Designing the Human-Computer Interface

Harold J. Goldes

Introduction

Computers present man with special opportunities and singular dilemmas. Computers are neither human nor like other machines. Cars, stereos, toasters, the lathe, and drill press all depend on human operation and control. Computers have an extra dimension—they can be programmed, and this allows them a significant degree of independence. Programmable computers are different from tools which extend human muscular and sensory capacity. Computers extend the human ability to plan, and planning is a signal property of human mental life.

Planning is a function of the human ability to use symbols to represent and manipulate objects and events without regard to their reality in space and time. The computer's capacity to perform in ways that seem to mimic human thought has sharpened some fundamental concerns humans have about machines in general and computers in particular:

1. What is the relationship between humans and machines?
2. How are humans and machines different?
3. Who is in charge of whom?

My subject is, primarily, improving the relationship or interface between humans and computers, but I will first discuss some obstacles to this improvement: the doubts and fears man has about the "machine."

Doubts and Fears

Images . . .

Films are capable of presenting images that express our fears with the non-rational intensity of nightmares. Earlier in this century, two films, "Metropolis" (1926) and "Modern Times" (1936), commented dramatically on the relationship between man and machine.

Harold J. Goldes is a Consultant in Organizational Analysis and Research at Merrill Lynch, and a doctoral student at Teachers College, Columbia University, New York.

"Metropolis" depicted the relationship as dismal and parasitic. Machines existed at man's expense. The human workers in Metropolis, the future city, performed limited tasks repetitively. A worker collapses at the controls of a machine and is replaced by another. Human beings are depersonalized and interchangeable. Men and machines are not very different except that in "Metropolis," machines rule and in one scene literally consume men (Baxter, 1970, p. 29).

"Modern Times" also offers a predatory image: Charlie Chaplin enmeshed in a machine and helpless in its grip. The film suggests that the relationship between man and machine is dehumanizing because man must work to match the machine's pace. To do this, he must labor tirelessly and mindlessly (Ginzberg, 1982).

A half-century later, the same Chaplin character is redefining the man-machine relationship. IBM Personal Computer advertising shows the little tramp at ease; his feet up on a table; a keyboard resting in his lap. The message is: people can relax. Man is in control. The computer *is* a different kind of machine!

. . . and Other Thoughts

True. Computers differ from other machines. But how do they differ from *man*? Computers have been called "thinking machines" and "electronic brains," suggesting that computers can trespass on man's unique preserve: the mind.

If computers can duplicate mental processes, made to think, then man is eclipsed. A computer could think like a human if human thought could be formalized as a series of rules to direct the machine. Weizenbaum (1974) argues that the problem of describing the laws of human thought on paper begs the issue. Being human implies having objectives, drives, and problems rooted in our "unique biological and emotional needs" (p. 223). Because the source of human behavior is the human condition, Weizenbaum questions the possibility of ever developing computers "whose range of thought is to be coextensive with that of humanity" (p. 197).

Bronowski (1971) emphasizes this essential difference between human and computer by presenting two decisions he can make about flicking his handkerchief. The first one involves just the physical consequence: "I won't make that flourish again because last time it hit my wife." The second carries the understanding of that consequence: "I won't make that flourish again because last time it embarrassed my wife" (p. 24). The first decision could be translated into rules for avoiding a fatal flick. But could a machine be told how to make the second decision? Bronowski thinks not. It is more than a decision. It is a choice founded upon empathy for his wife and upon his own knowledge of how *he* felt when embarrassed.

Human Factors and Interface Design

In contrast to Weizenbaum, Bronowski, and Chaplin, human factors or ergonomic studies offer an analysis of man-machine relationships that treats the human component as unique (Chapanis, 1956), and the modification of the machine to fit man (rather than fitting man to machine) as the correct priority (Sinaiko and Wallace, 1961). As human factors research also "supplies information to the process of designing the man-computer interface" (Clark, 1981, p. 278), it is an invaluable aid to the development of benign or user-friendly boundaries that, like Frost's fences, make good neighbors.

The human-computer interface is the physical and conceptual locus of human-computer interaction. It is represented physically by the computer terminal and conceptually as a virtual workspace that corresponds to some real model the user already knows.

Creation of the interface is a design process that seeks to identify an appropriate model or "defining metaphor" (Thomas and Carroll, 1981, p. 260). The model is translated into computer-based organizations and structures that will represent state of the system to the user. The design process entails an attempt to identify the specific human factors, behavioral and psychological, that the interface must engage, support, and amplify (Canfield Smith, Irby, Kimball, Verplank, and Harslem, 1982, p. 246).

