LEVEL ①

20000728017

AD A106194

Reproduced From
Best Available Copy

DTIC
ELECTE
OCT 28 1981
S D
E

DTIC FILE COPY

Virginia Polytechnic Institute and State University

Computer Science
Industrial Engineering and Operations Research
BLACKSBURG, VIRGINIA 24061

81 10 27 230

CSIE-81-2 September 1981

LEVEL

USER CONSIDERATIONS IN COMPUTER-BASED INFORMATION SYSTEMS

Compiled by

Beverly H. Williges

Robert C. Williges

DTIC ELECTE 1981

TECHNICAL REPORT

Prepared for
Engineering Psychology Programs, Office of Naval Research
ONR Contract Number N00014-[illegible]1-K-0143
Work Unit NR SRO-1[illegible]

Approved for Public Release; Distribution Unlimited

Reproduction in whole or in part is permitted
for any purpose of the United States Government

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: CSIE-81-2	2. GOVT ACCESSION NO.: AD-A106194
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): USER CONSIDERATIONS IN COMPUTER-BASED INFORMATION SYSTEMS	5. TYPE OF REPORT & PERIOD COVERED: Technical Report
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Beverly H. Williges, Robert C. Williges	8. CONTRACT OR GRANT NUMBER(s): N00014-81-K-0143
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Industrial Engineering & Operations Research, Virginia Polytechnic Institute & State University, Blacksburg, VA 24061	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: SRO-101
11. CONTROLLING OFFICE NAME AND ADDRESS: Office of Naval Research, Code 455, 800 North Quincy Street, Arlington, VA 22217	12. REPORT DATE: September 1981
	13. NUMBER OF PAGES: 84
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

human-computer systems, human-computer interaction, human-computer interface, human-computer dialogue, human-computer communications, form-filling, menu selection, input devices, command languages, computer aids

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The purpose of this report is to compile in one document the variety of user considerations relating to software design of computer-based information systems. Approximately 500 such considerations that currently exist in a variety of thirteen source documents are included. For organizational purposes, the compilation was divided into seven major parts including data organization, dialogue modes, user input devices, command language and command processing, feedback and error management, security and disaster prevention, and multiple user communication.

DD FORM 1 JAN 73 1473

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

4112

ACKNOWLEDGEMENTS

This research was supported by the Office of Naval Research under ONR Contract Number N00014-81-K-0143, and Work Unit NR SRO-101. The effort was supported by the Engineering Psychology Programs, Office of Naval Research, under the technical direction of Dr. John J. O'Hare. Detailed comments on a draft of this report were provided by various members of the TTCP UTP-4 Committee and Dr. Martin A. Tolcott of the Office of Naval Research who chaired that committee. The suggestions made by these individuals are gratefully acknowledged by the compilers of this report.

Accession For
NTIS GRA&I X
DTIC TAB
Unannounced
Justification
By
Distribution/
Availability Codes
Avail and/or
Dist Special
A

TABLE OF CONTENTS

SECTION I

SECTION II

SECTION I

INTRODUCTION

Purpose

To maximize performance, designers are becoming increasingly aware of the need to consider the human-computer interface in computer-based systems. In the past several years a number of technical reports and books have offered guidelines for the design of computer-based systems. This report is an attempt to bring together the variety of guidelines in the form of user considerations. The user considerations compiled are limited to those dealing directly with the human-computer dialogue primarily as it relates to software design. Information related to the design of computer hardware, including such topics as keyboard layout, system delays, and display quality assessment, was explicitly excluded. Likewise, no attempt was made to include user considerations related to workspace design for users of computer-based systems.

The purpose of this report was simply to compile into one document the various user considerations that currently exist in a variety of sources as an aid in structuring behavioral research to develop and evaluate empirically based guidelines. Because this compilation was not developed as a handbook for designers of human-computer interfaces, no evaluation of these user considerations is given nor is any indexing or cross-referencing provided. Obviously, the relevant user considerations for a designer vary from context to context, and the designer must determine which are appropriate for a particular human-computer interface. Additionally, in a few cases conflicting user considerations have been offered from different sources, and both have been included in this report for completeness. Where conflicting user considerations exist, each designer must determine which suggestions to adopt until behavioral research can resolve these conflicts and/or establish the appropriate context area for each.

In cases where similar user considerations were proposed by several authors, they have been combined into one guideline for this report. Because most of the source documents were not literature reviews, empirical support for the user considerations was not usually provided even in the few cases where such support may exist. Whenever other material was cited in the source documents to support a design guideline, the references were included in this compilation. The references cited vary and include both general discussions of basic human information processing capabilities and reports of specific empirical studies dealing with computer-based systems. Obviously, to represent the level of support for each of these user considerations accurately, a comprehensive literature review of empirical research dealing with human-computer interface problems is necessary.

Hopefully, the considerations have been presented in a manner which accurately represents the intent of the original authors. However, many of them have been rewritten for consistency within this document or shortened for brevity. Any misrepresentation of the original version is unintentional. Readers may wish to consult the thirteen source documents listed in Table 1 for clarification.

Fundamentals of Human-Computer Dialogue Design

In general, the designer of any system sets out to minimize equipment costs as well as personnel costs. However, these goals are often not compatible. Until recently the high cost and relatively limited

Table 1. Reports and Books Reviewed

---

(1) Brown, C. M., Burkleo, H. V., Mangelsdorf, J. E., Olsen, R. A., and Williams, A. R., Jr. Human factors engineering criteria for information processing systems. Sunnyvale, California: Lockheed, October 1980.

(2) Ehrenreich, S. L. Query languages: Design recommendations derived from the human factors literature. *Human Factors*, in press. (Also: Design recommendations for query languages (Tech. Rep. 484). Alexandria, Virginia: U. S. Army Research Institute, 1981.

(3) Engel, S. E. and Granda, R. E. Guidelines for man/display interfaces (Tech. Rep. 00.2720). Poughkeepsie, New York: IBM, December 1975.

(4) Gebhardt, F. and Stellmacher, I. Design criteria for documentation retrieval languages. *Journal of the American Society for Information Science*, 1978, *29*, 191-199.

(5) Hiltz, S. R. and Turoff, M. *The network nation*. Reading, Massachusetts: Addison-Wesley, 1978.

(6) Martin, J. *Design of man-computer dialogues*. Englewood Cliffs, New Jersey: Prentice-Hall, 1973.

(7) Miller, L. A. and Thomas, J. C., Jr. Behavioral issues in the use of interactive systems (Tech. Rep. RC 6326). Yorktown Heights, New York: IBM, December 1976.

(8) Newman, W. M. and Sproull, R. F. *Principles of interactive computer graphics*. New York: McGraw-Hill, 1979.

(9) Parrish, R. N., Gates, J. L., Munger, S. J., and Sidorsky, R. C. Development of design guidelines and criteria for user/operator transactions with battlefield automated systems. Volume IV: Provisional guidelines and criteria for the design of user/operator transactions (Draft Final Report, Phase I). Alexandria, Virginia: U.S. Army Research Institute, 1981.

(10) Pew, R. W. and Rollins, A. M. Dialog specification procedures (rev. ed.) (Rep. No. 3129). Cambridge, Massachusetts: Bolt Beranek and Newman, Inc., September 1975.

(11) Ramsey, H. R. and Atwood, M. E. Human factors in computer systems: A review of the literature (Tech. Rep. SAI-79-111-DEN). Englewood, Colorado: Science Applications, September 1979. (AD A075679)

(12) Shneiderman, B. *Software psychology: Human factors in computer and information systems*. Cambridge, Massachusetts: Winthrop, 1980.

(13) Smith, S. L. Man-machine interface (MMI) requirements definition and design user considerations: A progress report (Tech. Rep. ESD-TR-81-113). Bedford, Massachusetts: MITRE, February, 1981. (AD A096 705)

---

speed of computer systems has dictated a design tradeoff favoring the capabilities and limitations of the computer and not the human. With the rapid advances in computer technology and the subsequent cost reductions in computer hardware, designers can now effectively attempt to optimize the human aspect of the interface.

In terms of human performance, the system designer must work toward an acceptably low error rate and an acceptable cost in personnel time. In addition, user acceptance and satisfaction with the computer system seem to be critical to effective utilization.

It is quite likely that many basic psychological user considerations found to be fundamental to good system design in other applications will be equally important in the design of computer-based systems. Indeed, many of the user considerations proposed for computer-based systems appear to be nothing more than a restatement of basic human factors considerations as they specifically relate to systems involving computers. The general human factors principles that seem to be present in the specific human-computer dialogue design considerations reviewed include compatibility, brevity, flexibility, immediate feedback, and operator workload.

Compatibility. The principle of compatibility predicts high information transfer when the amount of information recoding necessary is minimal. Translated to the human-computer system this would suggest that the input required of the user should be compatible with the output of the computer and vice versa. Compatibility implications for human-computer dialogue can take several forms. The organization of data to be input should be compatible with the data organization of output. Both the input required of the user and the output of the system should be consistent across the display, module, program, and the information system. The choice of terminology, format, and system action should be consistent with user population stereotypes. The input required of the user should not be ambiguous, and the output of the computer should be clear and, therefore, useful. To minimize the information processing requirements of the user, information should be presented in a directly usable form. The need to translate, transpose, interpret, or refer to documentation should be minimized.

