

Deliberate Design and the Web

Michael Wojcik

30 April 2008

The Myth of the Intuitive

At least in the realm of general computing, there is no such thing as a truly intuitive user interface. The term “intuitive”, when describing user interfaces, is usually an exaggeration, marketing hyperbole used primarily by the proponents of WIMP (Window, Icon, Menu, Pointer) interfaces. The association of “intuitive” with a software user interface was probably introduced by Alan Kay, who derived it from Jerome Bruner’s educational psychology; but for Kay, interfaces were not “intuitive” in the sense of “instinctively understood”. Rather, they could appeal to Bruner’s three stages of learning—enactive, iconic, and symbolic—so that as users learned the interface enactively (“if I push *here*, then *that* happens”) and iconically (“*this* symbol means *that* result”), they would develop a foundation for understanding the interface symbolically (they would have an accurate conceptual model of how the software worked). As users enter the third stage, their knowledge from the first two becomes, for Kay, “intuitive”—in the sense that they have internalized it and do not have to consciously recall it.¹ But intuition was not something new users would simply possess *a priori*. Similarly, while Apple’s marketing team liked to claim that the original Macintosh was instantly comprehensible to the most naive of users,² Jef Raskin, the project leader for the Macintosh project, famously stated that “intuitive” merely means “familiar”.³ Raskin’s point is that, as the term is properly used in HCI (Human-Computer Interaction studies), “intuitive” does not mean “immediate”. Interfaces can only be intuitive once a user has learned them.

User interfaces are not intuitive, or not entirely intuitive, for at least three reasons. Since no two applications do precisely the same things in precisely the same ways, even the most experienced users coming to a new application for the first time will find discrepancies between their mental models and the application’s actual behavior. Second, an application cannot appeal directly and solely to instinctual or even learned human behavior for dealing with the physical world, because software exists to mediate between data and a human audience. An application that presents nothing but unmediated visual information isn’t software—it’s closed-circuit television. (And even that requires interpretation, since

¹John W. Maxwell, *Tracing the DynaBook: A Study of Technocultural Transformations*, diss., U British Columbia, 2006, 128–129, 26 April 2008 <<http://thinkubator.ccsf.sfu.ca/Dynabook/Maxwell-DynabookFinal.pdf>>.

²See for example the 1984 multipage *Newsweek* advertisement. Apple Computer Inc, “Macintosh”, *Newsweek* November-December 1984: 0-2+, 26 April 2008 <<http://www.guidebookgallery.org/ads/magazines/macos/macos10-newsweek>>. This advertisement only uses “intuitive” once, and not in relation to the user interface, but <<http://myoldmac.net/FAQ/Apple-Introduces-Macintosh1984.php>>, which claims to reproduce the original Apple press release for the Mac, does describe cut and paste as “intuitive operations”, for example.

³Jef Raskin, “Intuitive equals familiar”, *Communications of the ACM* 37.9 (1994): 17, 27 April 2008 <<http://www.asktog.com/papers/raskinintuit.html>>.

it limits the point of view, and so on.) Those are two reasons why application user interfaces generally cannot be intuitive.

The third and most important reason why user interfaces are not intuitive is that they *should not be*, and developers recognize this in specific instances as they create applications, even if they don't deduce the general principle. Most user interfaces should not appeal directly to the senses and instinctive behavior, because that's a poor information channel. While haptic, neural, and other close-to-the-meat technologies continue to improve and offer useful interface affordances for specialized tasks, it's likely that most applications will always be better served by text and other semiotic representations. We use those for a reason: they have relatively high information density, particularly for conveying abstract concepts and generalizations, and powerful rhetorical devices.

Semiotic Interfaces

Let's look at the specific case of web pages and web applications.⁴ Web pages and web applications are semiotic: they present information that must be understood as symbolic and decoded. A link (an HTML anchor) is not a road a user can look down to see where it goes; the destination has to be indicated with some sort of sign.⁵ As semiotic experiences, web pages must be interpreted by the user; and since they must be interpreted, there's an opportunity and a need to guide interpretation.

