

Visual design for the user interface, Part 1: Design fundamentals.

Yale C/AIM Web Style Guide

Visual Design for the User Interface

Part 1: Design Fundamentals

Patrick J. Lynch, MS

Yale Center for Advanced Instructional Media
Published 1994, Journal of Biocommunications,
21(1):22-30

[Contents](#)

Abstract

Digital audiovisual media and computer-based documents will be the dominant forms of professional communication in both clinical medicine and the biomedical sciences. The design of highly interactive multimedia systems will shortly become a major activity for biocommunications professionals. The problems of human-computer interface design are intimately linked with graphic design for multimedia presentations and on-line document systems. This article outlines the history of graphic interface design, and the theories that have influenced the development of today's major graphic user interfaces.

By the end of this decade digital audiovisual media and computer-based documents will be the dominant forms of professional communication in both clinical medicine and the biomedical sciences. Interactive computer-based instruction is becoming an essential component of medical education, supplementing or replacing many lectures, laboratory experiments and dissections throughout the curriculum. Today most diagnostic imaging techniques and patient case records are already at least partially digital, and by the turn of the century virtually all medical images, patient records, and medical teaching resources will be acquired, transmitted, and stored primarily in digital form. Communications theorists have advocated multimedia "paperless documents" since at least the 1940's (Bush 1945;

Engelbart 1963; Nelson 1987), but it was only in the late 1980's that computers powerful enough to store and display such documents became commonplace in hospitals and medical schools. High bandwidth networks of small computers are fast becoming the most influential medium for professional communication in science and medicine, and electronic documents will play an ever-increasing role in the education and clinical practice of medical professionals (Jessup, 1993; Lynch and Jaffe 1990; Shortliffe 1990).

Documents designed for the computer screen may contain and organize many forms of interactive media, including text, numbers, still illustrations or photographs, animations, visualizations of spatial or numeric information, and digital audiovisual material ([see Figure 1](#)). Due to the novelty of these computer-based multimedia (or hypermedia) documents, and to the conceptual difficulties of integrating many forms of media into cohesive presentations, there are no widely recognized standards for organizing electronic documents (Adsit 1992; Lynch and Jaffe 1990). The graphic design and illustration of multimedia electronic documents requires a thorough understanding of the principles and practice of user interface design. As a discipline interface design draws concepts and inspiration from such diverse fields as computer science, audiovisual media, industrial design, cognitive psychology, human-factors and ergonomic research, audiovisual design, and the graphic and editorial design of conventional paper publications. The principles and practice of graphic interface design will influence the professional lives of all biocommunications professionals, as new, highly audiovisual forms of digital communication media augment or replace existing forms of illustration, photography, video production, and print media (Patton 1993).

There are two salient problems in the design of multimedia documents: informing and guiding the computer user through a complex body of information, and the creation of a visual design rhetoric appropriate for interactive computer

displays. Both problems are intimately linked with the design of graphic user interfaces for computer systems. The graphic user interface (GUI) of a computer system includes the interaction metaphors, images and concepts used to convey function and meaning on the computer screen, the detailed visual characteristics of every component of the graphic interface, and functional sequence of interactions over time that produce the characteristic "look and feel" of graphic interfaces.

Origins of graphic user interfaces

Computers and computer software operate through largely invisible systems that provide few physical or visual clues to the operational state or organization of the system (Norman 1993). The potential complexity and functional plasticity of computer systems is both their major strength and most obvious weakness-changing the software or operating system can radically change the characteristics and behavior of the computer system. The purpose of graphic user interface design is to provide screen displays that create an operating environment for the user, forming an explicit visual and functional context for the computer user's actions. The graphic interface directs, orchestrates, and focuses the user's experiences, and makes the organizational structure of the computer system or multimedia document visible and accessible to the user.