Brevity. Theories of human memory suggest the existence of some upper limit of information that can be received in a given period of time. The limit of short-term memory is generally accepted to be seven or eight items. When longer input is required, chunking should be used such that meaningful units of information are grouped together. To increase the number of bits of information that can be included in one input sequence, larger chunks each containing more information should be built. In computer-based dialogues this would suggest that both the input required of the user and the output of the system should be brief to minimize both the short-term memory load on the user and the probability of input errors by the user. In addition, user input and computer output should be grouped into meaningful chunks, whenever possible.

Flexibility. Individual differences among users necessitate system flexibility to insure optimum performance of all users. In many systems a decision must be made as to whether the system should be designed to accommodate the extreme individuals or the average individual. However, using the capabilities of the computer, one is often able to provide a flexible or adaptive system that suits all potential users equally. In computer-based dialogues this would suggest that both the input required of the user and the output provided by the system should vary for a particular user depending upon the user's expectations and capabilities.

Immediate feedback. A human-computer system should be closed-loop with information feedback to the human operator about the quality of performance and condition of the system. Without immediate feedback which is readily understandable, the user cannot make decisions regarding the necessity for corrective action and the form it should take. In computer-based systems, users should at all times be aware of where they are, what they have done, and whether or not it was successful. The user should be given every opportunity to correct errors.

Operat r workload. An assessment of potential operator workload should be one of the first tasks in the design of human-computer dialogues. Because the probability of human failure increases in overload situations, the overall goal should be to keep the workload of the user within acceptable limits. This includes consideration of the limited channel capacity of the human as well as defining the operator's task and extending it to display terminal requirements. If one assumes that the human operator is a single channel capacity device, information from various sources arrives and is queued until processing can occur. Data should be organized to minimize the scanning required of the user. Workload considerations in human-computer interactions have implications for determining the information density on display screens, providing redundant information in multiple channels, determining the appropriate size for a command language, etc.

HUMAN-COMPUTER DIALOGUE DESIGN CONSIDERATIONS

The dialogue design considerations summarized in Section II of this report deal primarily with human-computer dialogues in the form of alphanumeric information. Few guidelines dealing with graphic information exist in the source documents used and, therefore, few are offered in this compilation. Additionally, no effort was made to include user considerations dealing with "intelligent" systems that incorporate rule-based or other artificial intelligence techniques. Although it is quite desirable to design and develop computer systems that adapt to the skill level of the user, design considerations of this nature were not stressed in the documents included in this summary. Consequently, the following compilation is restricted primarily to alphanumeric information displays; the need clearly exists to extend and develop design considerations dealing with adaptive human-computer systems and computer display of graphic information.

For organizational purposes, the resulting compilation is divided into seven parts as shown in Table 2. These major parts include data organization, dialogue modes, user input devices, command languages and command processing, feedback and error management, security and disaster prevention, and multiple user communication.

Part 1 deals with data organization in terms of information coding, information density, information labeling, display screen layout, and appropriate formats for various types of data. Part 2 deals with user considerations that are specific to a particular human-computer dialogue mode. Dialogue modes included are form-filling, prompting or computer-initiated question-and-answer dialogues, menu selection, command languages, query languages, and restricted natural language for data base query. Part 3 deals with considerations concerning techniques and devices involved in user input of information to the computer. Part 4 deals with command languages and the processing of commands. It includes considerations dealing with command organization, command nomenclature, the use of defaults, editor orientation, user control of command processing, and command operation. Part 5 is concerned with feedback and error management. A broad range of topics are included dealing with system feedback, error recovery, user control of system feedback, help and documentation, and computer aiding. Part 6 covers aspects of computer security requirements that impact the human-computer interface. Finally, Part 7 deals with systems in which the computer must coordinate the input of multiple users. A major category in this area is on-line message systems where the messages of one user must be buffered to prevent interference with another user.

Each design consideration in Section II is succinctly stated under the various classification headings listed in Table 2. The listings of design guidelines under this organizational structure are not given in order of importance. The numbers or series of numbers appearing in parentheses after the statement of each design consideration refer to the reports, listed in Table 1, in which the various design considerations were found.

Some of the human-computer dialogues design considerations in Section II are marked with an asterisk. Although the compilers of this report did not evaluate the efficacy of the design guidelines, the resulting compilation was reviewed by the TTCP UTP-4 Human Factors in Command and Control Committee. Whenever a member of that committee took

Table 2. Classification Scheme for User Considerations

---

1. DATA ORGANIZATION

1.1 Information Coding
- 1.1.1 Color Codes
- 1.1.2 Shape Codes
- 1.1.3 Blinking Codes
- 1.1.4 Brightness Codes
- 1.1.5 Alphanumeric Codes

1.2 Information Density
1.3 Labeling
1.4 Format
- 1.4.1 Prompts
- 1.4.2 Tabular Data
- 1.4.3 Graphics
- 1.4.4 Textual Data
- 1.4.5 Numeric Data
- 1.4.6 Alphanumeric Data

1.5 Screen Layout

2. DIALOGUE MODES

2.0 Choice of Dialogue Mode
2.1 Form-Filling
- 2.1.1 Default Values
- 2.1.2 Auditory Feedback
- 2.1.3 Form Layout
- 2.1.4 Data Entry Procedures
- 2.1.5 Cursor Movement

2.2 Computer Prompting
2.3 Menu Selection
- 2.3.1 Order of Options
- 2.3.2 Selection Codes
  - 2.3.2.1 Letter Codes
  - 2.3.2.2 Number Codes
- 2.3.3 Menu Layout
- 2.3.4 Menu Content
- 2.3.5 Control Sequencing

2.4 Command Languages
2.5 Query Languages
2.6 Restricted Natural Language

3. USER INPUT DEVICES

3.0 Data Entry Procedures
3.1 Selection of Input Device
3.2 Keyboards
- 3.2.1 Special Function Keys
- 3.2.2 Cursor Control

3.3 Direct Pointing Controls
3.4 Continuous Controls
3.5 Graphics Tablets
3.6 Voice Analyzers

4. COMMAND LANGUAGES AND COMMAND PROCESSING

4.1 Command Organization
4.2 Command Nomenclature
  4.2.1 Abbreviations
  4.2.2 Argument Formats
  4.2.3 Separators/Terminators
4.3 Defaults
4.4 Editor Orientation
4.5 User Control
  4.5.1 Command Stacking
  4.5.2 Macros
  4.5.3 Immediate Commands
4.6 Command Operation
4.7 System Response Time
4.8 Special Commands

5. FEEDBACK AND ERROR MANAGEMENT

5.1 Feedback
  5.1.1 Status Messages
  5.1.2 Error Messages
  5.1.3 Hard Copy Output
5.2 Error Recovery
  5.2.1 Immediate User Correction
  5.2.2 User Correction Procedures
  5.2.3 Metering and Automatic Error Checks
  5.2.4 Automatic Correction
  5.2.5 Stacked Commands
5.3 User Control
5.4 Help and Documentation
  5.4.1 Off-Line Documentation
  5.4.2 On-Line Documentation
5.5 Computer Aids
  5.5.1 Debugging Aids
  5.5.2 Decision Aids

6. SECURITY AND DISASTER PREVENTION

6.1 Command Cancellation
6.2 Verification of Ambiguous or Destructive Actions
6.3 Sequence Control
6.4 System Failures

7. MULTIPLE USERS

7.1 Separating Messages/Inputs
7.2 Separating Work Areas
7.3 Communications Record

---

exception to a guideline, felt it needed further explanation, or indicated that it should be restricted to a specific set of conditions, the guideline was marked with an asterisk to note that it had been questioned by at least one member of the committee. These evaluations are soleiy those of the TTCP UTP-4 Committee and do not necessarily reflect the views of the compilers of this report.

SECTION II

1. DATA ORGANIZATION

The first set of user considerations in the design of human-computer dialogues deals with aspects of structuring information on the computer display in an interactive environment. The major topics of consideration deal with the methods of coding information, control of the amount or density of information displayed, the use of labeling to organize the information displayed, various techniques for formatting the display, and considerations for the overall layout of information fields on the display screen. For additional information on coding schemes applicable to human-computer systems, see Parrish, Gates, Munger, and Sidorsky (1981) and Barmack and Sinaiko (1966). Foley and Wallace (1974), Martin (1973), and Prince (1971) provide some additional information on the use of graphics in human-computer communication.

These design considerations are presented under the following subheadings.

- 1.1 Information Coding
  - 1.1.1 Color Codes
  - 1.1.2 Shape Codes
  - 1.1.3 Blinking Codes
  - 1.1.4 Brightness Codes
  - 1.1.5 Alphanumberic Codes
- 1.2 Information Density
- 1.3 Labeling
- 1.4 Format
  - 1.4.1 Prompts
  - 1.4.2 Tabular Data
  - 1.4.3 Graphics
  - 1.4.4 Textual Data
  - 1.4.5 Numeric Data
  - 1.4.6 Alphanumeric Data
- 1.5 Screen Layout

1.1 Information Coding

Information coding should be used to discriminate among different classes of items presented simultaneously on the display screen. (7)

Meaningful codes should be used when possible. Codes should be clear and consistent with the user's expectations. (1)

Highlighting should be used for critical information, unusual values, items to be changed, items that have been changed, high priority messages, the source of alarms, special function areas of the display, errors in entry, warnings of consequences of commands, and targets. (9)

If the type of information coding selected reduces legibility, is not distinct, or increases transmission time, it should not be used. (9)

1.1.1 Color Codes

Color coding should be used to highlight related data which are spread about the display. Color coding may be used to locate headings, out-of-tolerance data, newly entered data, data requiring immediate attention, etc. (1)

Color coding should be used for search tasks. (1,11)
References: Christ, 1975; Teichner, Christ, & Corso, 1977.

