Seven plus or minus two central issues in human-computer interaction

Ben Shneiderman

Human-Computer Interaction Laboratory, and
Department of Computer Science
University of Maryland
College Park, MD 20742

ABSTRACT: This paper offers seven issues and specific challenges for researchers and developers of human-computer interaction. These issues are: interaction styles, input techniques, output organization, response time, error handling, individual differences, explanatory and predictive theories.

INTRODUCTION

One sign of the maturity of a scientific discipline is research community agreement on the central issues. This paper is meant to stimulate discussion of the central issues in human-computer interaction and thereby draw researchers together to engage in heated debates, vigorous research, and creative theories. The goal is to build elegant interactive systems that elicit enthusiasm, convey clarity, promote predictability, and build competence and confidence, while ensuring short learning times, rapid performance of tasks, low error rates, and ease of retention over time.

Successful research and development in the area of human-computer interaction will yield effective electronic mail systems, appealing word processors, attractive information retrieval services, error-free manufacturing systems, safe air traffic control, rapid hotel reservations systems, stimulating educational software, helpful expert systems, enjoyable art and music systems, and more.

The title was meant to be playful while conveying the intent of this paper. It also emphasizes the vital role understanding human cognitive and perceptual abilities plays in design of human-computer interactions. The following list of central issues is a personal one; it cannot cover every worthy issue. I invite thoughtful disagreements, extensions, amendments, and alterations. Certainly this list will need revision over time, but I hope it provides a timely moment of inspiration for students and fellow researchers.

Each issue invites a lifetime of effort and a book-length survey. The brief descriptions, references, and challenges are meant as a guide to those who seek research topics.

ISSUE 1. INTERACTION STYLES, INCLUDING DIRECT MANIPULATION: WHAT IS NATURAL?

There are five primary interaction styles:

- Menu selection: users are presented with a brief list of items using familiar terminology. They can conveniently choose the most appropriate item by pointing, typing, or speaking. The hope is that this structured approach enables users to accomplish their tasks with little or no training (Shneiderman, 1986a).

> **Challenge 1.1:** Develop an automated menu evaluation tool that takes a menu system description and predicts user performance measures: learning time, speed of use as a function of hours of usage, errors, and subjective satisfaction. The development of such a tool requires a deeper understanding of the cognitive processes in dealing with menus in a tree or other structure, and the ease-of-use of menu selection mechanisms (typing a letter or number, pointing with a mouse, pressing function keys, or moving a cursor).

- Form fill-in: when data entry is the primary goal, form fill-in offers a familiar context for entry of data with only modest training.

> **Challenge 1.2:** Measure user behavior with form fill-in systems and determine the importance of screen layout parameters such as grouping related items, use of highlighting techniques, alignment (left or right justification of labels and values), consistency across screens, and multi-screen vs. single screen layouts.

- Command languages: are attractive when sufficient learning time is available and frequent use is anticipated. Other conditions for appropriate use of command languages are that users are knowledgeable about the task domain and computer concepts, screen space is at a premium, response time and display rates are slow, numerous actions can be combined in many ways, and macro definition is desired (Shneiderman, 1986b).

> **Challenge 1.3:** Structure and meaningfulness have consistently been demonstrated to be helpful in

learning and using command languages, yet we have inadequate measures of these criteria (Carroll, 1982; Grudin & Barnard, 1984). Is it possible to develop validated measures of structure and meaningfulness? Are there guidelines for designers of command languages?

Even when suggested improvements are demonstrated to be effective in shortening time learning, speeding performance, or reducing error rates, there is tremendous resistance to change, especially by the most knowledgeable and experienced users.

Challenge 1.4: Can a commonly used command language be measurably improved and can users be won over to the revised system? What helps old dogs to learn new tricks?