In the 1960's most truly interactive computer systems were typewriter-like teletype (TTY) terminals that used paper as a display, printing both the instructions from the computer operator and whatever responses resulted from the computer's activities. Early designers of interactive computer systems using cathode ray tube (CRT) display monitors created graphic and text displays modeled after their familiar single-line-at-a-time TTY paper displays. This teletype metaphor (the "glass teletype") for computer displays is the basis for the [MS-DOS](#) operating system's command-line screen display that is still in wide use today. However, even in the 1960's researchers such as Ivan Sutherland (inventor or

the first interactive windowing computer display) and Douglas Engelbart (inventor of the computer "mouse") were designing spatial display systems for CRT screens that both emulated the graphic complexity of print documents and used the dynamic character of the computer display to transcend the limitations of graphics printed on paper (Engelbart 1963; Grudin 1990; Sutherland 1980).

Direct manipulation interfaces

During the 1970's the conceptual basis for most current graphic user interfaces was developed at [Xerox's Palo Alto Research Center \(PARC\)](#). These concepts include explicit on-screen graphic metaphors for objects like documents and computer programs, multiple overlapping windows to subdivide activities on the display screen, and direct manipulation of windows, icons and other objects within the interface using Engelbart's desktop mouse as a pointing device (Smith 1982). Two factors influenced the development of modern graphic interfaces: the direct manipulation of graphic "objects" on the computer screen, and the creation of appropriate interface metaphors-graphic representations designed to encourage and complement the user's understanding of the computer system.

The Xerox PARC work on direct manipulation computer interfaces was grounded in the observations of cognitive and developmental psychologists Jean Piaget and Jerome Bruner (Bruner 1966; Piaget 1954) that our understanding of the world is fundamentally linked to visual stimulation and the tactile experience of manipulating objects in our environment (Kay 1988; Kay 1990). In particular, Bruner's model of human development as a combination of enactive skills (manipulating objects, knowing where you are in space), iconic skills (visually recognizing, comparing, contrasting), and symbolic skills (the ability to understand long sequences of abstract reasoning) lead PARC researchers to try and build interfaces that explicitly addressed all three of these fundamental ways of understanding and manipulating the

world around us. Computers (then and now) have always required abstract reasoning; the task of the PARC researchers was to create an interface that would also exploit the user's manipulative and visual skills.

Using the mouse as a pointing device the PARC team created a on-screen cursor whose movements directly corresponded to mouse movements, and a highly graphic display screen that allowed easy combinations of text and graphics. Working with the PARC computer scientists graphic designer Norman Cox created a set of screen icons (documents, folders, mail boxes, etc.) to make basic components and operations of the computer system visible as concrete objects (Littman 1988). The objective was to create on-screen graphic analogs of familiar real-world objects, to foster the illusion that digital data could be picked up, moved, and manipulated as directly and easy as paper documents on a desktop. Laurel (1991) and many previous observers have noted the similarities between direct manipulation interfaces and Samuel Coleridge's concept of the "willing suspension of disbelief," a term Coleridge coined to describe the audience's intense psychological involvement with representations of reality in theatrical plays. A well-designed graphic interface establishes consistent and predictable behavior for all objects represented in the system, and thus the user suspends disbelief and comes to treat on-screen representations as if they were real, manipulable objects like physical documents, buttons, and tools (Schneiderman 1992). The interface research done at Xerox PARC in the mid-1970's established most of the visual and functional conventions of current graphic user interfaces, and were the direct ancestor of the [Apple's](#) Macintosh graphics interface (Apple Computer 1992), [Microsoft's](#) Windows (Microsoft Corporation 1992) and the various graphic interfaces that overlie UNIX workstations such as Motif, [NextStep](#), or Open Look (Hayes 1989).