"Semiotic" simply means content that employs signs—representational tokens which are associated by convention with arbitrary meanings.⁶ The fluidity and multiplicity of those associations creates ambiguity and space for multiple interpretations, but to be effective a user's interpretation of a user interface must have reasonable predictive power. Precisely what that means is a contested question in the philosophy of language: intentionalists like Walter Benn Michaels believe that there is a "correct" interpretation (and it's the one intended by the creator),⁷ while Richard Rorty, for example, believes that language is not a "medium" that conveys ideas but a set of behaviors that interlocutors use to coax one another's mental models into producing similar results.⁸ Under any of these models, though, all semioproductive activity—speech, writing, art . . . and creating web pages—does two things simultaneously: it impels and constrains meaning-production in the audience. It must impel meaning-production so that the user understands some meaningful consequence of interacting with the page; users have to feel the urge to integrate the information on the screen with their mental models and understand the page as satisfying some lack in those models.⁹ And, conversely, it must constrain meaning-production if it hopes to bring the

⁴I define the latter as those web pages with additional user-interface controls, beyond the basic ones provided by the browser (more properly the "user agent"), such as forms; and pages that let users make requests with side effects (what HTTP calls "non-idempotent requests"—see IETF RFC 2616 <[ftp://ftp.rfc-editor.org/in-notes/rfc2616.pdf](http://ftp.rfc-editor.org/in-notes/rfc2616.pdf)>). Basically, they're web pages that let the reader do more than read and navigate.

⁵It's a mistake to think that icons are any less signs than words are. Icons are all conventions, despite having a visual resemblance to some physical object the user may be familiar with. There is no necessary association between, say, a crude image of a pair of scissors and the process "take some data which has been previously demarcated in some way and move it from its current place into a temporary location".

⁶A meaning is arbitrary if it is not derived immediately from an inherent attribute of the signifier.

⁷See for example Walter Benn Michaels, "The Shape of the Signifier", *Critical Inquiry* 27 (2001): 266–83.

⁸This is Rorty's interpretation of Donald Davidson's work, though Davidson has some reservations about Rorty's formulation. See Richard Rorty, *Contingency, Irony and Solidarity* (Cambridge: Cambridge UP, 1989) 9–16 *et seq.*

⁹I elaborated on a version of this idea, a theory of "narrative energy" conveyed from author to audience, in an unpublished essay "Technologies of Transmission: Narrative Energy and Story Hybrids" (1993), which unfortunately is only currently available in obdurate offline formats. Essentially my argument there was that narratives always resist complete closure and

user's mental model into some agreement with the actual behavior of the system. So the user interpreting the arbitrary signs offered by the interface must be influenced into creating a model that more-or-less accurately predicts what the interface does—or the interface is a failure. (Note that the user's model need not actually correspond to the inner workings of the interface, provided they have for the most part the same observable effects. Often, though, interfaces break down by misleading users into constructing models with the wrong internal behavior. Donald Norman describes a number of such cases.¹⁰)

So nearly all user interfaces are not completely intuitive, but must rely to a significant (often overwhelming) extent on semiotic interaction. Semiotic systems are powerful: they prompt the audience to produce extensive, complex meaning. But that also means that they have a significant risk of leading a user to produce a model that does not predict the system. Also, because they make demands on the audience (to interpret and produce meaning), they can fatigue and discourage users. Clearly this presents a challenge for creators of web pages and web applications.

Inevitable Design

Put in more concrete terms: few pages contain undifferentiated text with no controls. Nearly all have some markup, for text flow (paragraph divisions, for example) and semantic hinting (emphasis, headers, lists, etc). Most have links. Most have mixed content, with images, forms, and other elements. All of these are user-interface elements; even paragraph breaks (attempt to) delimit sequences of related ideas, and the simplest sorts of typographical emphasis indicate importance, stress, or emotion—interpretive prompts to the user. They all affect how the user interacts with the application, even if that interaction is limited to reading. And as Jesse James Garrett neatly defines it, this is precisely the domain of user experience: “how [a product] works on the outside, where a person comes into contact with it”.¹¹ In the act of assembling a web page, of creating and collaging content, structuring and styling it, and linking to and from it, the web page author or web application developer is inevitably engaged in user experience design with every choice. The question is whether this design is accidental or deliberate.

Accidental design is the result of a developer serving as the sole target audience for a page. The obvious result, and the one that critics often point to, is poor aesthetic choices.¹² But accidental design and an impoverished conception of target audience also lead to less-obvious but often more troubling problems. The nonconforming HTML with nonstandard markup used on a great many sites, including important commercial ones, during the era of the browser wars effectively blocked users of minority browsers from much of the web and paradoxically stifled innovation by giving Microsoft a near-monopoly; this situation is only slowly being alleviated by improving market share for Firefox and other user agents and increasing pressure from web developers for standards compliance.¹³ The use of frame- and table-based

ask the audience to continue them.

¹⁰See for example the refrigerator anecdote in Donald Norman, *The Design of Everyday Things* (New York: Doubleday, 1988): 14–16.

¹¹Jesse James Garrett, *The Elements of User Experience: User-Centered Design for the Web* (New York: AIGA–New Riders, 2003): 10.