- Natural language: interaction by natural language dialog is seen as the "ultimately desirable" style by many researchers. However, the lack of guidance or context for novices, the potential for more typing or speaking, and the complexity and uncertainty of "clarification dialog" raise doubts about the merits of this style when compared with menu selection or a precise concise command language (Shneiderman, 1980). Natural language interaction (NLI) may be most suitable when users are knowledgeable about a task domain whose horizon is limited and where intermittent use inhibits command language training. Wishful thinking cannot replace scientific evaluation of the conditions under which NLI is advantageous.

Natural language parsing techniques are applicable to information search in large textual databases. Natural language output generated from structured data is used in medical, psychological, and business applications. These are both important techniques, but they are not NLI. A clearer definition of what is meant by natural language interaction would help to focus research efforts; for example, predicate calculus or musical notation are quite natural to some people, but they are not the basis of NLI.

We might ask: How natural is natural? What are the criteria for success of a natural language system? Is an NLI system successful if users can get 95% recognition rates with vocabularies of 4000 words after an hour's training and four hours of usage?

Challenge 1.5: Comparisons with other interaction styles have not yet demonstrated an advantage for NLI (Hauptmann & Green, 1983; Jarke et al., 1985). Design software for a a specific user community and set of tasks and conduct empirical tests in competition with other interaction styles to demonstrate the merits of NLI. This may sharpen understanding of the conditions under which NLI is useful.

- Direct manipulation: users are often attracted to systems which offer a visual representation of the task-domain objects and actions. The display taps the users' knowledge and analogical reasoning skills. The actions for accomplishing tasks require selection by pointing instead of typing and are rapid, incremental, and reversible (Shneiderman, 1983a). Examples include display editors, video games, air traffic control, and spreadsheets.

The term direct manipulation is descriptive, but imprecise. Empirical studies are necessary to sharpen the definition, identify the components, measure the benefits, determine appropriate applications and users, and refine design guidelines.

Challenge 1.6: Design a direct manipulation system for a common command language and evaluate the relative merits of each. The DMDOS (Direct Manipulation Disk Operating System) package attempts to replace the commands in MS-DOS with cursor motions and button clicks (Iseki & Shneiderman, 1985). While many observers feel that DMDOS is an improvement, controlled experiments and field tests are necessary.

Many examples of direct manipulation exist, but there are fewer examples of direct manipulation programming in which users can create new objects, or actions.

Challenge 1.7: Create a direct manipulation programming environment for creating direct manipulation systems.

ISSUE 2. INPUT TECHNIQUES: PUTTING INTENTION INTO ACTION

The primary input technique has been typing on a keyboard, but this is giving way to novel pointing devices. The profusion of novel input devices is overwhelming and designers are rapidly discovering imaginative ways to apply them appropriately. Naive comparisons between the mouse, trackball, touchscreen, joystick, graphics tablet, and lightpen are giving way to thoughtful analyses of when to employ a specific device and how to redesign the software to match the attributes of an input device (Foley et al., 1984). Those who argue about the overall superiority of one device have a limited and archaic view.

Challenge 2.1: Develop guidelines for the choice of one device over another for a specific task and user community. Demonstrate the efficacy of the guidelines in empirical tests. Show how the right device can speed performance or reduce error rates by an order of magnitude in a realistic situation.

Keyboard input is necessary in many applications, such as word processing. The contemporary QWERTY keyboard can be improved upon by physical redesign (Nakaseko et al., 1985) or by different letter layouts. The piano keyboard is an impressive input device that allows several simultaneous finger presses and is responsive to different pressures and durations. It seems that there is potential for higher interaction rates with novel devices. Task-specific input devices are the most attractive possibility; for example, consider designing a special keyboard for typing PASCAL programs or chemical equations.

Challenge 2.2: Design an improved keyboard for a specific task that provides sufficient benefits to entice users to convert.

Discrete word speech input offers advantages in special situations. Designers are moving ahead to discover these situations: hands occupied, eyes busy, mobility required, hostile environments, etc. Many opportunities exist for telephone-based speech interaction. Studies have not shown speech input to be advantageous when compared with keyboard for cursor motion and text editing commmands (Murray, et al., 1983; Morrison et al., 1984).