The interface design philosophy is based to a large extent on the view of man-machine relationships taken by human factors research. The human factors perspective sees a man-machine system whose characteristics are those of both the human and non-human components (Sinaiko and Wallace, 1961). This system functions as well as the components and their fit. The human component of the system is versatile, but the machine component is pliable. Machines can be designed and redesigned. By examining the strengths and weaknesses of the human component (the human factors), machines and man-machine interfaces are designed to enhance the efficiency of the system.

Abating Computer Fear: The Hardware and Software Shall Lie Down With the Wetware

As if to counter fears and insecurities, design principles for the human-computer interface suggest distinct and optimistic themes of intimacy and cooperation. On one hand, we are told that integrating human sensory characteristics and computer hardware specifications can yield man-machine systems that "function symbiotically" (Cahill, 1981, p. 45) and "synergistically" (Canfield Smith *et al.*, 1982, p. 272). Conflict and frustration can be avoided by ensuring that the computer does not "compete with man in areas where man is superior" (Martin, 1973, p. 7).

But conflict and confusion do contaminate human-computer interactions. People perceive in computers an authority which betrays a basic fear. Thomas and Carroll (1981) describe one assumption about authority made by new users of a text editing system who were asked to interpret the word "command." About 70 percent thought commands were directives given to them by the computer (p. 259).

Clark (1981) found a tendency for some people to act upon computer-issued instructions as if they were orders. In a study to evaluate "HELP" panels (screen displays to assist users of a chart creation utility), Clark noted that actively worded instructions: "To draw the picture, press key PF5," were preferred by subjects to passive wording: "Pressing key PF5 will cause the picture to be drawn" (p. 289).

But subjects also interpreted the active form as an "order." As a result, keys were pressed almost immediately on displays where they had no effect. Users who had been "ordered" to press these keys now became frustrated by their inactivity.

Sim (1975) suggests that certain "meta messages" are embedded in human-computer dialogues. The meta messages convey the authority of the computer and the subordinance of the human user. Although the intended communication of the dialogue may be friendliness, the actual subtext may read "Understand or you're a dummy." Intolerance, rigidity, and inflexibility become the undertones that transform interactive dialogue into interrogation.

Examining the Interface from the Outside in

Let's probe the human-computer interface starting with the terminal—for this discussion, a CRT. The CRT screen can represent images, objects, and text in two or three dimensions, monochromes, halftones, and colors. Properly configured, a CRT accepts human input from a variety of devices.

Ergonomic analyses of CRT design (Cooper, Thain Marston, Durrett, and Stimmel, 1982; Lu, 1982; Stammerjohn, Smith, and Cohen, 1981) have noted characteristics which promote or impede human use, among them: noise produced by the terminal, screen glare, keyboard height, keyboard tactile feedback, keyboard layout, glare produced by the reflective key surface, detachable keyboards, separate numeric keypads, separate cursor control keys, "soft" or program definable keys, and radiation levels emitted by screen phosphors.

In a concise review, Winkler and Konz (1980) identify dominant ergonomic parameters affecting display readability for alphanumeric information. These include character shape, size, width-to-height ratio, font, and figure-ground contrast. Within a given display, placement of the topmost text line can cause an increase in glare and operator head-tilting. Enlarging character size forces the viewer to increase the amplitude of eye and head movements because of the increased viewing angle subtended by the display (Hirsch, 1981, p. 150). Display flicker contributes to eye fatigue. Multicolor palettes may reduce resolution but may also increase, through color encoding, the information content (Helander, 1981; Lu, 1982).

Sugar-coating gets some users close enough to terminals to appreciate less glamourous, but more significant, features. To the business executive, the keyboard connotes secretarial status, manual labor, or a data processing background. High technology in the research laboratory is acceptable. Computer components on the desktop are not. For the executive, the computer is the source of anxiety ("Must I learn to program it?"), and an information resource. Reducing terminal weight and dimensions and attending to user psychology in packaging terminals are the elements of style used to overcome resistance at this level.

Evolution of the Interface

The need for a well-designed human-computer interface is a response to a changing population of computer users. At first, programming language was the only means for human-computer communications.

Merely connecting terminal to computer and displaying the raw contents of computer memory is insufficient for communication to all but a small number of computer specialists. It is as if two people tried to converse by showing each other pictures of their brain waves. Brain surgeons might get the idea. Most of us would not.