¹²See for example Ze Frank, “Ugly MySpace Contest”, *The Show with Zefrank*, 14 July 2006, 29 April 2008 <<http://www.zefrank.com/theshow/archives/2006/07/071406.html>>

¹³The deleterious effects of Microsoft's Internet Explorer on web standards and innovation are largely a matter of consensus in the industry. See for example Fred J. Aun, “Microsoft's IE ‘Monopoly’ Has Opera Singing the Blues”, *Linux Insider*, 13 December 2007, 30 April 2008 <<http://www.linuxinsider.com/story/60761.html>> on Opera Software's complaint to the EC over this issue. As for stifling innovation, note the five-year hiatus between the release of IE 6 (August

layouts produces pages that work poorly with non-GUI user agents (text-mode browsers like Lynx, screen readers used by the visually impaired, etc) and user agents with limited screen real estate (mobile devices, older computers often found in schools and public libraries, computers with display resolution set low for the visually impaired or elderly, and so on).¹⁴ Many of these problems are the result of two assumptions common to accidental design: the page author is a model for all users, and the author's browser is a model for all browsers.

Of course, even deliberate design can lead to many of the same problems, when it pushes the limits of the technology or abuses features. Jeffrey Zeldman illustrates this with a discussion of Suck.com, ironically one of the most-praised sites of the mid-1990s.¹⁵ After describing some of Suck's design innovations, Zeldman explains how these were achieved by abusing the limited presentation capabilities of HTML—in particular, by inserting extraneous paragraph separators to achieve double-spacing and a skeletal text line. This led to at least three problems: the site's appearance depended on rendering quirks of the Netscape browser, the site was difficult to maintain and built by a cobbled-together toolchain of scripts, and most importantly it was painful to use with a screen reader. Zeldman summarizes: “most commercial production still relies on bizarrely labor-intensive workarounds and hacks and continues to suffer from the problems these methods engender” (47).

Because of the inertia of software maintenance—the reluctance of maintainers to change anything that works (however poorly), the temptation to copy and paste, and the tendency to employ “cargo-cult programming”¹⁶ when solving problems that the author doesn't really understand—the only solution to this problem is a vigorous, robust commitment to deliberate design that includes a thorough understanding of web standards and best practices, and their advantages. This is the inspiration for movements like POSH (Plain Old Semantic HTML), whose advocates recommend using HTML markup in as semantically meaningful a manner as possible, and deferring all presentation to style sheets.¹⁷ This separation of content and presentation, and association of content with semantic metadata, gives the widest range of user agents the best chance at providing a pleasing, accessible, consistent experience for their users.¹⁸

2001) and IE 7 (October 2006), with no significant feature updates in between, for a browser that commanded as much as 85% of the market. For historical browser market share, see “Browser Statistics”, *W3schools*, 30 April 2008 <http://www.w3schools.com/browsers/browsers_stats.asp>.

¹⁴A related and still very common issue is overly-aggressive font sizing, one of the great controversies in CSS practice. See for example Richard Butter, “How to size text using ems”, *clagnut*, 18 May 2004, 30 April 2008 <<http://clagnut.com/blog/348/>>, and opposing views (which have my sympathy) such as Beauregard T. Shagnasty, “Font Sizes”, *Tekrider.net*, 30 April 2008 <<http://tekrider.net/html/fontsize.php>> and Bergamot, “Sensible type sizing on the web”, 18 April 2008, 30 April 2008 <<http://bergamotus.ws/misc/sensible-css-text-sizing.html>>.

¹⁵The praise lavished on Suck.com by Josh Quittner, executive producer of Suck's competitor *The Netly News*, is typical. See Quittner, “Web Dreams”, *Wired* 4.11 (November 1996), 30 April 2008 <http://www.wired.com/wired/archive/4.11/web_dreams_pr.html>. Zeldman's discussion is in Jeffrey Zeldman, *Designing with Web Standards*, 2nd ed. (New York: AIGA–New Riders, 2007): 43–47.

¹⁶The phrase “cargo-cult programming”, coined by analogy with Richard Feynman's “cargo-cult science”, refers to deploying code without understanding it, on the grounds that something like it seems to have worked elsewhere.

¹⁷“Plain Old Semantic HTML (POSH)”, *Microformats*, 30 April 2008 <<http://microformats.org/wiki/posh>>.

¹⁸Even the back end of web applications affects user experience; for example, a reliable application means a consistent user experience (while an unreliable one is frustrating), a good data model means clean data can be presented to the user, and so forth.