Challenge 2.3: Find applications in which discrete word speech input is a winner. Document its advantages over other input devices.

ISSUE 3. OUTPUT ORGANIZATION: CREATING MEANINGFUL PATTERNS

Graphic designers speak about visual literacy, but current screen designs are often chaotic (Marcus, 1983). Guidelines abound (Brown, 1986; Smith, 1984; Galitz, 1981) and improvement is apparent, but tools are necessary to aid designers in developing and evaluating screen layouts.

Questions abound:

- How should windows be used? The current crop of window managers focuses on the computer-related issues of window placement, movement, reshaping, overlap, and deletion. The second generation will certainly deal with the automatic placement and sizing of windows. For many applications, window actions should be automatic results of user activity in the task domain; for example, when a programmer places the cursor on a variable name, the declaration should automatically appear in a nearby window whose size just matches the declaration.

- How may the possibilities of highlighting be best employed? Older screens with all capital letters are giving way to flexible displays with multiple fonts, boldfacing, italics, varying font sizes, underscoring, reverse video, blinking, and color.

- When is color helpful? Color coding can speed recognition, but it can also lead to the confusion of lively but unreadable displays. Color use should be conservative and supportive of the user's tasks (Robertson, 1980; Durrett and Trezona, 1982).

- How do designers ensure consistency? In spite of the best intentions of designers and programmers, screen layouts, abbreviations, terminology, and highlighting are often inconsistent and misleading. Can there be a notation to describe screen layouts that leads to guaranteed consistency?

Challenge 3.1: Take contemporary screen designs and change them according to a set of explicit guidelines. Then demonstrate a fifty percent reduction in task performance times and error rates, and a statistically significant increase in subjective satisfaction.

Challenge 3.2: Develop an automated evaluator of screen layouts that produces a series of quality metrics (Tullis, 1984). This tool should be able to offer suggestions for improved designs.

ISSUE 4. RESPONSE TIME: ARE SPEED LIMITS HELPFUL?

In most situations, most users prefer faster response times and perform tasks more rapidly (Shneiderman, 1984a). Some designers claim remarkable speed-ups as the response time drops below one second, but these results are poorly documented (Thadhani, 1981; Lambert, 1984). The results of many researchers are consistent in demonstrating that users pick up the pace of the interaction, that is, there is a strong correlation between system response time and user think time. However, the change in work style as a function of response time has not been adequately measured. Grossberg et al. (1976) found that with slow response times users used fewer commands and made fewer errors, but this study used only four subjects on a specialized computational task.

Challenge 4.1: Collect data about the changing profile of actions as a function of response time. Do users make more or fewer errors as the response time shortens? There is evidence from from a business decision-making situation that the error curve is U-shaped with respect to response time (Barber & Lucas, 1983), but does this apply to other tasks? Is there support for the conjecture that the bottom of the U-curve approaches zero as the cognitive complexity declines? Do users issue fewer commands if the response time increases?

For years managers and designers have argued about the impact of response time variability. The evidence is mixed with many non-significant results and a few suggestions that variability has a mild negative impact on user performance and satisfaction. Unexpectedly long or short delays distract the user, but modest variability (within 20% of the mean) appears to have only minor impact.

Challenge 4.2: Study the impact of variability within a single session as well as the impact of changes over a week or longer. Detect the different profile of usage as a function of small or large variability. Do users who experience large variability become more frustrated? Do they avoid certain commands that they fear will generate a particularly long response time?

Anecdotal evidence suggests that as users pick up the pace of the system and work more quickly, they may make more errors, become more frustrated, and have elevated blood pressure or other signs of anxious behavior (Brod, 1984). Turner (1984) reports that after a substantial reduction in response time, users experience symptoms of job burnout.