* Information should not be coded solely by color if the information will be accessed from monochromatic as well as color terminals and/or when printed versions will be made. Where both kinds of terminals may be in use, color must be limited to assisting users of the color terminals without sacrificing information to the users of the monochromatic displays. (1)

Color coding should allow for potential color-blindness or color weakness (approximately 8% of males). Red is most likely to be a problem. (1, 9)

Color should be used conservatively to avoid an appearance of clutter. (1)

Color coding should generally be limited to three hues; the maximum is ten. (11)
Reference: Grether & Baker, 1972.

A maximum of eleven color codes should be used. (3)
Reference: Barmack & Sinaiko, 1966.

When characters are formed by a combination of primary colors, color registration problems can occur, particularly near the corners and edges of the display. Color displays should be adjusted periodically to maintain proper registration of images. When the display is out of adjustment, characters formed by a combination of primary colors (pink, yellow, turquoise, and white) may appear as characters in each of the component primary colors. Registration problems do not apply to characters formed by a single primary color. (1)

In the selection of color codes, color meanings should be considered because the color itself may convey information. Example: red/danger, yellow/caution, and green/OK. (1)

* Headings may be color coded in white if the table is so complex that highlighting the headers will help the user. Data associated with alarms, undesirable states, or information requiring immediate attention should be presented in pink. Yellow may be used for highlighting related data which are distributed about the screen or for updates which should be noticed. User inputs should be color coded in turquoise (cyan) because it has good brightness and no particular associated meaning. (1)

* The principal color employed in display screens should be green because it provides good contrast with the background, is a primary color, and is consistent with the color generally displayed by monochromatic displays. (1)

* The blue and red usually used in color displays have low brightness and should be avoided. Characters should not be displayed in blue, though it may be used for shading areas in a graphic display. (1)

Each color code should be defined at the bottom of the data display. A color should be used for only one meaning. (1)

* The definition for a color should be displayed in the hue of the defined color. (1)

1.1.2 Shape Codes

Shape coding should be used in search and identification tasks. (1)
References: Christ, 1975; Grether & Baker, 1972.

A maximum of fifteen different shape codes should be used. (3)
Reference: Barmack & Sinaiko, 1966.

1.1.3 Blinking Codes

Blink coding should be used for alarms. (9)

Blink coding should be used for coding in target detection tasks particularly with high density displays. (11)

* Blink coding should not be used with long persistence phosphor displays. (9)

* The user should be able to turn off the blinking. (9)

* To avoid interference with reading performance, the blink rate should be such that the user can match his scan rate to the blink rate. (11)

* To attract attention to urgent items, blinking should be on a 2-3 Hz cycle with a minimum duration of 50 msec. (3)
NOTE: Another reference suggested that a blink rate of 3-4 Hz should be used. (11)
References: Smith & Goodwin, 1971a; Vartabedian, 1970.

* Although users can discriminate up to four different blink rates, blink coding should probably be restricted to a binary code (one flashing and one static). (3,11)
Reference: Barmack & Sinaiko, 1966.

1.1.4 Brightness Codes

When an operation is to be performed on a single item on a display, the item should be highlighted. (3)

On crowded displays, auxiliary codes such as dim labels and bright data should be used to distinguish the labels. (13)

The option(s) in a list selected by a user should be highlighted. (3)

* No more than 10% of the display should be highlighted at one time. (9)

No more than three levels of brightness coding should be used. (3, 9)

* Maximum contrast should be provided between those items highlighted and those not. This seems to be achieved best with text by reversing the image (dark on a light background, for example) of the item specified. (3)

1.1.5 Alphanumeric Codes

Alphanumeric coding should be used when absolute identification is essential. However, problems with alphanumeric coding include confusability of similar symbols, providing space for the symbols, and learning the meanings of symbols. (11)
References: Christ, 1975; Grether & Baker, 1972.

1.2 Information Density

The number of items displayed simultaneously should be minimized. As the number of displayed items increases, so does the time required by the user to detect and extract information that has changed. One reference suggests that no more than 60% of the available character positions should be used. (1,11)
References: Coffey, 1961; Poulton & Brown, 1968; Schutz, 1961; Green, 1953; Shields, 1980.

Only information essential to the user's current needs should be displayed. (1)

* Interim data should automatically be removed from the screen once they are no longer needed. (6, 8)

Users should have the capability to eliminate irrelevant items from the display. Users should also be able to reverse these decisions. (11)
Reference: Stewart, 1974.

To avoid clutter, data should be presented using spacing, grouping, and columns to produce an orderly and legible display. (1)

1.3 Labeling

To make the display as meaningful as possible and to reduce user memory requirements, every variable or column heading should be labeled. Distinct and meaningful names should be selected to label columns of data. (1,3)

* The units for every variable or column heading that is displayed should be marked. (1)

Labels should be displayed in upper case only. (3)

Labels should have distinct and meaningful wording to distinguish them from data, error messages, etc. Jargon should not be used in labels. (1, 13)

Field labels should have a consistent format throughout the dialogue. (13)

Items continued on the next page (scrolled) should be numbered relative to the first item on the initial page. (1, 3)

* Every display frame should have a unique identification to provide a reference for use in requesting the display of that screen. The screen identification should be an alphanumeric code or abbreviation which is prominently displayed in a consistent location. It should be short enough (3-7 characters) or meaningful enough to be learned and remembered easily. (1)

1.4 Format

Standard data formats (e.g., MM/DD/YY) should be used. The military and other special occupation groups have their own standards. Therefore, use standards appropriate for the intended users. Formats should be changed only to differentiate similar tasks clearly. (3)

Identical data should be presented to the user in a standard and consistent manner, despite its origin or module. (3)

The same data formats should be used for related input and output. (13)

Formats for data entry should match the source document formats. (1, 10, 13)

Essential data, text, or formats should always be under computer, not user, control. Never assume voluntary compliance by the user. (3)

When meaningless arbitrary codes must be remembered or entered by the user, they should be no longer than four alphabetic characters or five digits. (1)

* Data entries should not exceed 5-7 characters. (13)

When items longer than seven characters must be entered, they should be partitioned into smaller symbol groups. (13)

1.4.1 Prompts

A special character can be used to denote an input prompt. Colons are commonly used for this purpose. If possible, a character which can be reserved to use only as an input prompt and for no other purpose during the transaction should be selected. (1)

Input prompts should be clear and understandable. They should not require reference to coding schemes or conventions which may be unfamiliar to infrequent system users. (1)

* Highlighting methods should be used to make prompts stand out. (1)

Input prompts should be placed in a consistent screen location, if possible. (1)

1.4.2 Tabular Data

* Data to be scanned and compared by the user should be presented in graphic or tabular form. As an option, the user should be able to look at raw data. (3)

Items in a list should each start on a new line. (1)

Items in a list should be arranged in some recognizable and useful order, such as chronological, alphabetical, sequential, functional, frequency of use, or importance. (1)

For items in a list that will not be used for input selection, "bullets" may be used to enumerate the items. (1)

For rapid scanning, lists should be left-justified and aligned vertically. Subclasses can be indented. (3)

Tabular displays should be broken into blocks, whenever possible. Breaking tabular displays into blocks improves search time. (11)
Reference: Cropper and Evans, 1968.

The computer should handle the left- or right-justification of data entries and the justification of numeric lists on the decimal point. (13)

When a list extends beyond the amount that can be shown on one display page, a short message should be provided to indicate that the list is not complete. (1)

1.4.3 Graphics

* Illustrations, line drawings, and animation should be used to supplement the explanations in the text. Graphics are especially useful for spatial visualization problems or where the problem to be solved has multiple interacting dimensions. Graphical dialogues are intrinsically motivating, at least for the novice user. (3, 11)

The axes of graphs should always be labeled. (6)

* The axes of graphs should be subdivided appropriately with divisions of 1, 2, 5, or 10, not with 3, 7, or other numbers obtained arbitrarily through division. (6)

If trend lines are to be compared, multiple lines should be used on a single graph. (11)
Reference: Schutz, 1961.

Symbols should be designed with consideration of the graphic conventions to which the user may be accustomed, while at the same time being as economical as possible in the use of screen space and image complexity. (8)

* Unnecessary ornamentation, unwanted graphic patterns and illusions, and flaws in alignment should be avoided in graphic displays. (8)

1.4.4 Textual Data

Active voice should be used, whenever possible. Active voice is generally easier to understand than passive voice. (1)

If a sentence describes a sequence of events, the word order in the sentence should correspond to the temporal sequence of events. (1)

Short simple sentences should be used. (1)

Sentences should begin with the main topic. (1)

Statements should be made in the affirmative. (1)

Text should be displayed in a mixture of upper and lower case, rather than in all upper case. (1, 3)

Text should be left-justified. (1, 3, 6, 9)

Paragraphs should be separated by at least one blank line. (3)

Hyphenation and unnecessary punctuation should be avoided. (1, 3)

No punctuation should be used in abbreviations. (3)

* In presenting data on small display screens, no more than 50-55 characters per line should be displayed. On larger display screens, text should be broken into two or more columns of 30-35 characters per line. Columns should be separated by at least 5 spaces if the text is not right-justified, otherwise by 3-4 blank spaces. (3)

1.4.5 Numeric Data

Long numeric fields should be punctuated with spaces, commas, hyphens, slashes, or by whatever is appropriate. Conventional punctuation schemes are preferred; if none exists, a space should be used between every third or fourth number. (1)

Lists of numbers without decimals should be right-justified. (1, 6, 9)

Lists containing decimals should use decimal alignment. (1)