It is entirely possible for humans to learn the binary language of computers and to conduct affairs on that very direct, yet cryptic, level. This was the way all programming was done during the first years of computing. There was no alternative. It was tedious work, susceptible to errors as simple as substituting a "0" for a "1." It was a task for people whose patience and commitment might, in earlier times, have led them to copy Torah scrolls.

Assembler languages were a step forward for the human user. Assemblers wrapped binary codes in mnemonics that were easier to remember and write. Each assembler instruction generated one binary instruction. This one-to-one correspondence did not directly produce more lines of code but did increase programmer productivity by placing the burden of translating assembler into binary onto the machine. This exemplifies a key tenet of interface design: reduce human memory demands. This was accomplished by capitalizing on human pattern recognition and information-encoding strategies, and by loading repetitive functions onto the computer.

High-level languages were the next development. Their purpose was to provide human users with languages as problem-solving tools rather than languages (like assembler) best suited to the manipulation of registers, memory locations, bytes, and bits. High-level languages have English-like commands which generated perhaps ten binary instructions and offer users more leverage over computer resources. Yet, some specialized languages like APL, Lisp, and Forth put power in the hands of only the sophisticated user while remaining esoteric to the vast majority of potential users.

For most of the short history of computing, users have been programmers. As a group, programmers are self-selected by their interest in computers, to be motivated, tolerant, and even captivated by the very intricacies and idiosyncrasies of computers that other types of users find discouraging and frustrating.

Now these specialists have been joined by the rest of humanity. The new user is "naive"; not interested in computers; not skilled in their use. Their interaction with the computers may be secondary to their primary task. Their education and their literacy cannot be assumed. Efforts to promote their access to information on computers may be insensitive to their needs and inappropriate to their abilities. As one observer has remarked:

> There are millions of people out there who can't read above an eighth-grade level, and they are expected to go through a hierarchical menu system that will take them through five levels before they get the information they need? (Weiss, 1982)

The unvarnished complexity acceptable to the knowledgeable user is intolerable to the casual or timid user. The design of the flexible, supportive human-computer interface is the unavoidable consequence of the widespread and spreading use of computers.

The Book: A Defining Metaphor for the Interface

What form will a well-designed interface take? Let's examine a book as a model. In considering the content of the book, we might say, "This book is a novel," or "This is a collection of plays." If it *were* a book of plays we could, if asked, add some details. For example, the book has a table which organizes the contents of the book. It is like a map of the book. We know that if we turned each page, starting with the first, we would encounter each of the items listed in the table of contents in the order listed. But the real power of the table of contents lies in the direct or random access it provides to each item.

The table of contents also lists an introduction and, because this is a book of Shakespeare's plays, a glossary as well. The introduction is another kind of map. It cites intellectual landmarks. After reading the introduction, we might seek a particular play because it contains a famous scene. The glossary is a third type of map. It is a map of original and, perhaps to us, obscure meanings. If the book had an index, we would have a table of contents for subjects, and we could read the book selectively, according to our needs, which might have nothing to do with the appreciation of the plays as plays.

What we are examining are the metacontents of the book: contents that refer to other contents of the book. This kind of reference is possible because the book is structured hierarchically. A hierarchy is a powerful way of organizing information, and this book is full of information about other parts of the book, other books, the author, and the contents of the book in relation to the contents of other books. It contains information about itself: how to read the contents, the players' lines, and how to imagine or prepare the setting of the plays. The organization and structures of the book make its contents accessible and useful to readers with varying degrees of interest, literacy, and purpose. A good human-computer interface would allow a user to interact with software in the same way. Inclusion of software structures like menus, HELP displays, icons, and frame markers spells the difference between user-friendly and user-hostile software. The practical consequence of designing software that ignores the needs of the user is software that is not used (Helander, 1981, p. 300).

The interface should reflect or resonate with some recognizable environment in the user's personal experience. That is why the interface design process begins with the development of a defining metaphor that enables the user to "understand and interact with the system" (Canfield Smith *et al.*, 1982, p. 252). To the extent the metaphor is (or can become) meaningful, it encourages use.

An example is the Xerox STAR user interface, which assumes as its defining metaphor, the office. The base display is a graphic representation of a "desktop." On the desktop are icons, small pictures of "documents, folders, file drawers, in-baskets, and out-baskets" (Canfield Smith *et al.*, 1982, p. 256). User interactions involve manipulation of these icons to perform the functions of their real-world analogs.

