Behavior Research Methods & Instrumentation
1983, Vol. 15(2), 195-199

SESSION VI
HUMAN FACTORS IN COMPUTING

Human factors in user-computer interaction: An introduction

RICHARD F. DILLON
Carleton University, Ottawa K1S 5B6, Canada

The goal of human-factors research on human-computer interaction is to make computers usable. Describing the issues that determine whether a system is usable or not provides an introduction to the type of problems with which human-factors specialists are concerned.

Starting in the mid-1970s with the growth of home computers, computer networks, and office automation, there has been a corresponding growth in concern for the problems faced by humans using computers. This concern is the focal point for the field of human factors in computer use. The field is inherently interdisciplinary. Psychologists, computing scientists and system engineers, hardware engineers and industrial designers, sociologists, educators, library scientists, mathematicians, graphic specialists, and those concerned with using computers to aid the handicapped have all contributed to the field.

As the field has matured, research has been directed toward two distinct populations of computer users. In the first population, the users are computer professionals: programmers, computing scientists, software engineers, system engineers, and so on. For this population, the computer is not a means to an end, but rather the end itself. Here the research has been directed primarily toward methods of enhancing programming practice by refining programming techniques, improving system design methods, improving teaching of programmers, and improving selection of programmers. Schneiderman (1980) has delineated the field of human-factors problems for this population and given the area a name: "software psychology." The second population of computer users for whom human-factors research is oriented is the non-computer professional. For this user, the computer is not a central focus but is merely a means to an end. Moran (1981) uses "computer-user psychology" to describe research performed to aid this population of users. This paper provides an introduction to selected human-factors problems encountered by the second group.

This research was supported by Strategic Grant G0365 from the National Science and Engineering Research Council of Canada.

HUMAN INTERACTION WITH COMPUTER SYSTEMS

Sophisticated computers are now available and being used by hundreds of thousands of people who welcome computer capabilities but curse computer complexity. Human-factors research addresses the complexity problem; one goal of human-factors research with computers is to make computers easier to use. To appreciate the magnitude of the problem, it is constructive to consider the evolution of the interface between users and computer systems.

With large mainframe systems, an artificial buffer has been placed between the user and the computer. This buffer consists of highly trained professional computer personnel: computer operators who run the computer, system engineers and programmers who create and maintain system and application software, program librarians who keep track of software, engineers and technicians who take care of hardware, and keypunch or key-to-disk operators who prepare materials to be submitted to the computer. The typical computer center has an army of such individuals, all highly trained and knowledgeable about some aspect of the system, all working on a daily basis on the system and familiar with many obscure details, and all capable of and willing to master one or more volumes in the library of complex technical manuals that chronicle the dark secrets of system functioning. In many computer systems, the buffer between the user and the computer is impenetrable, with only the specialists permitted access to the system. All interaction with the computer is handled by personnel from the data processing department. In installations in which non-computer specialists are permitted to interact directly with the system, the users are often untrained and are typically one-time or occasional users who learn only enough about the system to accomplish their immediate needs and then promptly forget even those details. The typical computer center solution to this problem is to provide computer consultants who understand the seemingly trivial, but critical, details that will make an application program run. This buffer of specialists has made it possible for users to obtain desired results, typically using an application package, without knowing very much about the computer system. Clearly, the user-computer interface has always been poorly human engineered, but large amounts of money to support specialists have made it tolerable.

The protective buffer has been crumbling for a number of years as timesharing has replaced batch processing. The timesharing user has to know a little about terminals, a little about the operating system, a little about files, perhaps a little about editing, and a little about a few application programs.

With the appearance of the inexpensive microcomputer, the protective buffer has crumbled to the ground. The microcomputer user is operator, systems engineer, hardware engineer, program librarian, and, often, programmer. In the interactive, single-user environment that characterizes microcomputers, the user is on his own. And he is in trouble. As use of interactive computers, microcomputers, and networks expands, and as office automation proceeds, even more computer-naive persons will attempt to interact with computers. These potential computer users are unique in terms of training, motivation to use a computer system, and diversity.

Lack of Training

Many people trying to use computers today in business, education, and so on, have had no formal computer training. Thus far, most serious microcomputer users have come pretrained with experience on other computer systems that transfers to the new system. There are a few new users who are willing and able to pick up the necessary knowledge on their own, but many totally naive users probably give up or, at the least, never realize the full potential of their systems (e.g., Nickerson, 1981).