Challenge 4.3: Study users in professional settings to determine if faster response time leads to increased error rates, blood pressure, anxiety, or dissatisfaction.

ISSUE 5. ERROR HANDLING: PREVENTING USER ERRORS

The frustration of using computers is apparent in the results of user surveys and in satiric cartoons from daily newspapers. Violent messages such as FATAL ERROR, RUN ABORTED, may have been acceptable in the rough-and-tumble pioneering days of computing, but by now novice and expert users expect more specific and constructive guidance. Instead of ILLEGAL ENTRY users expect to see MONTHS RANGE FROM 1 TO 12. Instead of ERROR 637-02 WRITE FAILURE AT HEX ADDRESS 0A42E users expect DISK FULL: MAKE FREE SPACE OR USE ANOTHER DISK. Such changes not only raise user satisfaction and speed error correction, but they result in fewer errors because the

user learns more about the system (Shneiderman, 1983b; Dean, 1982). Fortunately, there is widespread sympathy for this issue and changes are apparent in contemporary systems (Isa et al., 1983). Designers are putting error messages in a single file to make them available for evaluation, to enable review for consistent terminology and style, to allow easy revision, and to permit monitoring the frequency of occurrence of each error. When the ten most frequent errors are identified efforts can be focussed on revising manuals, improving training, rewriting messages, or altering commands.

Improvements in error messages are easy to make and provide measurable benefits to users. However, bigger gains can be achieved by preventing errors. Just as automobile transmissions prevent the driver from going into reverse while the car is moving forward, future systems will prevent many errors from occurring. Video games already provide rich functionality while avoiding the need for error messages, and designers are discovering the ways to prevent errors in office automation, educational, and commercial applications (Shneiderman, 1984b). Syntax-directed editors offer programming language grammar templates that make it impossible to enter syntactically incorrect programs. Early versions may have been clumsy, but techniques are improving daily. The old-fashioned word processors with hundreds of esoteric format commands are giving way to direct manipulation display editors where what you see is what you get (WYSIWYG). Command languages with complex and confusing syntax are yielding to pull-down menus. Typing awkward strings of text is being replaced with selection by pointing.

> **Challenge 5.1:** Identify the strategies by which the number of error conditions and messages can be reduced (Norman, 1983). Apply them to a widely used office automation system to cut the rate of errors by an order of magnitude. Apply them to a life critical system such as air traffic or nuclear reactor control to cut the rate of errors by a factor of two.

ISSUE 6. INDIVIDUAL DIFFERENCES: BEYOND EGO-CENTRISM

In the early days of computer systems, programmers developed applications for themselves and their colleagues with a similar background and orientation. Now, designers produce interactive systems for users with very different skills from their own. Therefore, introspection is no longer a sufficient guide to designers. Testing with actual users becomes necessary. Pilot, prototype, and acceptance tests often produce surprising results because of the vast and difficult to predict differences among user communities.

Sophisticated designers recognize that detailed user profiles can be a tremendous asset, but researchers, designers, and managers are still groping for a suitable survey or test instrument to discriminate among users. Still more fuzzy are the design guidelines to support individuals with differing:

- gender
- age
- education
- ethnic background
- cultural heritage
- linguistic background
- cognitive styles
- learning styles
- personality styles.

Differences between men and women emerge in studies of educational (Fulton, 1985) and video games. Males are attracted to the more violent shooting games, while the women prefer Pacman. One female observer of Pacman noted the amusing chewing action and soundtrack and labeled it "oral agressive." Another women appreciated the pursuit of closure and completion in cleaning the screen of dots. Are these beliefs shared by others? Could designers develop strategies for determining the metaphorical preferences of specific user communities? Is it true that women are more interested in relationships with people, while men are more object oriented? Understanding gender differences and preferences has tremendous commercial potential.

> **Challenge 6.1:** Create a video game that is more attractive to women than to men.