Fundamentals of graphic interface design

Unlike the static graphics of conventional print

documents, or the fixed linear sequences of film and video, graphics on the computer screen are interactive, dynamic, and constantly change in their presence or absence on-screen, in spatial position, and in visual or functional character. The visual structure of a graphic user interface consists of standard objects such as buttons, icons, text fields, windows, and pull-down or pop-up screen menus. Through their familiarity, constancy, and their visual characteristics, these interface objects convey very particular messages to the user about the functional possibilities and capabilities of the software in use. This constancy of form and function is a fundamental tenet of graphic user interfaces-the behavior of interface elements should always be consistent and predictable. Graphic interfaces also offer a visual and functional theme or metaphor to the user. Interface metaphors use references to familiar habits, tasks, and concrete objects as a means of making the abstract and invisible functions of the computer easier to understand and remember.

Interface metaphors

After some experience with a complex, abstract system like a computer users begin to construct a conceptual model or "user illusion" (Kay 1990) of the system as they imagine it to be organized. This mental model allows the user to predict the behavior of the system without having to memorize many abstract, arbitrary rules (Norman 1988). The primary goal of interface design is to create and support an appropriate and coherent mental model of the operations and organization of the computer system. Graphic user interfaces incorporate visual and functional metaphors drawn from the world of everyday experience to help orient the computer user to the possibilities and functions of the computer system. By emulating the look and behavior of familiar, concrete screen objects such as file folders, paper documents, tools, or trash cans the functions of the computer system are made visible and placed into a logical, predictable context.

One of the most familiar and widely imitated metaphors is the "desktop" interface created at

Xerox PARC in the 1970's for the Alto and Star computers. The designers at PARC reasoned that since those small computers would be used in an office environment an on-screen emulation of everyday office objects would make the computers easier to understand. The Alto and Star systems were the first computers to employ graphic icons representing commonplace office objects to represent documents, file folders, trash cans, mail boxes, and "in" and "out" boxes to represent other office fixtures. Interface metaphors facilitate what Norman (1993) calls experiential or reactive cognition, where you gain information about the functionality of the computer as you interact with various objects in the interface. You don't memorize commands- you react to a rich set of information presented by the graphic interface. Various interface elements both tell the user what actions are possible- the items listed on a pull-down menu, for example. The proper function of objects ought to be self-exemplifying through metaphor: to throw things away, put them in a "trash can", to store things, put them in a folder. But simply adding graphics and a mouse to the user interface does not automatically make a system easy to use or understand. As computer system mature and add capabilities many computer users now (justly) complain about the functional and visual complexity of current graphic user interfaces. Although graphic interface metaphors are widely accepted they are often poorly executed, resulting in software that is difficult to understand and use. Difficulties in the design of graphic interfaces most often arise because from two problems: inconsistent or confusing relationships between interface objects, and poor visual design of the computer screen.

Successful interface metaphors should be simple systems that do not require the user to learn and remember many rules and procedures. If the user is forced to remember many arbitrary rules the primary value of the metaphor is lost, because the "rules" governing the user's interactions ought to be self-evident in the

metaphor. For example, after placing a document icon inside a folder you ought to be able to then open the folder and see the document inside. You naturally assume that the document will stay inside the folder until you move it, and that you could put one folder inside another just as you can with real physical folders. If any of these assumptions were not consistently supported throughout the user interface the whole concept of folders as an organizational metaphor would be pointless. Most document metaphors are based on book or paper models because most people are familiar with the basic organization of books, but designers of electronic documents often neglect to fully support the print metaphor with page numbers, chapters, contents displays, or an index. [Figure 1](#) shows the design of medical teaching application that uses a print-like screen metaphor, with paging buttons and page numbers at the bottom right of the screen. Successful interface metaphors draw heavily on the user's knowledge of the world around them, and on established conventions that allow the user to predict the results of their actions in advance (Norman, 1988).

