid time to get trained or train themselves. Since most icons are used infrequently, it is best to avoid constant “retraining” by attaching labels to the icons. By the way, adding labels doesn’t cost any time in interpreting labels. Scanning a label or textbutton adds only 30 to 300 milliseconds. That’s one-third of a second, at most.

Remember that tooltips are only marginal help. First, tooltips take about two-thirds of a second to appear and be registered in our attention. In a row of 12 icons, that’s about 8 seconds of additional work. Second, research shows that users hate to “dig” for information. In one study, users relinquished the benefits of bringing up a dialog box menu, and made do with what they already knew. They happily worked slower and less efficiently. Historically, human factors specialists invented toolbar ribbons as a solution to the problem of

remembering hidden pulldown options. Similarly, labels are the solution to remembering hidden tooltips.

But let’s not be dogmatic! Omit labels on icons if you really need to save space on the screen for drawing or word processing, etc. and are willing to pay the penalty of training and retraining. In these events, tooltips are great!

Reduce Motor Work

Issue: Icons require reaching for the mouse to either click the icon or drag it. This is extra work when the user is trying to keep typing. Furthermore, icons without labels cannot display an accelerator key (e.g., Alt + key).

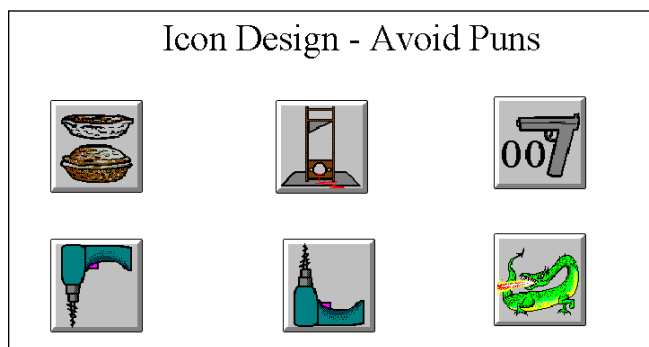


Figure 4: Humor does not “travel” well. (L to R: “Summary” (in a nutshell); “Execute” (run a program); “Access Bonds” (financial program); Drill Down; Drill Up; “Engage drag-and-drop function” (drag-on).

Solution: Any corporate, data-entry application should be designed to help users keep their fingers on the keyboard. Therefore, consider using regular text-labeled buttons instead of icons. While accelerator keys can be used on text buttons or on icon labels, the simultaneous use of two or more keys

results in high rates of typing errors. Instead, for application-wide functions use a function key assignment, like “Take Order (F3)” on a text button.

Another symptom of losing the psy-ops war to cryptopsychosis involves “useless” drag and drop. For example, one fax application had two scrollable list boxes side-by-side. The left scrolled list contained names of people in an “address book.” The right scrolled list was intended to hold any selected names for the current fax event. Much trial and error ensued. Clicking a name in the address book failed to send it to the right list! Only after accidentally moving the mouse while clicking, did the user discover that he had to drag the name to the list on the right. Since there was only one possible destination, it is clear that drag and drop was crypto-overkill. A simple click would do. However, if the application required the user to choose one out of multiple destinations, then drag and drop could be suitable.

WRAPUP AND CONCLUSIONS Psychological warfare is as brutal as physical warfare. Icons take on roles of both the beauty and the beast, to use a metaphor. The beauty of icons can spruce up a dull interface. However, we need to know the pitfalls of icons to defend the beauty against untoward advances by the beast. The VIMM model can guide our religious use of icons. And in our most creative moments, we may even find opportunities for the highest use of icons—as drag-and-drop metaphors that permit direct, kinesthetic interaction with clearly defined objects. Icons were meant to be handled.