The computer is potentially a great gift to people over 65 who might seek intellectual stimulation, communication with others by electronic mail and bulletin boards, information resources, and creative opportunities that are not physically demanding. However, very few applications have been developed for older people.

> **Challenge 6.2:** Create a software package that is attractive to people who are over 65 years old.

The Myers-Briggs Type Indicator is an increasingly popular test that helps identify individual preferences along four dimensions (Myers & Myers, 1980):

- extroversion vs. introversion
- sensing vs. intuition
- thinking vs. feeling
- judgment vs. perception

Correlations of personality type with occupation have been extensively studied and evidence of linkage with computer usage is emerging. Personality mismatches between designers and users are being identified as a source of information system failures (Levitan & Willis, 1985).

> **Challenge 6.3:** Find correlations between personality types and design guidelines; for example, do intuitive types perform better with shallow or deep menu structures? Do intuitive types prefer pointing devices more than sensing types? Do perceptive types prefer overlapping menus while judging types prefer tiled windows?

A key difference among users is their knowledge and experience with computer systems. Users who have overcome their anxiety about using computers are likely to have more positive expectations and a framework for acquiring knowledge. The experienced user can rapidly acquire many details by recognizing similarities with previously learned systems. As users gain experience with an interactive system they work more rapidly and may explore an increasing subset of the features. Expert frequent users expect rapid performance, need less informative feedback, can work with denser, more coded displays, create larger operation chunks, and seek to redefine the commands in the system to more closely suit their task needs.

> **Challenge 6.4:** Follow the progress of users from novice to expert, measuring user think times, capturing error rates, and recording profiles of command usage over time.

Challenge 6.5: Develop strategies for supporting multiple levels of proficiency in one system. Candidates include level-structured training, user control of screen display density, user control of system response time and display rate, hidden commands, and user definition of macros.

ISSUE 7. EXPLANATORY AND PREDICTIVE THEORIES: PRACTICAL PHILOSOPHY AND GENERATIVE GUIDELINES

No issue is more important for the long-term health of human-computer interaction research than the development of a sturdy theoretical foundation. Theories form the basis for research, teaching, and constructive guidelines for developers. No single theory will encompass all topics. Grand theories will help shape thinking and smaller focussed theories will be predictive over narrow domains.

Several researchers have proposed multi-level theories to guide designers, researchers, authors of instructional manuals, trainers, and users (Foley & Van Dam, 1982; Norman, 1984). The syntactic/semantic model, originally developed to describe programmer behavior (Shneiderman & Mayer, 1979) has been refined to describe the knowledge necessary for human-computer interaction (Shneiderman, 1986b).

Users use computer systems because they have a task to accomplish, for example, writing a scientific paper, managing a stock portfolio, or controlling air traffic. They must have knowledge of the semantics of the task domain: the objects and the actions (Figure 1). For a stock portfolio the objects include the stocks, transaction records, graphs of price over time, etc. The actions include buying and selling stocks, posting information, tallying profits, printing summaries, etc. In addition, the users must acquire knowledge about the semantics of the computer domain. The objects may include files, directories, disk space, or printers. The actions include opening or closing a file, copying and printing results, permissible computations, etc. Semantic knowledge is meaningfully acquired by example, analogy, and explanation. Semantic knowledge can be related to familiar concepts and tends to be durable in memory.

Finally, the users must acquire the device-dependent syntactic details such as which key to press to delete a character, which icon to select to close a file, and which command string to type to print a graph. Since syntactic knowledge is often arbitrary it must be rote memorized and is easily forgotten unless frequently rehearsed. The syntactic knowledge in the stock portfolio example might be the GRP and SAV commands for drawing a graph and saving the portfolio on disk.

The syntactic/semantic model and the other multi-level models have a lot in common. These models are descriptive or explanatory, providing guidance for designers or researchers. They are a framework for analyzing system, rather than a precise predictive models.