One view, often held wistfully by computer manufacturers and programmers, is that the general population will become more computer literate, and so the problem will vanish. Presumably, widespread computer literacy will come about because of (1) exposure of youth to computer games and (2) the introduction of computer courses into elementary and high schools. The evidence so far (e.g., Minnesota Educational Computing Consortium, Note 1) suggests that universal computer literacy may be difficult to achieve. In addition, it is not clear what is learned from computer games except perceptual-motor skills. For some time in the future, it seems likely that users will not be familiar with computers and will tend to avoid them.

Discretionary Use

Another important characteristic of nonprofessional computer users is that they are discretionary users. After paying thousands, even millions of dollars for a system, owners of large computers see that the computer will be used regardless of how difficult it is to use, even if specialists have to be hired to use it. Purchasers of inexpensive microcomputers are discretionary users. If they do not like the system or cannot use it, they have alternative ways to solve their problems.

Heterogeneity

Contemporary computer users can be characterized by extreme heterogeneity in terms of background and in terms of diversity of application (Smith, Note 2). A word processing program may be used by anyone from a grade school student to a PhD. A spread sheet, a scheduling program, or electronic mail is as likely to be used by a housewife as by a business executive. Heterogeneity of users will increase as more application programs become available.

Because of these user characteristics, major changes will have to be made to hardware and software if computers are to be successfully introduced into homes, schools, and offices of new users on a large scale. These changes are being made at the human-computer interface. The people finding out what changes to make are working in the realm of human factors in human-computer communication, and the primary goal, to use current buzz words, is to make systems "user friendly."

USABILITY

It is almost impossible to find an advertisement for a microcomputer system that does not claim that the system is "user friendly." Reading promotional literature might lead one to conclude that all the human-factors problems have been solved. Unfortunately, the promotional claims about user friendliness are rarely justified. This paper attempts to explain what a user-friendly system really is. For a slightly different approach to the definition of usability, see Borman and Karr (1982) and Wasserman (Note 3). Since human-factors researchers are trying to attain usability, a description of usability will provide an introduction to the realm of interest for the human-factors specialist.

It may be helpful to describe some characteristics that many people erroneously believe will make a system user friendly. Some programs are chatty. They are polite; they say "please" and "thank you." When you do something that is correct, they may use operant techniques to reinforce correct behavior (e.g., Dwyer, 1981). They say "hello" and "bye." They may even use stored system accounting information to refer to you by your first name. For a while our university timesharing system occasionally slipped in an "eh?," as many Canadians do in conversational speech.

None of these gimmicks contributes to the usability of a system. The term "user friendly" is rapidly falling into disrepute because it is misused so often. In fact, the term "usability" is preferred over "user friendliness" in an attempt to circumvent the unfortunate connotations of "friendliness." If these techniques do not add to usability, what does characterize a usable system?

Good Work-Station Design

The physical working environment, the work station, can dramatically influence usability. Work-station design is concerned with problems such as improving layout in terms of access to work-station components, keyboard layout and design, display legibility, glare (gray vs. green vs. amber displays), lighting, fatigue and stress, noise, and safety. Most of these issues, in contrast to those discussed later, are concerned with the physical aspect of the computer interface. Although work-station design principles have been "borrowed" from noncomputer studies (e.g., Galitz, 1980; Woodson, 1981), the widespread introduction of video display terminals has resulted in another round of research on work-station issues. The results of research on computer work stations will also be applicable to work-station problems in noncomputer areas.

Easy to Learn and Use

A usable system must be easy to learn and easy to use. Users must be able to concentrate on the details of the application rather than on details of using the computer. The computer is a means to an end and, in the extreme, should be transparent to the user. To give an oversimplifed example, modern microwave ovens are used by people who often do not realize that a computer is controlling the oven. Unfortunately, ease of learning and ease of use are often difficult to achieve in a complex application. The challenge to the human-factors specialist is to make complex applications easy to learn and use.

When the system cannot be designed to suit the needs of the of the user, the user must adapt to the system. System designers often erroneously assume that it is easy for users to learn a new system using reference manuals written for other purposes or tutorials written without an understanding of either the learning process or the user. Human-factors specialists recognize the need for training and the need for understanding the cognitive processes involved in the transition of users from novices to experts (e.g., DuBoulay, O'Shea, & Mark, 1981; Mayer, 1981).