* Dates should be entered in six digit code with field separators built into the protected fields. (10)

If the digits 0-9 must be visually presented, display them in a 3 X 3 matrix, with zero at the center bottom of a fourth row, similar to the 1-2-3 arrangement of the American touch-tone telephone. (3)

Leading zeros should not be required in numerical data except when needed for clarity. (1, 3, 13)

1.4.6 Alphanumeric Data

When a code consists of both letters and digits, each character type should be grouped together and not interspersed. (1)

Strings of five or more alphanumerics should be grouped into three or four characters where no natural split or predefined break occurs or should be grouped at natural breaks. (3)

1.5 Screen Layout

The organization of displayed fields should be standardized. Functional areas should remain in the same relative location on all frames. For example, functional areas reserved for a particular kind of data should remain in the same relative display location throughout the dialogue. (1, 3, 9)

Data should be arranged in logical groups: sequentially, functionally, by importance, or by frequency. (1)

* Data should be arranged on the screen so that the observation of similarities, differences, trends, and relationships is facilitated for the most common uses. (1)

* The display should not be divided into many small windows. (3)

* The user should be permitted to divide the screen into windows or functional areas of an appropriate size for the task. (8)

* Dashed lines may be used to segment the display. (6)

* The unused areas should be used to separate logical groups, rather than having all the unused area on one side of the display. (1)

* To discriminate among different classes of information, the screen should be functionally partitioned into different areas: for example, a main work area (20 lines), a preparation area (1-2 lines), a system facility indicator (1/2 line), a diagnostic area (1 line), and a fixed response area (1-4 lines). (7)

* The last four lines on each display page should be reserved for messages, to indicate errors, communication links, or system status. (10)

When command language is used for control input, an appropriate entry area should be provided in a consistent location on every display, preferably at the bottom of the screen if the cursor can be conveniently moved there. (13)

On large, uncluttered screens, the display or functional areas should be separated by blank spaces (3-5 rows and/or columns). On smaller and/or more cluttered screens, structure can be defined by other coding techniques, such as using different surrounding line types, line widths, intensity levels, geometric shapes, color, etc. (3)

Displays should be designed so that information relevant to sequence control should be distinctive in position and/or format. (13)

The home position for the cursor should be consistent across displays. (13)

Frequently appearing commands should appear in the same area of the display at all times. (3)

* To enhance important or infrequent messages and alarms, they should be placed in the central field of vision relative to the display window. (3)

Each display page should have a title that indicates the purpose of the page. (10)

* Instructions should stand out. For example, instructions may be preceded by a row of asterisks. (1, 6)

2. DIALOGUE MODES

Design considerations dealing with dialogue mode are organized into six general types of interaction including form-filling, computer prompting, menu selection, command languages, query languages, and restricted natural language. Form-filling is a structured dialogue mode in which the user provides information in designated fields on the interactive display. Considerations in form-filling involve the choice of default values, auditory feedback to the user, form layout, data entry procedures, and cursor movement to designated fields on the form. Prompting dialogue is a computer-initiated, rather than user-iniated, query mode. Menu selection is a type of structured dialogue in which the user must select among a variety of options. User design considerations include the order of options, the selection of codes for the options, the display layout of the menu, the content of the menu, and the control sequence of the menu dialogue. Command language dialogues allow the user to communicate with the computer by providing specific commands which specify various functions to be performed. Query languages are specialized command languages used to retrieve information from a data base. Restricted natural language is the most unstructured dialogue and is used as a flexible method to query a data base. Although sentence-like commands are used, vocabulary size and/or syntax may be restricted.

The design considerations related to dialogue mode are presented in this part under the following headings and subheadings.

2.0 *Choice of Dialogue Mode*

A consistent dialogue mode for a given equipment configuration should be used even when the equipment may be used for various applications. This permits standards to be drawn up for the various software designers who will work on developing support software and provides a consistent environment for the user. (6)

Choice of an appropriate dialogue mode should be based on user characteristics, skill, and training. If user characteristics are variable, a variety of dialogue types should be provided. (13)

* An appropriate degree of dialogue flexibility should be chosen. Highly flexible dialogues have been found to help very experienced computer users but to degrade performance of moderately experienced computer users to a significant degree, especially by increasing error rates. (11)
  References: Walther, 1973; Walther & O'Neil, 1974; Eason, 1976; Stewart, 1976; Carlisle, 1974.

2.1 *Form-Filling*

A form-filling dialogue should be selected when flexibility in data entry (optional items) is needed, the users will have only a moderate amount of training, and the computer response time may be slow. (6, 13)

A form-filling dialogue should be used when the user is typing in commands which have been written or typed previously on a hard copy form. (9)

* A form-filling dialogue should not be used when the computer must handle multiple form types and the computer response time is slow. In this situation it will take too long to display the different forms when the user must shift among forms. (9)

* A form-filling dialogue is not as flexible as a branching tree of questions, and error correction procedures may be difficult. (6)

2.1.1 *Default Values*

Currently defined default values should be displayed automatically in their appropriate data fields with the initiation of a data entry transaction. (13)

User acceptance of stored data or default values should be accomplished by a simple means such as by a single confirming keystroke. (13)

The user should be able to replace any default value during a particular transaction without changing the current default definition. (13)

2.1.2 *Auditory Feedback*

An auditory signal should be used to alert the user that an attempt has been made to enter data into a blank area rather than an entry field. (13)

2.1.3 *Form Layout*

A standard input form should be used. (9)

The image of the form on the display screen should look like the hard copy input form. (9)

Field labels should consistently indicate what data items are to be entered. (13)

Optional entries should be distinguished from required information. When an input is optional, a default value, if any, should be displayed. (1, 10, 13)

Input fields should be defined by implicit cues, such as underscores for maximum field length or colons for format, whenever possible. (1, 3, 13)

See also: Paragraph 1.5, Screen layout.

2.1.4 *Data Entry Procedures*

Data entry by overwriting a set of characters in a field is confusing and should not be used. (6)

When data other than text are to be entered into a computer-based form, the data should be entered by replacement of a special character such as underscores in a defined data field. (13)

* To reduce user waiting time, user entries should be collected in a buffer and the entire form should be updated at one time. (3)

When multiple items will be entered by a skilled typist, each entry field should end with an extra blank to permit consistent use of the tab key to move to the next field. (10, 13)

When an item length is variable, the starting position should be defined with a special character or the cursor. (3)

With variable length entries, the user should not have to left- or right-justify data entries within the field. (1, 13)

When an item length is variable, the user should not have to remove any unused underscores. (1, 3, 13)

When a dimensional unit is consistently associated with a particular entry field, the unit should be displayed as part of the fixed label rather than entered by the user. When the dimensional unit varies for a given field, it should be provided by the user. (13)

When required data entries have not been entered by the user but can be deferred, their omission should be indicated and either immediate or delayed input of the missing items should be allowed. When entry of a required data item is deferred, the user should enter a special symbol in the field to indicate that the missing item has been temporarily omitted rather than ignored. (13)

Item-by-item prompting for form-filling is very slow and should be employed only with novice or infrequent users. (6)

If no source document exists, data entry should be in a logical sequence, or all required fields should be filled before all optional fields. (13)

2.1.5 Cursor Movement

* Form-filling usually requires cursor manipulation by the user. Entry prompts should be arranged to minimize the requirements for cursor positioning. (1,6)

* To minimize the need for cursor movements the entry fields should be aligned. (1, 3)

For minimal cursor movement all entry areas should be aligned at the left side of the screen. (6)

Easy cursor movement should be employed for movement from field-to-field as well as from line-to-line and character position-to-character position. (9)

Non-entry areas of the display should be made inaccessible to the user via the cursor. (13)

2.2 Computer Prompting

Computer-initiated question-and-answer dialogues should be used for routine data entry tasks, when the data items are known and their order is constrained, when the computer response is fast, and when naive users are involved. (6, 9, 11, 13)

* Computer-initiated question-and-answer dialogues should be used when the information to be obtained cannot be placed on a list or easily encoded. (9)

Computer-initiated question-and-answer dialogues should not be used for frequent or experienced users of computer systems because there is little flexibility in the sequence of operation and the dialogue can be lengthy and often slow. However, Ramsey and Atwood indicate that this principle is a widespread belief with no empirical support. (11)

Computer prompting can be used to supplement other dialogue modes. (13)

* In question-and-answer dialogues an example of the correct syntax for each response should be given to the user, whenever possible. (9)

* In question-and-answer dialogues an example of the appropriate content for each response should be given to the user, whenever possible. (9)

2.3 Menu Selection

A menu selection dialogue should be used when the command set is so large that users are not likely to be able to commit all the commands to memory. (9)

Menu dialogues should be used where at least some of the users may not be familiar with all the functions of the system. (9)

Menu selection should be considered for routine tasks with fixed procedures requiring only minimal entry of arbitrary data. (7, 13)

Because little training is required for menu selection dialogues, they should be considered for inexperienced users. (9)

A menu selection dialogue should not be used when the transmission rate will be less than 1200 baud. Relatively fast computer response time is required for menu selection dialogues because the menu options must be transmitted and displayed for each selection. (9, 13)

Menu selection as a supplementary dialogue can be helpful when the command set is large. (9)

When menu selection is used to train novices to use a command language, the wording and order should be consistent with the command language. (13)

2.3.1 Order of Options

Menu items should be ordered in the list on the basis of a logical structure. (10, 13)
Reference: Palme, 1975.