In well-designed, well-documented user interface systems such as the [Apple Macintosh](#) or [Microsoft Windows](#) graphic interfaces the proper functional and visual design of all standard interface metaphors and other elements is thoroughly described (Apple Computer 1992; Microsoft 1992). Although the graphic design and illustration of computer documents may involve many issues not explicitly addressed in standard interface guidelines the visual designer of computer documents should nevertheless be thoroughly familiar with the functional standards of the particular graphic user interface system in use. Unfortunately there is no digital equivalent of the Chicago Manual of Style (1982) for the design of multimedia computer documents. Most current graphic interface standards were written with tool-oriented software in mind, and are only now beginning to incorporate guidelines for the integration of text, graphics, hypermedia links

(see Lynch and Jaffe, 1990), and audiovisual media within computer software documents. In the absence of widely agreed-upon editorial standards for computer documents visual designers must proceed carefully to avoid creating systems that are more confusing than helpful to the computer user. The graphic interface standards set by Apple and Microsoft offer some of the few consistent stylistic and functional guidelines available to computer document designers.

Modality

Software modes exist to provide special (usually temporary) interpretations or contexts for the actions of the user. Poorly designed modal behavior can confuse users and artificially limit their freedom of action. For example, early word processing programs required the user to enter a "Copy Mode" before selecting and copying blocks of text. Once the user enters copy mode no other text editing actions were possible until the user left the copy mode. Although all complex software inevitably incorporates some modal behavior early personal computer software was often highly modal, and therefore was often difficult to learn to use. In the late 1970's and early 1980's most personal computer software interactions followed a "verb-noun" model of user interface design that relied heavily on modal states. Verb-noun models of interaction relied on modes primarily to limit the user's range of action, because by artificially restricting the user's range of choices the software was much easier to program. To paste a piece of text you had to enter "paste mode" (the verb), then select the text (the noun) to be pasted. This style of interacting with computers is often confusing because it is very easy for users to forget which mode they are in, and it is difficult to remember the commands to get into and out of all the modes within a complex program (Schneiderman 1992). Current graphic interfaces like Windows or the Macintosh operating system follow a generally modeless noun-verb model of user interaction. For example, to copy a piece of text you point and select the text (the noun), then

copy the text (verb) to the new location. No special modes restrict the user's actions.

However, not all software modes are detrimental or confusing. Most graphics software incorporates mild forms of modal behavior in drawing and painting tools. When a "paint brush" is selected the cursor typically changes to a unique brush cursor, and from then all of the users actions are interpreted as painting-related until another tool is selected. As long as the mode makes sense to the user (painting with a brush-like cursor in a graphics program) and the shift in context is clearly signaled (by changing the cursor, or highlighting a tool in a tool palette), then well-designed modes may actually make the software easier to understand and use (Apple Computer 1992). In general the use of restrictive modal behavior should be avoided in electronic documents unless there is a logical, highly functional purpose to restricting the user's freedom of action.

Locus of control

The user should always feel in direct control of the computer interface, and should never feel that the computer has "automatically" taken actions that could arbitrarily change the user's preferences, destroy data, or force the user to waste time. Well designed interfaces are also forgiving of user's mistakes, and are stable enough to recover "gracefully" if the user makes mistakes, supplies inappropriate data, or attempts to take an action that might result in irreversible loss of data. For example, it is very easy for programmers to change basic system variables like screen colors, the colors of standard interface objects, sound volume settings, or other visual and functional aspects of the interface normally controlled by the user through "Control Panels" or "Preferences" features of the operating system. These actions are strongly discouraged by the Macintosh and Windows interface guidelines, because these fundamental choices about the set-up of the computer should always be left exclusively to the computer user. (By analogy, imagine what it might be like if advertisers could control the

volume level or brightness of your television set during commercials.) Abrupt changes in the perceived stability and constancy of the interface are confusing to the user and rapidly lead to a lack of confidence in the design integrity and reliability of the computer system. For similar reasons the interface guidelines for most graphic interfaces strongly discourage programmers from attaching any consequences to moving the cursor around the computer screen (Apple Computer 1992; Microsoft Corporation 1992). Users correctly assume that they are free to move the cursor around the screen, and that only after explicit action is taken (by pressing the mouse button and clicking on a screen control object like a button or window) will there be any action taken by the computer.