Good Documentation

A usable system must have good documentation, either in manual form and/or, perhaps, on-line (Tombaugh & Dillon, Note 4). Any desired information must be easily accessible in the documentation. Usable system documentation will typically include three distinct types of information: a technical reference manual, a user's guide, and a system to provide quick answers to routine questions (e.g., an on-line help system) (e.g., Relles, Sondheimer, & Ingargiola, 1982). An on-line help system in which the user cannot easily find information (the usual case) is of no help at all.

System Reliability

A usable system must be "bullet proof"; that is, it must be impossible for the user to "bomb" or "crash" the system. No matter what the user does, the system must recover and maintain its integrity. System failures, whether caused by the user or not, are unacceptable. Many beginning users are afraid to use a system because they are afraid they might do something to harm the system.

Error Handling

A usable system will incorporate good error prevention, detection, reporting, and recovery. The system must minimize user errors by good software design; that is, the system must make it difficult for the user to go wrong. Nevertheless, users will make errors, so a usable system will also detect when errors occur and report them in a helpful form that can be understood. The usable system allows, even helps, the user to recover from the error. An exit from an application program to the operating system for a new start is a drastic measure to be avoided. Unfortunately, it is common for many programs (e.g., statistical packages) to return to the operating system when errors are encountered that the programmer considers fatal. The additional work devoted to error prevention, detection, reporting, and recovery goes a long way toward making a system usable.

Appropriate I/O

A usable system will have a software and hardware I/O interface between the user and computer that is appropriate for the task and the user. In the early days of computing (and even in some present-day computer systems), input was by means of a deck of punched cards and output was a listing on a line printer. It is ironic that I/O systems developed in the 19th century have dominated the interface between users and multimillion dollar, leading-edge technology computers. Currently, the integrated CRT terminal and keyboard constitute the most common user-computer interface. Most computer systems and application programs assume everyone has a terminal. The terminal is used as the only I/O device, whether appropriate or not. The need for a variety of I/O devices, even for different individuals performing the same task, has been documented by Hiltz and Turoff (1982). Input devices that may be appropriate for a particular application include keyboards, buttons and switches, joysticks, digitizers, lightpens, mouse, and voice recognition units. Yet, with the exception of joysticks used in games, it is rare to see any input device except a keyboard. Output devices that must be considered in addition to the CRT display are flat screen plasma displays, high- and low-resolution color and monochrome graphics, touch-sensitive screens and panels, lights, tones, and synthetic speech.

In addition to the appropriate I/O devices, the type of dialogue between user and computer must also be appropriate for the application and the user (e.g., Cuff, 1980; Martin, 1973; Peterson, 1978; Shackel, 1980; Ramsey & Atwood, Note 5; Savage, Habinek, & Barnhart, Note 6). Command lines (with their burden on human memory) that the user must enter at the correct time in the correct form, question-and-answer techniques under computer or user control, form filling, and menu selection can all add to or detract from the usability of a system, depending on the application and the user. In view of the difficulty of satisfying the needs of all users with a single dialogue type, consideration should be given to the provision of multiple dialogue types for the same application (e.g., Hiltz & Turoff, 1982).

Human-factors research has tended to concentrate on the I/O devices and techniques because they provide the most direct paths between user and computer. Recently, program design, which considers the program functions independently of I/O devices and techniques, has been viewed as an integral part of interface design (e.g., Smith, Note 2).

System Response Speed

A usable system is capable of responding fast enough. Currently, speed limitations are determined by the speed of the computer, the complexity of computations, the number of users on a system, and the capabilities of I/O devices. Ellis, Dillon, and Tombaugh (Note 7) report research on the speed issue.

Cognitive Organization

The system must be conceptually organized in a comprehensible form that corresponds to the cognitive abilities of the users for whom the application is intended (e.g., Martin, 1973). For example, most computer systems are designed around a hierarchical organization that all naive users and many experienced users have difficulty conceptualizing. The operating system is at the root of the organization tree. From the operating system, the user can move along a number of branches (e.g., to edit, compile a program, run a business package, etc.). These, in turn, may branch out to other programs or subprograms. When one of these applications is complete, the user moves up a branch to the operating system before going down another branch to perform some other operation. Beginners invariably have difficulty figuring out "where they are" in the system. Systems are available now that place the user in BASIC or in the word processor when the system is turned on. For these systems, conceptually, the operating system is embedded in the application program, rather than vice versa. In addition, issues such as file organization, automatic creation of backup files, ability to retrieve files that have been erroneously deleted, and organization of functions within a program may appear to be buried deep within the system. Yet such organizational factors can drastically influence usability. To be usable, the organization must be appropriate for the user, and the organization must be apparent to the user. Savage et al. (Note 6) demonstrated the difficulty users have with a menu organized in terms of general job classifications (e.g., system console jobs) and the improvement that results when the menu is organized in terms of specific tasks to be performed. If the menu cannot be organized in terms of job titles, it would be interesting to see if the system organization itself is at fault.