Dependent or mutually exclusive options should be grouped together. (3)

Related options should be grouped from general to specific. (3, 9)

If the list has no logical structure, then items should be ordered according to a ranking of their expected frequency of use. (3, 10, 13)

If the list contains logical subunits, these subunits should be ranked in expected frequency of use and ordered accordingly. (10)

* In long lists (more than seven items) or where there is little difference in frequency of use of the options, the selections should be placed in alphabetic order. (3)

2.3.2 Selection Codes

When options can be selected by coded entry, the code associated with each option should be included on the display in some consistent, identifiable manner. (13)

2.3.2.1 Letter Codes

If menu selections must be made by keyed codes, options should be coded by the initial letter or first several letters of their displayed labels rather than by more arbitrary numeric codes. Exception: selection from long lists of options where line number might be an acceptable code alternative to keying entire item (13)

If letter codes are used for menu selection, they should be used consistently throughout the dialogue. (13)
Reference: Palme, 1979.
* NOTE: Several other references suggested that numbers, not letters or bullets, should be used to list selectable items. (1, 3)

2.3.2.2 Number Codes

* Menu items should be numbered beginning with one, not zero. (3, 10)

* A period should be used after the item selection number and at the end of the sentence. (1, 3)

* At least one blank should be used between the selection number and the text descriptor. (3)

* Selection numbers should be right-justified. (1, 3)
* However, another reference suggested that selection numbers should be left-justified. (10)

2.3.3 Menu Layout

Compatible formats, terminology, and selection ordering should be used at all levels of the dialogue. (3, 13)

* Each menu frame should present a set of selectable items and a space for entering the item selected. (10)

The field for entering the selection code should be separated from the menu items by at least one blank line. (10)

* Each page of options should have a title that reflects the question for which an answer is sought. (1, 3)

* Directions to the user, when provided, should always precede the list of choices. (3)

If the menu items are brief and if it seems to be logically appropriate, menu items may be arranged in two separate columns. (10)

* If two columns of options are used, the location of the columns on the screen should be balanced. (10)

When the number of selections can fit on one page in no more than two columns, a simple menu should be used. (9)

If the selection options exceed two columns, multiple step (hierarchic) menus should be used. Because multipage option lists will generally hinder learning and use, multipage menus should not be used. (3, 9, 10, 13)

If the selection list exceeds 10-15 items, then the designer should consider reorganizing the list into two separate menu frames, maintaining the logical organization within the hierarchy. (1, 3, 10, 13)

When the user must step through a sequence of menus to make a selection, the hierarchic structure should be designed, insofar as possible within the constraints of display space, to minimize the number of steps required. (13)

Displayed menu lists should be formatted to indicate the hierarchic structure of logically related groups of options rather than as an undifferentiated string of alternatives. (13)

If selection items have been grouped, a label should be given to each group. (13)

When hierarchic menus are provided, the user should be given some displayed indication of current position in the menus structure. (13)

* Selection codes should each be presented on a single line. (10)

When control inputs will be selected from a discrete set of options, then the menu of options should be displayed at the time of selection. (13)

A standard location for the user to enter the code for the selected item should be provided. (13)

The selection area should be prominently labeled for novice users. (13)

* If menu options are included as a portion of a display intended also for data review and/or data entry, the displayed labels for control input should incorporate some consistent distinguishing feature to indicate their special function. (13)

* Only one user entry should be required per menu. (6, 10, 13)

Menus should be presented successively in the same area of the display rather than simultaneously in different areas. (11)
Reference: Ramsey, unpublished study.

When selection among displayed options is to be accomplished by pointing, the cursor should be placed automatically on the first (most likely) option at initial display generation. (13)

When selection among displayed options is to be accomplished by keyed entry of a corresponding code, the cursor should be positioned automatically at the first character of the choice entry line (first unprotected field). (10, 13)

2.3.4 Menu Content

A displayed menu should include only options appropriate at that particular step in the transaction sequence, and for the particular user. However, menu displays for a system still under development might indicate future options not yet implemented, but those options should be specially designated in some way. (13)

The displayed menu should include all options but only the appropriate options for a particular step. (11, 13)
Reference: Baker & Goldstein, 1966.

* Whenever possible, the number of alternatives should be limited to 5-9 items. To increase the accuracy of comprehension of previously learned items within a new list, selections should be limited to 4-6 items. (3)

Control options that are generally available at any step in a transaction sequence may be treated as implicit options and need not be included in a menu of options. Frequently used implicit options should be input by special function keys, others by coded command entry. (13)

The wording of menu items should reflect the current concerns and likely questions of the user at that step in the transaction. (13)

Menu items should be worded so as to permit direct selection of an option as an acceptable control input, either by pointing or by code entry. Options should not be worded so as to imply a question requiring a yes/no answer. (10, 13)

2.3.5 Control Sequencing

If the user population is variable, various menus with different levels of detail should be provided. (9)

Multiple paths to accommodate both experienced and inexperienced users should be provided. For example, experienced users should be able to bypass the menu hierarchy and directly access a given menu by entering its page number or identification code. (1)

Menu frames should be sequenced in an order dictated by the logical flow of the user's analysis of the transaction. In some cases this will mean holding choices in computer memory within a transaction until the choice is relevant to later menu branching or to selection of an input or output frame. (10)

An initial menu of control options should always be available for user selection to serve as a consistent starting point for control inputs at the beginning of a transaction sequence. (10, 13)

The user should always have immediate access to critical or frequently accessed options. (13)

* Menu selections from the user should be accepted in either abbreviated or complete form. (1)

Users should be able to enter a series of menu selections (command stack) to speed the dialogue by avoiding the need to display each menu. (3, 10, 13)
Reference: Palme, 1979.

When an error occurs in a menu command stack, the computer should proceed as far as possible and then give a message indicating where it stopped processing and which commands could not be processed. (10)

Command stacking must be available when system response time is such that over 2 seconds is required to display a menu. (9)

2. *Command Languages*

Command language dialogues should be considered for sophisticated users working with a system having a large number of capabilities. (9)

Command language dialogue should be considered for tasks involving a wide range of user inputs which may be entered in an arbitrary sequence, where users may be highly trained in the interests of achieving efficient performance, and where computer response is expected to be relatively fast. Command languages should be concise, precise, powerful, and flexible. (6, 11, 13)

Command languages are inappropriate for most users who have not been trained to use them and do not wish to be (managers, general public, administrative staff). (6)

See also: Part 4, Command languages and command processing.

2.5 Query Languages

Query language dialogue should be considered as a specialized subcategory of general command language for tasks emphasizing unpredictable information retrieval (as in many analysis and planning tasks), with moderately trained users and fast computer response. (13)

If the user population is diverse, a partitioned query language may be appropriate where the easier "layers" are intended for users of limited computer sophistication. (2, 11)
Reference: Reisner, 1977.

With query languages the user's perception of the data base should be sufficiently structured so as to enable rapid identification of those parts in which the user is interested. (2)

The organization of the data base should match the organization perceived to be natural by the users. (2)
Reference: Codd, 1974.

Query languages should minimize the use of quantification terms such as some or all. (2)

2.6 Restricted Natural Language

Quasi-natural language should be considered when one cannot teach a command set. Restricted syntax or vocabulary size does not hinder problem formulation. (2)
References: Gould, Lewis, & Becker, 1976; Kelley, 1975.

Restricted natural language dialogue should be considered when unsophisticated users must use a system with a moderate number of functions. (9)

* Restricted natural language dialogue should be used when the set of commands can be made to reflect usage of common English language terms. (9)

3. USER INPUT DEVICES

Various user considerations are appropriate to determine the device by which the user makes a dialogue entry to the computer. The guidelines presented in this section are concerned with the selection of an input device, keyboard considerations for special function keys and cursor control, the use of pointing controls such as light pens and touch panels, the use of continuous controls such as trackballs and joysticks, the choice of graphic tablets for graphical data entry, and considerations for voice input.

The following headings and subheadings are used to organize user considerations concerning user input devices.

- 3.0 Data Entry Procedures
- 3.1 Selection of Input Device
- 3.2 Keyboards
  - 3.2.1 Special Function Keys
  - 3.2.2 Cursor Control
- 3.3 Direct Pointing Controls
- 3.4 Continuous Controls
- 3.5 Graphics Tablets
- 3.6 Voice Analyzers

3.0 Data Entry Procedures

Procedures for entering data should be standardized. The format, location, grammatical structure, and input mode should be as consistent as possible throughout the system. (1)

Data should be entered in units which are most familiar to the user. (1)

When longer items must be entered, the item should be partitioned into shorter symbol groups for data entry and display. (13)

The user should not be required to re-enter parameters that have not changed since the previous interaction. (1, 3)

The system should prompt for incorrect or missing data only. The user should not have to re-enter the entire command string. (3, 13)

* To reduce short-term memory load, the user should be allowed to enter highly familiar or redundant portions of a long list last. However, the sequence should not violate functional requirements (e.g., initial keying of area codes in telephone numbers). (3)

The user should not be required to enter data already known by the system. Only data that are unknown, necessary for security, ambiguous, or required for verification should be entered by the user. (1, 3, 10, 13)

The user should not be required to remember information not displayed on the current screen. The user should not have to decide what action to take from memory. (1)

The user should not have to enter information already available to the system, such as the current date. (1, 13)

The user should not have to re-enter repetitive data or calculate numbers. (13)

3.1 Selection of Input Device

Whenever possible, a single entry device should be used to eliminate time spent switching among devices. (3, 11, 13)
References: Earl & Goff, 1965; Card, English, & Burr, 1978.