Feedback and time in the interface

Proper management of time is essential in user interfaces. Computer users engage in a complex dialog of event and response, action and reaction with the operating system and user interface of their computers. Interface feedback is the process of managing the timeliness and manner of the computer's response to a user's actions. Feedback from the user interface should be immediate and unambiguous, in the form of visual or auditory signals that the computer has received input from the user and is acting upon that stimulus. Even small gaps in time (0.25-0.50 seconds) between the user's actions and any reaction from the computer can confuse the relationship between cause and effect, or force the user to assume that the computer or software has misinterpreted the user's actions (Horton 1990; Marcus, 1992). Visual signals that provide feedback from the interface are fundamental design features that are often overlooked until they are poorly executed or absent. In most graphic interfaces clicking on a screen button momentarily causes the button colors to reverse (white buttons turn black for a second) as an explicit signal that the button was "pressed" or clicked on. Since tactile cues are absent in these "virtual" screen buttons explicit visual or audible cues (playing a button "click"

sound, for example) are necessary to give users confidence that their actions are "understood" by the computer and are being processed.

Our expectations about the "normal" speed of events is determined by the world around us, not by the slower and sensory-poor environment depicted on the computer screen. It doesn't take much computer experience to realize that personal computers process information too slowly to mimic the speed at which most "real world" events occur. This technological limitation will disappear within a few years as silicon-based "reality engines" bring high speed, fully shaded animations and high-quality video to the personal computer. However, at today's more modest computing speeds interface designers must carefully manage processing delays in the user interface, and provide users with feedback the proper visual, text, and other on-screen cues about the state of the computer's operations at any give moment. In addition to the immediate visual feedback after the user clicks on a button (confirming that some process has been initiated), the interface should always give the user a visual signal to wait while the system processes information even if the delay is only a second or two. Any delay longer than a few seconds without any indication of normal processing activity (such as the Macintosh "watch" cursor, or the Windows "hourglass" cursor) is likely to be interpreted as at least or troubling ambiguous behavior. Long delays without feedback are likely to be seen as system or program errors (Apple Computer 1992; Microsoft Corporation 1992).

Computers excel at storing and retrieving information, but in one important sense most personal computers have very little memory. Although today's computer interfaces may often be a bit slow at providing information the instant the user requests it, by design today's graphic interfaces are largely trapped in the immediate moment and provide little evidence of the history of a user's interactions with the computer. For example, even the most advanced graphic user

interfaces usually support only one level of the "undo" command; the system only remembers the user's last action and has no other record of the user's previous interactions with the system. This lack of memory is particularly unfortunate in multimedia teaching or testing systems, where the user could often benefit from a detailed record of past actions, lists of screens that were visited, or a record of the sequence of actions that lead to a particular result. Multiple levels of "undo" could also prevent mistakes and data loss where the user did not realize there was a problem until many further steps had been taken and no single-step "undo" was possible. Some applications have started to implement "historical" features that record as least some aspect of the user's interactions with the program over time. HyperCard's "Recent" screen ([see Figure 2](#)) gives the user a chronological listing of the last 42 screens (or "cards") visited during the current session (Apple 1991). Users can quickly scan a graphic review of their HyperCard session, and "back up" to a previous screen by clicking the image of the screen. As system software becomes more sophisticated software "agents" can be designed that can learn and remember the user's action over longer periods of time, and process this information to help predict the user's needs, or provide a detailed "audit trail" over an extended period of time so that almost any action could be identified and reversed if necessary.