In general, the defining metaphor ought to be as explicit as is the STAR interface, which refers to a concrete "source knowledge structure" (Thomas and Carroll, 1981, p. 262). Where the metaphor is vague or inconsistent, the user will be confused. The book can both succeed and fail as such a model.

Weaknesses of the Book Model

In the design of many interactive applications, the defining metaphor has been print media. In these applications, the operations used to access information indicates their print cognates.

"Paging" is a term that describes the user's movement from frame to frame. "Scrolling" describes the movement of the user through the text but with a different source metaphor; text printed on a scroll (a fascinating juxtaposition of old and new media models). The user maneuvers a window "up" and "down" over the text. Some arrangements require left and right scrolling, too.

But scrolling confuses users. Which is the figure? Which is the ground? Is the window moving or is the text? Does scrolling "up" move the text up or the window down? (Bury *et al.*, 1982, pp. 385-394). Paging oversimplifies the problem of user orientation by understating a crucial difference between printed and computer-based text: the evanescence of electronic displays.

In a book, pages past and future remain physically present. Books provide rough kinesthetic clues about your position in them: the pages beneath your fingers tell you where you are. On computers, displays disappear. The next "page" is the next *logical* page.

The problem is an implied isomorphism between print source metaphor and electronic target domain. Certain features of books cannot be mapped onto electronic text. Other features are admirably suitable.

Strengths of the Book Model

The book is an eloquent defining metaphor when we examine just those features that offer the reader support and orientation. For example, as mentioned earlier, a book can be self-referential; informing its reader about its contents. The book preserves its contents thus relieving the reader of the task of memorizing them. It is consistent in its format and organization, and, most important, its contents indicate their logical and spatial relationships to each other through conventions like footnotes, citations, indices, and pagination. The book succeeds by providing support and orientation to its user. The interface should do the same.

Organizing Ideas for Interface Design

Support and Orientation are useful categories for organizing ideas on interface design. A review of the literature reveals many suggestions to guide design. In this section, an attempt will be made to group these suggestions under Support and Orientation. Since we have been sensitized against exhaustive mapping, a third category, personalization, is introduced. Personalization is a capability unique to interface design. It has no real correspondence in print media. Briefly, it describes the extent to which a user can custom tailor the interface to his or her requirements.

Support

1. The interface must support the user. One form of support is short-term memory support. The limitations of short-term memory are often cited as a factor in interface design (Canfield Smith *et al.*, 1982; Clark, 1981; Fried, 1982; Helander, 1981; Simpson, 1982). Miller's "magic number seven (plus or minus two)" limits the number of menu choices displayed. Frequently chosen menu items should appear at the top of the list. The item selected becomes the label for the display invoked by the selection, providing continuity between menu and invoked display.

2. Support is provided by using the human tendency to encode information in color and kinesthetic patterns to assist short-term memory. Support is achieved through consistency in display format, key use conventions, and terminology. As modes frequently confound key use conventions, key names ought to be meaningful. For example: the Epson QX-10 keys are organized functionally and, within functional groupings, labeled meaningfully: Typestyles—bold, italics, size, style; File Controls—store, retrieve, print, index, mail (Williams, 1982). PF (program function) or soft keys need labels. Key function needs to be consistent. For example: the same key should not page forward on one display and select options on another.

3. Support requires prompting, levels of prompting, and HELP panels. Support the user against himself. Ensure him against the consequences of his own actions. The interface must be forgiving. It should offer UNDO functions to reverse actions (Rutkowski, 1982; Weinberg, 1982). Potentially catastrophic actions (file deletions) should not be easy to perform and should require confirmation before execution. For example: a delete function should require a keypress combination or a keypress sequence. Deleting text in the PLATO text editor uses the shift and HELP keys to execute the command. This combination is seldom used elsewhere in the system, and when it is, the function is deletion.

4. When failure is inevitable, the interface must fail softly, permitting graceful degradation, not utter collapse of the system. This kind of support should be implicit: user selectable parameters and switches should have default settings to ensure that there will always be an outcome for a user action.

Orientation

1. Orientation in the interface implies stabilizing and positioning the user. Movement from display to display (interpanel navigation) can shift the user from one level of a hierarchy to another or into a new mode, i.e., a state which interprets user input in a specific way.

2. Interpanel navigation is fundamentally disorienting (Simpson, 1982, p. 114). The interface must provide positional landmarks (Clark, 1981, p. 287). Interpanel navigation requires maps and landmarks. The user's location within interface structures must be stated and tracked. Short-term memory decay (p. 287) limits the recall of display lineage. So, like Theseus unraveling his thread, the user, too, should have a means to retrace his or her steps back through a maze of displays.