Data entry mode should be shifted as few times as possible. (6, 13)

When data entry is a significant task function, it should be accomplished via the user's primary display. (13)

3.2 Keyboards

The amount of typing required should be minimized by using numbered lists and abbreviations when that can be done without ambiguity. (3)

3.2.1 Special Function Keys

When keyboard data entry is needed as well as position designation, keyboard special function keys should be used for cursor control. (7, 13)

Special function keys should be used to minimize the dialogue needed. (9)

Special function keys should be used where a command language is limited and is dominated by commands rather than parameter values. (6, 11)

Special function keys should be used for critical inputs to avoid syntax errors and minimize input time. (13)

For time consuming, complex, or repetitive interactions special function keys should be provided (e.g., NEXT PAGE, BACKUP, CONTINUE, HELP, OPTIONS, and HARD COPY). (1, 6, 13)

A special function key should be provided for users to turn off noncritical alarms. (13)

User confirmation of a control input or data entry should be accomplished with an explicitly labeled CONFIRM function key. Confirmation should not be accomplished by pushing some other key twice. (13)

A DITTO key should be provided to facilitate the entry of duplicative data, particularly when vertical repetition of entries is frequent. (13)

Function key assignments should be displayed at all times, preferably through direct marking. (1, 9)

If the uses of the keys vary across users, key caps or other overlays should be used to differentiate the functions of the special keys. (1, 9)

If a key is used for different functions depending upon the defined operational mode, then alternate self-illuminated labels should be provided to indicate which function is current. (13)

* Special function keys should be physically marked with functional labels (command labels) so that there will be no confusion as to their use. (13)

If direct marking or the use of overlays is not possible, the assigned key functions should be displayed on the screen. (1, 13)

Once a key has been assigned a given function, it should not be reassigned to a different function for a given user. (1, 13)

Special function keys not needed for current inputs should be temporarily disabled under computer control. Mechanical overlays manipulated by the user should not be used for this purpose. (10, 13)

Special function keys should not be shifted characters. (3, 13)

Fixed function keyboards should be considered when there is a small command set to be employed by naive users. (9)

The function keys should be back lighted when enabled if they are not always active. (9)

Special function keys should require only a single activation to accomplish their function and should not take on different functions with repeated activation. (13)

3.2.2 Cursor Control

Cursor control dialogues should be used for systems which have interactive graphics as their primary purpose, but which must use menu selection at some points. (9)

* The cursor should be able to move across the screen rapidly. Positioning of the cursor from any one point on the screen to another should not take more than 0.5 second for every 23-30 cm (9-12 inches) travel in any direction. (3)

The rate of control movement should be compatible with the positioning accuracy desired. (3)

The cursor should not drift. (3)

When cursor positioning is incremental by discrete steps, the step size of cursor movement should be consistent in the vertical and horizontal axes. (13)

When displayed character size is variable, incremental cursor positioning should have a step size corresponding to the currently selected character size. (13)

If proportional spacing is used for displayed text, the software should adjust the cursor movement automatically when the cursor is being positioned for data entry or data change. (13)

* When using cursor control dialogues, the target for the cursor should be at least ten times the size of the positioning accuracy required for interactive graphics or 0.6 cm (1/4 inch) square, whichever is smaller. (9)

The cursor should not obscure any other character displayed in the position designated by the cursor. (13)

When fine accuracy of positioning is required, the cursor should include a point designation feature. Accuracy with cursor positioning is usually limited to one character. (3, 13)

* Actual entry of a designated position should occur by an explicit user action distinct from cursor placement. (13)

When multiple cursors are used (e.g., one for alphanumeric entry and one for tracking), they should be distinct from one another. (13)

If multiple cursors are controlled by the same device, there should be a clear indication to the user which cursor is currently under control. (13)

If multiple cursors are separately controlled by different devices, their controls should be compatible in operation. (13)

See also: Part 7, Multiple users.

3.3 *Direct Pointing Controls (Light Pens/Touch Panels)*

Direct pointing controls, rather than cursor controls, should be used when item selection or position designation is the primary type of data entry. (3, 11, 13)
References: Earl & Goff, 1965; Goodwin, 1975.

* A light pen should be used for gross drawing or for tracking moving objects. (3)

* A light-pen dialogue should be used where the operators are likely to be unfamiliar with the commands and function of the system. (9)

A light pen should not be used for precision control. A light pen lacks precision control because of the pen's aperture, distance from the display surface, and parallax. (3)

* Because it may be awkward or difficult to use, a light pen should not be used with left-handed operators. (3)

The area in which an item is selectable should be as large as possible. The user should be able to specify a word or number by selecting anywhere within the area of that word or number and also in the area surrounding that choice. (3)

The selectable area should be as large as possible, including at least the size of the displayed label plus one-half a characters distance all around the label. (13)

* If a light pen is to be used continuously for more than 15 minutes or more than once every 5 minutes, the display screen should be placed in a horizontal or nearly horizontal position so that the user does not tire. (9)

3.4 Continuous Controls

Continuous position designation should be accomplished by continuous controls. (13)

A trackball should be considered to draw straight lines or circles. A trackball is superior to a light pen or joystick to draw straight lines or circles. (11)
Reference: Irving, Horinek, Walsh, & Chan, 1976.

When direction designation is based on graphic representation, then some "analog" means of entry should be provided such as a rotary control. (13)
Reference: Smith, 1962.

* To select particular words or characters from a text display, as in text editing, a mouse should be considered. A mouse is faster and more accurate than cursor control-keys or special function tab-keys. (11)
Reference: Card, English, & Burr, 1978.

3.5 Graphics Tablets

A stylus with graphics tablet should be used for graphic entry. However, recognition of hand printed characters by the system is very slow (fewer than 40 characters per minute) as compared with typewriter entry (averaging 200 characters per minute). (3, 7)

The graphics tablet should be at least as large as the graphics screen (minimum 1:1 mapping). (3)

3.6 Voice Analyzers

Voice input should be considered when the hands and eyes are already occupied. (9)

* Voice input should not be used when the ambient noise level exceeds 90 dbA unless special provisions are made. (9)

4. COMMAND LANGUAGES AND COMMAND PROCESSING

A major source of dialogue in interactive computer systems is the use of commands given to the computer by the human. User considerations in the design of these command languages include the number and organization of the commands; command nomenclature including abbreviated command names, argument formats, and separators and terminators; standard default values for command arguments; use of command languages for text editing; user control of commands in the form of multiple command inputs (command stacking), labeled command sequences (macros), and commands that execute immediately and supersede other commands (immediate commands); the acceptable system response for command execution; and the use of special commands for moving and scrolling displayed information.

The organization used in this section for the compilation of user considerations is as follows.

4.1 Command Organization

The number of commands in the command language should be minimized. (4)

A small number of commands with many possible arguments should be used. Most users use only a small subset of commands because command organization makes it difficult for users to recall more powerful commands. (7, 11)
Reference: Boies, 1974; Kennedy, 1974.

Multiple commands for the same function should not be available. (4)
NOTE: Another reference indicated that command synonyms should be permitted where various users may use different terms to mean the same thing. (9)

Use of quantifiers and logical operators in a command language should be avoided. If quantifiers and logical operators are essential, expect many errors and provide good feedback. Quantification information can be obtained by prompting the user from a menu. (2, 6)
Reference: Thomas, 1976.

The data base should be organized in a way perceived as natural by the user. (2)
References: Durding, Becker, & Gould, 1977; Codd, 1974.

In information retrieval systems, commands for global retrieval of related information should be available. Global commands should be provided only for data that are normally retrieved together. (2)
Reference: Potash, 1979.

The number of command modes in a command language should be minimized to avoid errors related to forgetting which mode you are in. (8)

4.2 Command Nomenclature

* Particularly with unsophisticated users, command names should reflect common language (e.g., English) usage. (9)

* Key words should be short to minimize the amount of typing required. (1, 4)

Distinct command names should be used. Semantically similar names, such as SUM and COUNT, should not be used. (1, 2, 11, 13)
References: Gould & Ascher, 1975.

* Command names for interactive and noninteractive languages should be identical. (3)

All words in a command language should be consistently used and standardized in meaning from one transaction to another and from one task to another. (1, 3, 13)

The words chosen for a command language should reflect the user's point-of-view and not the programmer's, correspond consistently with the user's operational language, and incorporate whatever jargon is common on the job. (3, 13)

All upper case or all lower case letters should be used within a code that is made up of more than one letter. (1)

Command names should be selected to minimize possible errors that could occur when misspellings produce valid command names. (1)

A command language should provide flexibility. For example, a user should be permitted to assign personal names to files as well as frequently used command sequences. (13)

4.2.1 Abbreviations

Users should have the option to enter either full command names or abbreviations. (1)

* Users should be allowed to define data-entry codes. (9)

Punctuation should not be used in abbreviations. (1, 3)

Simple truncation should be used to abbreviate command names. Novice users can type in the entire command, while experienced users can truncate it. [illegible])
Reference: Moses & P[illegible] 1979.

Contractions should not be used on electronic displays. (1)

Abbreviations should be considerably shorter than the original term. (1, 2)

Abbreviations should be mnemonically meaningful. (1, 2, 10)

Abbreviations should be distinctive to avoid confusion. (1, 13)

The user should be permitted to enter the full command name or an abbreviation. Allowing abbreviated command input is important to the experienced user. (1, 9)

Abbreviated command input should be consistent with unabbreviated command input. (1, 11)

Each word should have only one acceptable abbreviation. (1)

Abbreviations should be used to supply commands for writing programs. The computer should supply the full command and prompt for arguments or use defaults. (7)

Abbreviations should be permitted in text processing and expanded later by the computer. (7)
Reference: Schoonard & Boies, 1973.