Organizing information

Most of our modern concepts about structuring information stem from the organization of printed books and periodicals, and the library indexing and catalog systems that grew up around printed information. The "interface standards" of books in the English-speaking world are well established and widely agreed-upon, and highly detailed instructions for creating books may be found in guides like *The Chicago Manual of Style* (1982). Every feature of a book, from the table of contents to the index and footnotes has evolved over the centuries, and readers of early books faced some of the same organizational

problems facing the users of hypermedia documents today. Gutenberg's bible of 1456 is often cited as the first modern book, yet even after the explosive growth of publishing that followed Gutenberg it took more than 100 years for page numbering, indexes, tables of contents and even title pages to become routine features of books. Multimedia and hypermedia documents must undergo a similar evolution and standardization of the way information is organized and made available in electronic form.

Highly audiovisual and interactive computers have lead designers to propose novel spatial and conceptual metaphors in data organization and storage, and many digital information theorists have explicitly rejected print standards as an organizing metaphor in electronic documents in favor of hypertext metaphors (Landow 1989; Nelson 1987). Unfortunately many readers find the hypertext or hypermedia disorienting and difficult to navigate through, and lately the interest in complex hypertext systems has cooled as designers struggle with the task of creating systems that incorporate the unique capabilities of computers without disorienting the reader (Gygi 1990; Norman 1990). There seem to be no widely agreed-upon spatial topologies or other organizing principles for an multi-dimensional electronic information space (Conklin 1987; Norman 1993), and it is proving to be very difficult to give the reader of free-form electronic information databases an understandable conceptual model that represents a complex, interconnected web of both existing and potential links between units of information.

The most practical current solutions to the organization of electronic documents build upon widely established print metaphors while gradually incorporating search, retrieval, and associative linking functions that are only possible in computer documents. Graphic maps ([Figure 3](#)) that give an overview of information structure are make it easier for users to establish a sense of location within the organization of electronic documents (Ambron and Hooper 1988;

1990). Figure 3 artwork courtesy of [Anne Altemus](#), National Library of Medicine. Standard elements of graphic interfaces such as pull-down menus ([see Figure 4](#)) can form a highly interactive "table of contents" that both gives the reader a constant reference to the information topics available, and using menu checkmarks or other signals to mark the current location also gives the reader a sense position within the document (Lynch, et al. 1992). Building a conceptual model that tells the user what is possible within the document, and makes explicit the organizational structure of the document.

Summary

The world-wide digital communications networks that are now being built will dramatically improve the availability and flexibility with which medical and scientific information may be stored, transmitted, and retrieved, but the benefits and opportunities offered by the new digital media will only be fully available to those biocommunications professionals able to create publications and audiovisual systems specifically designed for highly interactive digital media.

In spite of all of the obvious power, efficiency, and flexibility of digital media, it is a curiously disembodied form of communication. Unlike older media such as print or even videotape, digital information has no required physical form, and one of digital media's main advantages is precisely that it can change form and arrangement in response to the user's interactions. The homogenous, highly abstract, and largely invisible form of digital media requires an interface to give form and accessibility to information. Human interface design, as applied to the design of interactive digital audiovisual systems and electronic documents, will shortly become the dominant activity of many biocommunications professionals.

Digital display screens pose unique challenges to graphic designers and medical illustrators. The second part of this paper concerns the visual design of digital multimedia systems.

Literature Cited

Adsit, K. I. 1992. Designing a user interface and computer screens for instruction: Some considerations. *Journal of Biocommunication* 19 (2): 2-7.

Ambron, S. and K. Hooper, eds. 1988. *Interactive Multimedia*. Redmond, WA.: Microsoft Press.

Ambron, S. , and K. Hooper, eds. 1990. *Learning with Interactive Multimedia*. Redmond, WA: Microsoft Press.

Apple Computer, Inc. 1987. *Human interface guidelines: The Apple desktop interface*. Reading, MA: Addison-Wesley.

Apple Computer, Inc. 1991. *HyperCard 2.1*. Cupertino, CA: Apple Computer, Inc.

Apple Computer, Inc. 1992. *Macintosh human interface guidelines*. Reading, MA: Addison-Wesley.

Bruner, J. 1966. *Towards a theory of instruction*. New York: W. W. Norton.