3. Interpanel navigation should employ paging not scrolling (Simpson, 1982, p. 122). Display manipulation (intrapanel navigation) requires firm initial establishment of figure-ground relationships.

4. Entry into and exit from mode must be clearly announced. Once invoked, mode status should be distinctive. For example: when the Texas Instruments LOGO Sprite editor is invoked, the screen changes from its current color to green.

5. While the interface makes much transparent to the user, it is essential that the user be aware of processes and transitions. Every action initiated by a user should have a feedback message or some response (Canfield Smith *et al.*, 1982, p. 262). The screen should never go blank as an intermediate result of a command or while executing a command.

Personalization

1. The user should be able to customize the interface to his or her needs or experience level. Command and control terminology should be extensible; the user should be able to modify and

expand the interface. Display format and screen content should be flexible enough to bend to the user's wishes or taste.

2. Prompting should be optional and user-adjustable on a "need/want to know" basis. The interface should offer user-defined keys, functions, and parameters.

3. The user should be able to override the system. If an interface is calibrated to a given level of user sophistication, the experienced user should not be handicapped. A fast track of advanced features, short cuts, and enrichments should be available.

Conclusions and Speculations

The human-computer interface evolved in response to the needs of people to use computers. The user-friendly interface has appeared to meet the requirements of an expanding population of naive users who have difficulty and perhaps some fundamental concerns about interacting with computers. Creating the human-computer interface is a design process which begins with the analysis of human factors or user needs. The design process includes specification of a defining metaphor that is a conceptual model for the interface. The book lends itself in some ways as a strong defining model. The book works well as a source knowledge model by providing support and orientation to its users. Support, orientation, and personalization help categorize the principles underlying the interface design process.

Since the interface is a function of computer technology, how will the continued growth of this technology alter the interface?

Imagine primitives viewing a modern movie for the first time. Would they not be awed by the giant faces? Would they not shrink from the objects that rush toward them through a zoom lens? Would they not be confused by the jump cuts and dissolves that abridge and confound time and space as they understand them? In time, given a taste for this sort of thing, they would learn to watch movies without this wrenching disorientation. Cinematic conventions and real-world physics would coexist as they do for us. In this way, perhaps human sophistication and greater tacit knowledge of computer worlds will dampen the shock of human-computer interaction.

Perhaps not. It is more likely that we shall see technology and design experience lead to novel and refined ways of interacting with computers. Assume present trends continue. Memory cheapens. Processing speed increases. Storage capacity grows. What techniques for support, orientation, and personalization can we imagine?

Artificial intelligence (AI) research is the field that will shape the future human-computer interface. In place of the CRT, we will have subtlety and transparency. The interface will support interaction through dialogue. Speech input, true dialogues, will free users from the keyboard and the screen. The problem of orientation evaporates. Using the computer becomes no more disorienting than a local phone call to an intelligent colleague.

Personalization is achieved by having the machine respond in languages, male or female voices, levels of explanation keyed to inferences about the user's intelligence, or interest. Negroponte's "talking head" might provide the perfect cosmetic touch, allowing the user to have a problem-solving session with the bust of Descartes (Negroponte and Parker, 1981). In contrast to the Von Neumann machine, we might have a Proust machine capable of constructing "vast structures of recollection" (Proust, 1970, p. 36) from a few verbal crumbs of user reminiscence.

Display modes based on heads-up displays (especially displays projected onto the inner surface of a pilot's visor) suggest that enhanced graphics may permit an interface that overlays reality. Imagine reading a marketing report, and at the appropriate point, glancing at a blank page and seeing a graphic of the most recent sales figures from the corporate data base.

Imagine a room or the inside of a helmet as an environmental interface. The user edits audio-visual primitives that can be programmed to assume the shapes and sounds the user requires. These objects, perhaps they will be called "super sprites," can be modeled using templates from attribute libraries. Moreover, the super sprites can be matted into existing education and entertainment programs. One could create a likeness to be inserted into a movie in addition to or as a stand-in for a character. Dynamic editing would permit one to alter super sprite qualities during the program.

Information could also be represented as a complex holograph. One could take a walk through a report on agricultural productivity pausing to investigate, in depth, corn production in Kansas. Two-dimensional data could become tangible. "What if?" queries could produce immediate and substantial responses.