* When alphabetic data-entry is required, restricted alphabetic sets should not be used. To resolve any input ambiguities resulting from hardware limitations on alphabetic sets, software should be provided to interrogate the user. (13)
Reference: Smith & Goodwin, 1971b.

Abbreviations the user is not likely to understand or remember should not be used just to make room for more data on the display. (1)

Autocompletion of command names by the computer should not be used. (2)
References: Reisner, 1977; Fields, Maisano, & Marshall, 1978.

4.2.2 Argument Formats

Keyword argument formats in which both the argument and its value are specified should be used. Positional argument formats in which argument values must be specified in a given order impose a greater memory load and result in more errors. (7, 11)
References: Weinberg, 1971; Heafner, 1975.

Argument menus should be used to construct commands when the commands have many often-used arguments. (11)

With a relatively small set of alternatives, an argument list (menu) should be provided to select missing information. (7)

4.2.3 Separators/Terminators

Insofar as possible, the user should not be required to provide punctuation in command entries. (13)

A standard delimiter, preferably a slash (/), should be used. (13)

If a delimiter is required to distinguish optional parameters, or the separate keyed entries in a stacked command, a standard symbol should be used consistently for that purpose, preferably the same symbol (slash) used to separate a series of data entries. (3, 13)
NOTE: Another reference suggested that a semicolon (;) be used as the delimiter between stacked commands. (1)

In a text-processing environment with a sentence orientation, special sentence separators should be used. (7)

Neither the user nor the computer program should have to distinguish between single and multiple blanks in a command entry. (13)

4.3 Defaults

Default values should be used to reduce operator workload. (11)

The user should select default values if the system designer cannot select appropriate levels. (13)

Use of default values promotes natural and concise dialogue. (6, 11)
Reference: Gilb & Weinberg, 1977.

* Defaults should be used between commands to supply missing commands, supply missing arguments, or supply a missing command when the arguments are given. (7)

* A profile of the user should be employed to set up defaults for program processing to match the language the user generally employs. (7)

The system should show the predefined default value, and the user should actively indicate acceptance of the default. (3)

4.4 Editor Orientation

For editing programs, a line editor orientation is appropriate. (7)

In text-processing environments, sentences, not lines, should be returned to the user. (7)
Reference: Stone & Webster Engineering, 1973.

In text editing the user should be able to search for synonyms and/or logical relations. (7)
References: Burton, 1974; Skinner, 1972; Mittman & Borman, 1975; Donzeau-Gouge, Huet, Kahn, Lang, & Levy, 1975; Kruskal, 1976; Wilks, 1973; Sauvain, 1971.

A scheme to number the lines in a program file should be provided for ease in editing and to permit communication between processors so that the line numbers are consistent in error messages. (7)

Scrolling should not be used when the user must discern a pattern. Scrolling is acceptable for locating an item in a list. (3)

4.5 User Control

The user should be able to manipulate data without concern for internal storage and retrieval mechanisms of the system. (13)

If control input is accomplished by command entry, then the user should have some consistent means to request prompting for options or control parameter values not already shown on the display. (13)

The sequence of transaction selections should generally be dictated by the user's choices and not by internal computer-processing constraints. (10, 13)

The user should be able to make at least some sequence control inputs directly at any step in a transaction sequence without having to return to a general options display. (13)

4.5.1 Command Stacking

Whenever possible, stacking of input or multiple entries should be allowed. This permits experienced users to work ahead. (1)

When command entries are prompted automatically, experienced users should be able to use command stacking to bypass the prompts. (13)

In command stacking, the user's inputs should be in the same order as they would normally be made in a succession of separate command entry actions. (3, 13)

Stacked commands should be entered by key word not selection number. (3)

4.5.2 Macros

To reduce the number of keystrokes required, users should be allowed to use user-defined macros (labeled command sequences) for frequently used command sequences. (9, 13)

4.5.3 Immediate Commands

An immediate command to cancel or abort an unwanted sequence or a well-defined transaction sequence of commands should be provided. (13)

The system should provide the capability to stop ongoing processing and return control to the user at any time with the use of immediately processed commands. (1, 10, 13)

Differently named options should be provided to accomplish different degrees of interruption in sequence control. The user should not have to push a single special function key a specific number of times to obtain a particular level of interruption in sequence control. (13)

Interrupting system processing by use of the ATTN key should lead to a menu of options. (6)

If appropriate to sequence control, a RESTART option should be provided which will have the consistent effect of returning to the first display in a defined transaction sequence, permitting the user to review a sequence of entries and make necessary changes. RESTART implies cancellation of any interim entries made in a pending transaction. (13)

If appropriate to sequence control, a BACKUP option should be provided which will have the consistent effect of returning to the display entered in the last previous transaction. BACKUP implies cancellation of any interim entries made in a pending transaction. (13)

If appropriate to sequence control, a CANCEL option should be provided which will have the consistent effect of regenerating the current display without processing any interim changes made by the user. (13)

If appropriate to sequence control, an END option should be provided, which will have the consistent effect of concluding a repetitive transaction sequence and returning control to a general options menu. (13)

To reduce the need for escape sequences, the system should give the user warning-information when a command is invoked which will be time consuming and/or expensive to process. (5)

The system should keep a record of the use of escape sequences, and this information should be used to redesign the system. (5)

4.6 Command Operation

Insofar as possible, sequence control software should be designed to carry forward a representation of the user's knowledge base and current activities. (10, 13)

Command operation should be consistent throughout the system. (1, 7, 10, 13)

Linked transactions should be the result of a task analysis to determine logical units. (10, 13)

The system should save sets of commands so that they can be checked and corrected without re-entering the entire sequence. (9)

Ease of command operation should be compatible with the desired ends: frequent procedures should be easy; destructive actions should be difficult. (13)

Command operation should involve a minimum number of control inputs by the user. Intermediate steps should be performed by the computer with feedback to the user, if necessary. (13)

Command sequencing should be flexible and under control by the user. The exception is emergency situations where the computer should automatically signal the user. (10, 13)

Command sequencing should be compatible with the user's skill level: step-by-step for beginners and efficient coding for experienced users. (13)

Sequence control should be closed-loop. The user should be required to take a specific action to leave a command loop such as text editing. (3)

To assure consistency when the user must perform similar activities on different equipment, certain procedural conventions should be standardized and presented as requirements. (1, 10)

A uniform interpretation of missing command parameters should be followed. (4)

The enter action for command entry should be the same as that for data entry or selection of menu options. (1, 13)

Commands for file manipulation and program compilation and execution should be consistent. The commands should process data files regardless of their size, content, or structure. (7)

At any step in a defined transaction sequence, if specific control options are not displayed, then a standard command should be provided so that the user can continue to the next step. (10, 13)

The user should be able to return easily to previous steps in a transaction sequence in order to correct an error or make any other desired change. (6, 13)

When considerations of data security are not involved, the user should be able to change any data that are currently displayed. (13)

The user should be required to take more complicated actions in order to respond to critical alarms and to acknowledge special alarms in special ways. (13)

Command sequencing should never result in a dead-end for the user. (3, 6, 10, 13)

A sensible next step should be provided at every point. The user should have the ability to back track to checkpoints established in a lengthy dialogue. (6)

4.7 System Response Time

NOTE: Specific values for ideal system response time under various conditions are not included in this list.

In complex problem-solving situations, artificial system lockout of user commands should be used to cause the user to concentrate on the problem. This procedure benefits problem solving, but may reduce user satisfaction. (11)
Reference: Stewart, 1976.
NOTE: Another reference indicated that artificial lockout should not be used for pacing and should not exceed 20 msec. (13)

When command stacking is not possible, keyboard lockout and disappearance of the cursor should be used to indicate that no user entry is allowed. (13)

User input should be user-paced. (13)

4.8 Special Commands

Special characters used in data entry (, * = /), particularly if used frequently, should be chosen insofar as possible so that the user will not have to shift from one case to another on the keyboard. (13)

Tab controls or other provisions for establishing and moving from field to field should be provided for editing programs. (7)

Easy to use MOVE and COPY commands should be provided for editing purposes. (7)

In text processing the MOVE or COPY commands should be based on sentences, paragraphs, or higher-order segments. (7)

For text processing, special editing commands for adding, inserting, or deleting text segments should be provided. (7)

Users should be provided a means to search for groups of related files and store the sorted collection into a new file for processing. (7)

If scrolling is incorporated for displaying portions of a large data base, commands for UP, DOWN, LEFT, and RIGHT should be devised in a standardized way. (1)

The ROLL and SCROLL commands should refer to the text/data, not the display window. (3)

5. FEEDBACK AND ERROR MANAGEMENT

Dialogue considerations contained in this section deal primarily with communications from the computer to the user. These communications include feedback, error recovery, user control of the transaction sequence, help/documentation, and computer aids. Feedback information provided by the computer incorporates design considerations for status messages, error messages, and hard copy output. User actions required to correct an error involve user correction procedures, computer metering of transactions, automatic correction, and stacking of multiple commands. Guidelines are considered for the level, amount, and type of user control of feedback and error messages. User considerations for on-line and off-line documentation as well as help information to enhance feedback and error management are presented. And, finally, guidelines for computerized program debugging and decision aids are listed.