Bush, V. 1945. As we may think. *Atlantic Monthly* 176 (1): 101-108.

Conklin, J. 1987. Hypertext: An introduction and survey. *IEEE Computer*, September, 1987, 17-41.

Engelbart, D. C. 1963. A conceptual framework for the augmentation of man's intellect. In *Vistas in information handling.*, ed. H. Howerton and Weeks, B. Washington, DC: Spartan Books.

Grudin, J. 1990. The computer reaches out: The historical continuity of interface design. In *Empowering people: CHI '90 conference proceedings.*, ed. J. C. Chew and J. Whiteside. Reading, MA.: Addison-Wesley.

Gygi, K. 1990. Recognizing the symptoms of hypertext...and what to do about it. In *The art of human-computer interface design*, ed. B. Laurel. Reading, MA: Addison-Wesley.

Hayes, F. and N. Baran. 1989. A guide to GUI's. *Byte* 14 (7): 250-257.

Horton, W. K. 1990. *Designing and writing online documentation*. New York: Wiley.

Jessup, M. E. 1992. Update in biomedical visualization: The professional communicator's role. *Journal of Biocommunication* 19 (3): 2-7.

Kay, A. 1988. Doing with pictures makes symbols: communicating with computers. Videotape., Stanford, CA: University Video Communications.

Kay. 1990. User interface: A personal view. In *The art of human-computer interface design.*, ed. B. Laurel. 191-207. Reading, MA: Addison-Wesley.

Laurel, B., ed. 1990. *The art of human-computer interface design.* Reading, Mass.: Addison-Wesley.

Laurel, B. 1991. *Computers as theatre.* Reading, Mass.: Addison-Wesley.

Littman, J. 1988. Keeping it simple: Norm Cox. *MacWeek*, September 20, 18.

Lynch, P. J., C. C. Jaffe, P. I. Simon, and S. Horton. 1992. Multimedia for clinical education in myocardial perfusion imaging. *Journal of Biocommunication* 19 (4): 2-8.

Lynch, P. J. and C. C. Jaffe. 1990. An Introduction to Interactive Hypermedia. *Journal of Biocommunication* 17 (1): 2-8.

Marcus, A. 1992. *Graphic design for electronic documents and user interfaces.* New York: ACM Press, Addison-Wesley.

Microsoft Corporation. 1992. *The Windows interface: An application design guide.* Redmond, WA: Microsoft Press.

Nelson, T. 1987. *Dream machines/Computer lib.* Redmond, WA: Tempus Books.

Norman, D. A. 1988. *The psychology of everyday things.* New York: Basic Books.

Norman, D. A. 1990. Why interfaces don't work. In *The art of human-computer interface design*, ed. B. Laurel. Reading, MA: Addison-Wesley.

Norman, D. A. 1993. *Things that make us smart: Defending human attributes in the age of the machine.* Reading, MA: Addison-Wesley.

Patton, P. 1993. Making metaphors: User interface design. *ID* 40 (2): 62-66.

Piaget, J. 1954. *The construction of reality in the*

child. New York: Basic Books.

Schneiderman, B. 1992. Designing the user interface: Effective strategies for effective human-computer interaction. 2nd ed., Reading, Mass.: Addison-Wesley.

Shortliffe, E. H., and L. E. Perreault, ed. 1990. Medical informatics: Computer applications in health care. Reading, MA: Addison-Wesley.

Smith, D. C., C. Irby, R. Kimball, and B. Verplank. 1982. Designing the Star user interface. Byte 7 (4): 242-282.

Sutherland, I. E. 1980. Sketchpad: A man-machine graphical communication system. Reprint of 1963 MIT thesis., New York: Garland Publishing.

Tognazzini, B. 1992. Tog on interface. Reading, MA: Addison-Wesley.

University of Chicago Press. 1982. The Chicago manual of style. 13th ed., Chicago: University of Chicago Press.