Design, Rhetoric, and Writing

The two general problems I identified of accidental and haphazard design, misinterpretation (by user agent or user) and fatigue, are traditional concerns of rhetoric. (Consider for example the Enlightenment rhetors: Locke and his followers strove to create a language free of misinterpretation, while Spencer and others investigated how rhetors might keep an audience engaged.¹⁹) Even in common parlance, the term “rhetoric” is associated with techniques of speech and literary devices that appeal to the audience; and despite the popular association of rhetoric with propaganda and deception, academic rhetoric has always been interested in producing “right thinking” in the audience. It should be no surprise, then, that it is easy to draw parallels between basic principles of user experience design and rhetoric.

Take the CRAP design principles introduced by Robin Williams.²⁰ *Contrast* is related to the rhetorical canon of invention: a web page must offer something new, something surprising, that contrasts with the user’s existing experience. (One formulation of information theory defines the information content of a message as the amount of surprise it produces in the recipient.) Many tropes—metaphor, for example, or irony—are functions of contrast, and the term “trope” itself means “turn”, essentially a synonym for contrast. The canon of delivery, too, incorporates contrast, as rhetors would learn to use changes in pitch, volume, and so forth to stir the audience and signal changes in tone and meaning as they spoke. *Repetition* is also part of delivery, and essential to the canon of memory; and repetition itself is a classical trope. Designers usually think of Williams’ *alignment* and *proximity* in terms of spatial arrangement, which does not apply directly to the linear speech and text of classical rhetoric (though it does to contemporary visual rhetorical theory). But because speech and text are temporal, alignment and proximity are prominently featured in the ordering of elements in classical rhetoric; the canon of arrangement is clearly the natural rhetorical companion to the latter two CRAP principles, and allied concerns in traditional rhetoric are the canon of style (parallel linguistic constructions, part of style, are a textual form of alignment, for example) and tropes such as metonymy.

Deliberate designers operate in a field of vast possibility but also numerous, often conflicting, demands and constraints. They can present user interfaces with rich semiotic information channels, including dynamic and interactive ones; but this also means that they must take steps to guide how users produce meaning from that information. They have a wealth of technological tools for semantic metadata and presentational effects, but they must take into account how heterogeneous user agents and users will use and experience the results. Zeldman points out that the standards bodies and software developers who provide these tools have a natural “emphasis on science over style or consumer-friendly ease of use”—which is exactly the opposite of most users (90). And, as Garrett notes, sites must accommodate the disparate needs of different classes of users, such as novices versus experts (50). Because rhetoric is at its core the study of the effective use of semiotic systems, with all their complexities, ambiguities, strengths, and frailties, rhetors are trained in making these kinds of choices.²¹ Even in the development process, designers have to negotiate among conflicting demands from various stakeholders (Garrett 81–82), which

¹⁹See eg John Locke, from *An Essay Concerning Human Understanding, The Rhetorical Tradition: Readings from Classical Times to the Present*, ed. Patricia Bizzell and Bruce Herzberg, 2nd ed. (Boston: Bedford-St. Martin’s, 2001) 817–827; and Herbert Spencer, from *The Philosophy of Style*, Bizzell and Herzberg 1171–1180.

²⁰Robin Williams, *The Non-Designer’s Design Book: Design and Typographic Principles for the Visual Novice* (Berkeley: Addison-Wesley–Peach Pit, 1994).

²¹For a longer exposition of my definition of rhetoric, see “Defining Rhetoric”, 26 April 2008 <<http://ideoplast.org/rw/portfolio-2007-2008/rhetoric.html>>.

is another key rhetorical practice for them.

Of course, an analytic understanding of rhetoric alone is not sufficient qualification for the designer, who also must understand how to employ rhetorical principles and devices (what might be called “synthetic” or “applied” rhetoric). And it should be equally obvious that the designer needs to understand the tools of the trade. Web page production is not an innate human skill like speech; it depends on complex information technology that must be mastered by the designer. In other words, designers must be writers: they need to understand not just how to create content, but how to implement it using the appropriate equipment.

Web pages and applications naturally have semiotic user interfaces. Semiotic meaning-production must be directed if the audience is to produce a model that is consistent with the presenter’s, and a consistent model is necessary for a successful user interface. Such direction has historically been the province of rhetoric. Then again, the interface must be appealing, or at least not onerous, to avoid discouraging users; and that task too has often been associated with rhetoric, or with one of its cognate fields like poetics or *belles lettres*. So user interface design is intrinsic to web design, and is a modern manifestation of rhetoric. The skill set for web design includes not only proficiency with the technology but knowledge of rhetoric, understood in the broad sense, as well. And that is—or should be—the skill set of a professional writer.