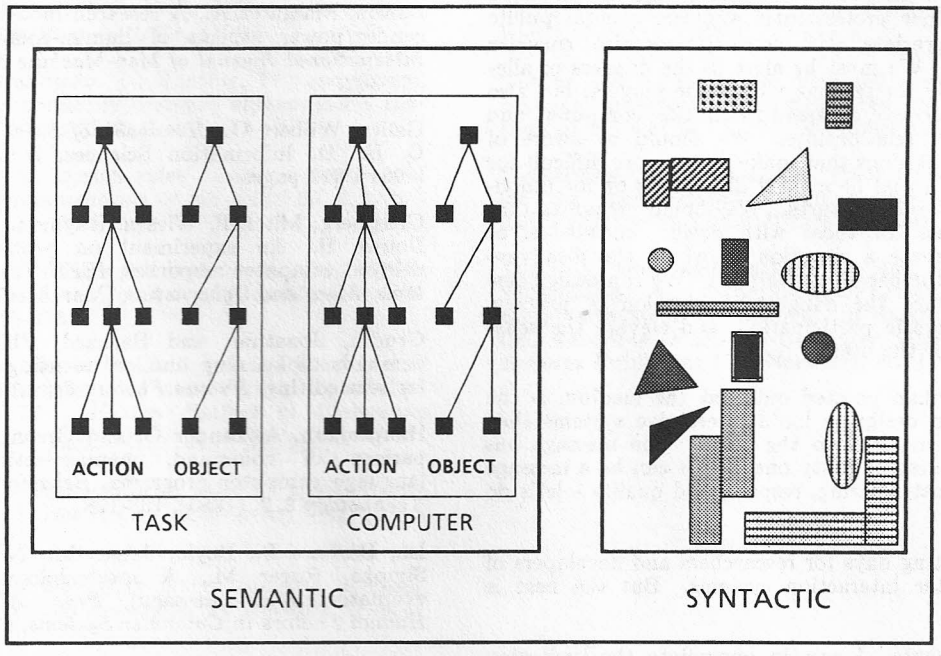

Figure 1. A representation of knowledge in long-term memory required for interactive systems users

Challenge 7.1: Starting with a multi-level model, develop a design process and notation that records the specifications for the system. Then use this notation to generate the actual system (Jacob, 1985; Wasserman, 1985; Yunten & Hartson, 1985).

Challenge 7.2: For one application domain and user community, develop a predictive theory to determine learning time, speed of performance, rate of errors, subjective satisfaction, or retention over time for alternative designs before they are built (Polson & Kieras, 1985; Card, Moran, and Newell, 1983).

CONCLUSION

These seven issues and multiple challenges are just a beginning. There are many other issues that call out for attention, including:

1) Training manuals, reference manuals, online help, and online tutorials.

2) Interface evaluation techniques: informal observation, thinking aloud strategies, controlled experiments, written or online questionnaires.

3) Specification methods and notations.

4) Dialog Management Systems, also called User Interface Management Systems.

With maturity comes responsibility. As researchers and developers we must complement technical excellence with a genuine concern for the impact that interactive systems have on the lives of individuals and the direction of society. Other professionals and the general public will most appreciate our eforts if we also consider broader issues. We must be alert to the dangers of alienation, not only for teenage videogame players, but also adults who become obsessed with the computer and ignore personal relationships. We should be aware of subtle design decisions that make usage more difficult for people with physical or mental disabilities or for individuals from different cultures. We should recognize that computer access for those with power, knowledge, or wealth may create a situation in which the disadvantaged become further disadvantaged. We can design systems that reduce the dangers of invasion of privacy, increase democratic participation, and elevate the sense of self-worth for the user.

Marshall McLuhan pointed out that the medium is the message. When designers build interactive systems they are sending a message to the users. The message has often been a harsh or nasty one, but it can be a message that communicates caring, respect, and quality - let's do it.

These are exciting days for researchers and developers of human-computer interaction systems. But the best is yet to come!