The possibilities defy easy description because so much is possible. In the computer, man has a tool that allows individuals to search, wonder, and play in ways beyond the reach of mere wealth. The computer to some is a kind of magic or perhaps sorcery. Some who never care to know how it is done can believe what they wish and enjoy it nonetheless. Others who want to know can learn the tricks, and in doing so, find the science that underlies the magic. ☐

References

Baxter, J. *Science Fiction in the Cinema.* New York: Paperback Library, 1970.

Bronowski, J. *Identity of Man.* New York: Natural History Press, 1971.

Bury, K., Boyle, J., Evey, J., and Neal, A. Windowing Versus Scrolling on a Visual Display Terminal. *Human Factors*, 1982, *24*(4), 385-394.

Cahill, M. *Digest of Technical Papers.* SID International Symposium, 1981, p. 45.

Canfield Smith, D., Irby, C., Kimball, R., Verplank, B., and Harslem, E. Designing the STAR User Interface. *Byte*, 1982, 7(4), 242-282.

Chapanis, A. *The Design and Conduct of Human Engineering Studies.* San Diego: San Diego State College Foundation, 1956.

Chaplin, C. (Producer) *Modern Times.* California: 1936.

Clark, I. Software Simulation as a Tool for Usable Product Design. *IBM Systems Journal*, 1981, *20*(3), 272-293.

Cooper, R., Thain Marston, P., Durrett, J., and Stimmel, T. A Human Factors Case Study Based on the IBM Personal Computer. *Byte*, 1982, 7(4), 56-72.

Fried, L. Nine Principles for Ergonomic Software. *Datamation*, 1982, *28*(12), 163-166.

Ginzberg, E. The Machinization of Work. *Scientific American*, 1982, *247*(3), 66-75.

Helander, G. Improving System Usability for Business Professionals. *IBM Systems Journal*, 1981, *20*(3), 294-305.

Hirsch, R. Procedures of the Human Factors Center at San Jose. *IBM Systems Journal*, 1981, *20*(2), 123-171.

Human Factors: Guidelines for Software Design. Minneapolis: Control Data Corporation, 1981.

Lu, C. Microcomputers: The Second Wave. *High Technology*, 1982, *2*(5), 36-52.

Martin, J. *Design of Man-Computer Dialogues.* Englewood Cliffs, NJ: Prentice-Hall, 1973.

Negroponte, N., and Parker, W. Talking Heads—Display Techniques for Personna. *SID 81 Digest*, 1981, *1*(2), 174-175.

Poole, H. *Fundamentals of Display Systems.* Washington, D.C.: MacMillan, 1966.

Proust, M. *Swann's Way.* New York: Vintage, 1970.

Rutkowski, C. An Introduction to the Human Applications Standard Computer Interface. *Byte*, 1982, 7(10), 291-310.

Sherr, S. *Fundamentals of Display Systems.* New York: John Wiley and Sons, 1970.

Sherr, S. *Electronic Displays.* New York: John Wiley and Sons, 1979.

Sim, F. On the Sociology of Computing: Conceptual Frameworks and Curriculum. *SIGSOC BULLETIN*, 1975, *7*, 9-21.

Simpson, H. A Human Factors Style Guide for Program Design. *Byte*, 1982, 7(4), 108-134.

Sinaiko, H., and Wallace, E. Human Factors in the Design of Systems. In H. Sinaiko (Ed.), *Selected Papers on Human Factors in the Design and Use of Control Systems.* New York: Dover, 1961.

Stammerjohn, L., Smith, M., and Cohen, B. Evaluation of Work Station Design Factors in VDT Operations. *Human Factors*, 1981, *23*(4).

Stamps, D. Human Use a Design Factor for Executive Workstations. *MIS Week*, October 13, 1982, p. 12.

Thomas, J., and Carroll, J. Human Factors in Communication. *IBM Systems Journal*, 1981, *20*(2), 237-263.

Ufa (Producer). *Metropolis.* Germany: 1926.

Weinberg, S. *Human Factors Guidelines: Designing Interactive Computer Applications.* Minneapolis: Control Data Corporation, 1982.

Weiss, H. Problem with Videotex: Still too Smart for Some Users. *MIS Week*, April 7, 1982, p. 10.

Weizenbaum, J. *Computer Power and Human Reason.* San Francisco: W.H. Freeman, 1974.

Williams, G. The Epson QX-10/Valdocs System. *Byte*, 1982, 7(4), 54-57.

Winkler, R., and Konz, S. Readability of Electronic Displays. *Proceedings of the SID*, 1980, *21*(4), 309-313.

Wooldridge, S. *Computer Output Design.* New York: Charter, 1975.