CONCLUSIONS

Human factors in computer research is a challenging and rewarding field. It is challenging because of the difficulty of the many problems the field addresses. It is rewarding because the researcher is working on improvements that can have a real impact on people. A side benefit, some would say the main benefit, is that human-factors research is advancing our understanding of human behavior.

REFERENCE NOTES

1. Minnesota Educational Computing Consortium. *A study of computer use and literacy in science education, Final report 1978-1980.* 2520 Broadway Drive, St. Paul, Minn: Author, 1980.

2. Smith, S. L. *User-system interface design for computer-based information systems* (Tech. Rep. ESD-TR-82-132; also MTR-8464). Boston: Mitre Corporation, 1982. (Prepared for Technical Operations and Product Assurance, Electronic Systems Division, Air Force Systems Command, United States Air Force.)

3. Wasserman, A. I. *User software engineering and the design of interactive systems.* Paper presented at the IEEE Conference on Software Engineering, San Diego, Calif., March 1981.

4. Tombaugh, J. W., & Dillon, R. F. *Recommendations concerning human factors guidelines for designing office system manuals.* Unpublished report, Carleton University, Behavioural Research and Evaluation Group, Office of Communication System Program, Department of Communications, 1982.

5. Ramsey, H. R., & Atwood, M. E. *Human factors in computer systems: A review of the literature* (Tech. Rep. SAI-79-111-DEN). Englewood, Col: Science Applications, 1979.

6. Savage, R., Habinek, J., & Barnhart, T. *The design, simulation, and evaluation of a menu driven user interface.* Paper presented at the Human Factors in Computer Systems Conference, Gaithersburg, Md., March 1982.

7. Ellis, J., Dillon, R. F., & Tombaugh, J. W. *The effect of system response times on a word processing task.* Poster presented at the annual meeting of the Eastern Psychological Association, New York, April 1981.

REFERENCES

Borman, L., & Karr, R. Efficient for computers or easy for users? Evaluating the "friendliness" of a timesharing system. *SIGSOC Bulletin*, 1982, 13, 31-34.

Cuff, R. N. On casual users. *International Journal of Man-Machine Studies*, 1980, 12, 163-187.

DuBoulay, B., O'Shea, T., & Mark, J. The black box inside the glass box: Presenting computer concepts to novices. *International Journal of Man-Machine Studies*, 1981, 14, 237-249.

Dwyer, B. A user-friendly algorithm. *Communications of the ACM*, 1981, 24, 556-561.

Galitz, W. O. *Human factors in office automation.* Atlanta: Life Office Management Association, 1980.

Hiltz, S. R., & Turoff, M. Human diversity and choice of interface: A design challenge. *SIGSOC Bulletin*, 1982, **13**, 125-130.

Martin, J. *Design of man-computer dialogues*. Englewood Cliffs, N.J: Prentice-Hall, 1973.

Mayer, R. E. The psychology of how novices learn computer programming. *Computing Surveys*, 1981, **13**, 121-141.

Moran, T. P. An applied psychology of the user. *Computing Surveys*, 1981, **13**, 1-11.

Nickerson, R. S. Why interactive computer systems are sometimes not used by people who might benefit from them. *International Journal of Man-Machine Studies*, 1981, **15**, 469-483.

Peterson, D. E. Screen design guidelines. *Small Systems World*, 1978, **8**, 19-21.

Relles, N., Sondheimer, N. K., & Ingargiola, G. P. Recent advances in user assistance. *SIGSOC Bulletin*, 1982, **13**, 1-5.

Schneiderman, B. *Software psychology: Human factors in computer and information systems*. Cambridge, Mass: Winthrop, 1980.

Shackel, B. Dialogues and language—Can computer ergonomics help? *Ergonomics*, 1980, **23**, 857-880.

Woodson, W. E. *Human factors design handbook*. New York: McGraw-Hill, 1981.