Acknowledgements: I greatly appreciate the invitation of the program committee to make this presentation. Marilyn Mantei and Peter Orbeton were especially sympathetic and helpful. Janis Morariu, Susan Flynn, and Suzanne Stevenson offered helpful comments on drafts of this paper.

REFERENCES

Barber, Raymond E. and Lucas, H. C., System response time, operator productivity, and job satisfaction, *Communications of the ACM 26*, 11, (November 1983), 972-986.

Brod, Craig, *Technostress: The Human Cost of the Computer Revolution*, Addison-Wesley Publishing Company, Reading, MA, (1984).

Brown, C. Marlin, *Human-Computer Interface Design Guidelines*, Ablex Publishing Company, Norwood, NJ, (1986).

Card, S. K., Moran, T. P., and Newell, A., *The Psychology of Human-Computer Interaction*, LEA, Hillsdale, NJ, (1983).

Carroll, J. M., Learning, using and designing command paradigms, *Human Learning 1*, 1, (1982), 31-62.

Dean, M., How a computer should talk to people, *IBM Systems Journal 21*, 4, (1982), 424-453.

Durrett, John and Trezona, Judi, How to use color displays effectively, *BYTE*, (April 1982), 50-53.

Foley, James D., Wallace, Victor L., and Chan, Peggy, The human factors of computer graphics interaction techniques, *IEEE Computer Graphics & Applications*, (November 1984), 13-48.

Foley, James D. and Van Dam, Andries, *Fundamentals of Interactive Computer Graphics*, Addison-Wesley Publishing Co., Reading, MA, (1982).

Fulton, Margaret A., A research model for studying the gender/power aspects of human-computer interaction, *International Journal of Man-Machine Studies 23*, (1985), 369-382.

Galitz, Wilbert O., *Handbook of Screen Format Design*, Q. E. D. Information Sciences, Inc., Wellesley, MA, (1981), 212 pages.

Grossberg, Mitchell, Wiesen, Raymond A., and Yntema, Douwe B., An experiment on problem solving with delayed computer responses, *IEEE Transactions on Systems, Man, and Cybernetics*, (March 1976), 219-222.

Grudin, Jonathan and Barnard, Phil, The cognitive demands of learning and representing command names for text editing, *Human Factors 26*, 4, (1984), 407-422.

Hauptmann, Alexander G. and Green, Bert F., A comparison of command, menu-selection and natural language computer programs, *Behavior and Information Technology 2*, 2, (1983), 163-178.

Isa, Barbara S., Boyle, James M., Neal, Alan S., and Simons, Roger M., A methodology for objectively evaluating error messages, *Proc. ACM SIGCHI '83 Human Factors in Computer Systems*, (1983), 68-71.

Iseki, Osamu and Shneiderman, Ben, Applying direct manipulation concepts: Direct Manipulation Disk Operating System (DMDOS), University of Maryland Department of Computer Science Technical Report, (December 1985).

Jacob, Robert J. K., An executable specification technique for describing human-computer interaction, In Hartson, H. Rex, Editor, *Advances in Human-Computer Interaction: Volume 1*, Ablex Publishing Co., Norwood, NJ, (1985), 211-242.

Jarke, Matthias, Turner, Jon A., Stohr, Edward A., Vassilliou, Yannis, White, Norman H., and Michielsen, Ken, A field evaluation of natural language for data retrieval, *IEEE Transactions on Software Engineering SE-11*, 1, (January 1985), 97-113.

Lambert, G. N., A comparative study of system response time on program developer productivity, *IBM Systems Journal 23*, 1, (1984), 36-43.

Levitan, K. B. and Willis, E. A., Barriers to practitioners' use of information technology: A discussion and results of a study, In Figley, C. R., Editor, *Computers and Family Therapy*, Haworth Press, New York, NY, (1985), 21-35.

Marcus, Aaron, Graphic design for computer graphics, *IEEE Computer Graphics & Applications*, (July 1983), 63-70.

Morrison, D. L., Green, T. R. G., Shaw, A. C., and Payne, S. J., Speech-controlled text editing: effects of input modality and of command structure, *International Journal of Man-Machine Studies 21*, 1, (1984), 49-83.

Murray, J. Thomas, Van Praag, John, and Gilfoil, David, Voice versus keyboard control of cursor motion, *Proc. Human Factors Society - 27th Annual Meeting*, (1983), 103.

Myers, I. B. and Myers, P. B., *Gifts Differing*, Consulting Psychologists Press, Palo Alto, CA, (1980).

Nakaseko, M., Grandjean, E., Hunting, W., and Gierer, R., Studies of ergonomically designed alphanumeric keyboards, *Human Factors 27*, 2, (1985), 175-187.

Norman, Donald A., Design rules based on analyses of human error, *Communications of the ACM 26*, 4, (April 1983), 254-258.

Norman, Donald A., Stages and levels in human-machine interaction, *International Journal of Man-Machine Studies 21*, (1984), 365-375.

Polson, P. G. and Kieras, D. E., A quantitative model of learning and performance of text editing knowledge, *Proc. ACM CHI'85 - Human Factors in Computing Systems*, ACM, New York, NY, (1985), 207-212.

Robertson, P. J., A guide to using color on alphanumeric displays, IBM Technical Report G320-6296, IBM, White Plains, NY, (1980).

Shneiderman, Ben, Designing menu selection systems, *Journal of the American Society for Information Science*, (1986a) (to appear).

Shneiderman, Ben, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Addison-Wesley Publishing Co., Reading, MA, (1986b) (to appear).

Shneiderman, Ben, Response time and display rate in human performance with computers, *ACM Computing Surveys 16*, 4, (September 1984a), 265-285.

Shneiderman, Ben, Correct, complete operations and other principles of interaction, In *Human-Computer Interaction*, Salvendy, G., Editor, Elsevier Science Publishers B. V. Amsterdam, The Netherlands, (1984b), 135-146.

Shneiderman, Ben, Direct manipulation: A step beyond programming languages, *IEEE Computer 16*, 8, (August 1983a), 57-69.

Shneiderman, Ben, System message design: Guidelines and experimental results, In Badre, A. and Shneiderman, B., *Directions in Human-Computer Interaction*, Ablex Publishing Co., Norwood, NJ, (1983b).

Shneiderman, Ben, *Software Psychology: Human Factors in Computer and Information Systems*, Little, Brown and Co., Boston, MA, (1980), 320 pages.

Shneiderman, Ben and Mayer, Richard, Syntactic/semantic interactions in programmer behavior: A model and experimental results, *International Journal of Computer and Information Sciences 8*, 3, (1979), 219-239. Reprinted in Curtis, Bill, Editor, *Human Factors in Software Development*, IEEE Computer Society EHO 185-9, (1981), 9-23.

Smith, Sid L. and Mosier, Jane N., Design guidelines for the user interface for computer-based information systems, The MITRE Corporation, Bedford, MA 01730, (September 1984), 448 pages.

Thadhani, A. J., Interactive user productivity, *IBM Systems Journal 20*, 4, (1981), 407-423.

Tullis, T. S., Predicting the usability of alphanumeric displays, Ph. D. Dissertation, Department of Psychology, Rice University, (1984), 172 pages.

Turner, Jon A., Computer mediated work: The interplay between technology and structured jobs, *Communications of the AACM 27*, 12, (DEcember 1984), 1210-1217.

Wasserman, A. I., Extending state transition diagrams for the specification of human-computer interaction, IEEE Transactions on Software Engineering, (1985).

Yunten, Tamer and Hartson, H. Rex, A SUPERvisory Methodology And Notation (SUPERMAN) for human-computer system development, In Hartson, H. Rex, Editor, *Advances in Human-Computer Interaction: Volume 1*, Ablex Publishing Co., Norwood, NJ, (1985), 243-281.