The design considerations in this section are organized in the following manner:

- 5.1 Feedback
  - 5.1.1 Status Messages
  - 5.1.2 Error Messages
  - 5.1.3 Hard Copy Output
- 5.2 Error Recovery
  - 5.2.1 Immediate User Correction
  - 5.2.2 User Correction Procedures
  - 5.2.3 Metering and Automatic Error Checks
  - 5.2.4 Automatic Correction
  - 5.2.5 Stacked Commands
- 5.3 User Control
- 5.4 Help and Documentation
  - 5.4.1 Off-Line Documentation
  - 5.4.2 On-Line Documentation
- 5.5 Computer Aids
  - 5.5.1 Debugging Aids
  - 5.5.2 Decision Aids

5.1 Feedback

The log-on frame should be presented immediately after connection, regardless of user input. (1, 3, 6)

Keyed inputs, except security items such as passwords, should be echoed on the display. (3, 13)

When the user has entered a synonym for a command, the synonym should be used subsequently in system messages. (3)

* The wordiness of system messages should be adapted to the needs of the user population. Relatively brief messages using standard terms and abbreviations are appropriate for frequent system users. However, users should not have to "translate" messages via a reference system. (3, 10)

If the baud rate is less than 250 wpm (reading rate), compact messages should be used. (3)

Abbreviations should not be used in output unless necessary. Even then only meaningful, unique terms should be used. Similar abbreviations in the same entry should be avoided. (2, 3)
Reference: Moses & Potash, 1979.

When abbreviations must be used in system messages, they should be used consistently. (1)

The most difficult to remember information should be placed at the beginning of the message; the most easily remembered information in the middle. (3)

Information for immediate recall only should be placed at the end of a message. Items which must be remembered should be placed towards the beginning of the text. (3)

Information should be presented to the user in a directly usable form. The user should not be required to translate, transpose, change units, or interpolate. (1, 2, 3)

Jargon that would be unfamiliar to the user should not be used in system messages. Write from the user's point-of-view, not the programmer's. (1)

Standards and conventions for data presentation appropriate to the user should be used in system messages. (1)

Except for mathematical notation, standard alphabetic characters should be used for system messages. (3)

Alarm signals and messages may take a variety of forms, but should be distinctive and consistent for each class of events. (3, 13)

5.1.1 Status Messages

If a user cannot get onto the system, a message should be sent telling why and approximately when the problem is expected to be corrected. (1, 3)

Status information should be provided throughout the dialogue. The user should be informed when menu selections are accepted. (1, 3)

System messages should recap lengthy transactions periodically. (2)

After command interruption or a system crash, the user should receive a message that the system has been restored to its previous status. (1)

The user response necessary to continue the dialogue should be indicated on each display page. (1)

If a long response time is expected, confirmation of receipt of the request should be made as soon as possible. The computer should confirm completion of all requests (within 10-15 seconds). (3)

When command processing will be lengthy, the user should be informed that the request has been received and what action will result from the request. The user should be asked to confirm the request. The user should receive periodic messages that the request is being processed. (1, 2, 3, 4, 8, 13)

When the processing of delayed commands is completed, the user should be informed of the outcome and whether any user action is required. (13)

When a stored data entry is changed without first being displayed, both the old and new values should be displayed before action is taken. (13)

For efficiency, confirmation of repetitive data transactions can be accomplished by regeneration of the data entry page. (3, 13)

Coding, such as highlighting, should be used to indicate which option has been selected from a menu or what position has been designated by the user. (6, 13)

Confirmation of user input should occur without removing the display of data. (13)

Entry of multiple items should be acknowledged by the computer regardless of the cursor position. (13)

A status message should indicate the current functions of multiple-purpose special-function keys. (13)

If the consequences of the user's input will vary depending on prior commands, the user should be given the context by recapitulating previous inputs affecting present actions and indicating currently available options. (13)

When the user is forced to scroll through a large information display, an indication should be provided on the viewable portion of the display of present location versus maximum location. (3, 10)

When multiple modes of operation are possible, some means should be provided to remind the user of the current operating mode. (3, 13)

The value of any control parameter(s) currently operative should be displayed for user reference. (13)

Default values assumed should be displayed to the user. (3, 7, 13)

Information concerning control options specifically appropriate at any step in a transaction sequence should be provided for the user. (13)

5.1.2 Error Messages

* Error messages should appear as close as possible to the user entry that caused the error. (3)

Error messages should be appropriate to the user's level of knowledge. Error messages which may be useful to system analysts are often of little or no value for system users. (1, 3)

Error messages should be phrased politely. They should not place fault, use patronizing language, or attempt to be humorous. (1, 3, 10)

Error messages should provide information as to what error has been detected, where the error occurred, and how the user can correct the error. (1, 3, 7, 10)

* In a programming environment, the user should always be informed as to what rule was violated and where in the program the error occurred. (7)

To specify what remedial action to take it is permissible to refer the user to off-line documentation. (10)
NOTE: Another reference indicated that the user should not have to translate messages via an off-line reference system. (3)

* Error messages should begin with an identification number which corresponds with off-line documentation. Off-line documentation should be used to provide additional details, not as a "translation system" for obscure error message codes. (1, 10)

Error messages should be as specific to the user's particular application as possible. This kind of specificity requires more programming effort by the applications programmer, but contributes to the friendliness of the system to the end user. (10)

Error messages should be for quick reference only. Error messages should be brief, but informative. Off-line documentation or help sequences should be used to teach the system. (1, 3)

5.1.3 Hard Copy Output

If a video display terminal is to be the user's primary work station, an option to print a hard copy of the contents of the screen should be provided. Requiring the user to transcribe data from a video screen to notes not only underutilizes the data handling potential of a computer system, but also invites transcription errors. (1, 3)

5.2 Error Recovery

User action to correct an error should result in displayed changes in the state or value of the altered items. (13)

All error corrections by the user should be acknowledged by the computer either by indicating a correct entry has been made or by another error message (10)

5.2.1 Immediate User Correction

The user should be able to edit an extended command during its composition, by backspacing and rekeying, before taking an explicit action to ENTER the command. The user should be able to alter the input line during entry without retyping. Special function keys or special commands should be provided so that user input can be corrected immediately. (1, 3, 6, 7, 13)

When a data entry transaction has been completed and errors have been detected, the software should permit direct, immediate correction by the user. (1, 3, 10, 13)

5.2.2 User Correction Procedures

When missing data are detected, the system should prompt the user for these data, not use default values. (3)

When the missing data involve a relatively small set of alternatives, an argument list should be provided for the the user to select missing information. (7)

Users should be required to modify only the incorrect portion of an input. When the system detects an error, the cursor should automatically be positioned at the field which contains the first error, thereby minimizing the action required of the user. (1, 3, 13)

When a user completes the correction of an error, the user should be required to take an explicit action before the computer accepts the corrected inputs. (10, 13)

5.2.3 Metering and Automatic Error Checks

The system should monitor and record user errors continuously or on a sampling basis to aid in the design of future systems and to improve the current system. (1)

Spelling and other common errors should not produce valid system commands or initiate processing sequences which are different from those intended. (1)

The system should check all data entered by the user for appropiate format, appropriate content, and for missing data. (13)

Alphabetic data should be checked for stray digits or nonalphabetic codes. (10)

For variable-length numeric fields where an acceptable range can be defined, numeric data should be checked for stray alphabetic or non-numeric characters. (10)

For fixed-length fields, input should be checked for an incorrect number of characters or numbers. (10)

The computer should check for missing information required to complete the transaction. (10)

5.2.4 Automatic Correction

* When possible, a system should recognize common misspellings of a command and execute the command as if it had been spelled correctly. (1)

When a command entry is subject to misinterpretation or a default value has been assumed, the user should be asked to review a displayed interpretation for correction or confirmation. A positive action by the user should be required to validate data that have been corrected by the computer. (1, 2, 5, 10, 13)

If the user selects a special function key that is invalid at a particular step in the transaction, no system action should result except to display an advisory message indicating what functions are appropriate at that point. (10, 13)

5.2.5 *Stacked Commands*

When errors occur in stacked commands, the command sequence should be processed up to the error and then the user should receive an indication of the problem and guidance to permit completion of the control input. (1, 3, 13)

To prompt for correction of an error in stacked commands, the computer should display the page that needs to be corrected. (1, 13)

If stacked commands are potentially ambiguous, the computer should display the interpreted command sequence for user correction or confirmation prior to command processing. (2, 3, 13)

Reference: Codd, 1974.

5.3 *User Control*

In tasks where transaction sequences vary, the user should be able to request a displayed list of prior entries to determine present status. (3, 6, 13)

The user should be permitted to define the nature of each alarm as well as its initiating event. (13)

* The user should control whether a multiple-entry transaction is validated item-by-item. (13)

When line-by-line syntax checking is available, it should be a feature under user control. (3)

The user should be able to request prompts as necessary to determine required parameters in a command entry or to determine available options for an appropriate entry. (13)

* The user should be able to control the amount of detail given in the explanation of errors and other HELP facilities. (1, 3)

Data entry by the user should require a specific enter-action. (13)

The user should control the amount, format, and complexity of information from the system including core dumps, program outputs, and system messages. (3)

5.4 *Help and Documentation*

On-line documentation, off-line documentation, and help sequences should use consistent terminology. (13)

5.4.1 *Off-Line Documentation*

All error messages should be listed and explained in the off-line system documentation. (1, 9)

* Every nonmenu frame should contain a reference to a specific section of off-line documentation to provide a ready source of explanation. (10)

5.4.2 *On-Line Documentation*

After accessing help, the user should be provided with an easy way to return to the main dialogue. (6)

On-line access to help facilities should be provided for each command. (1)

* All error messages should be listed and explained in the on-line help sequences. (1)

* A dictionary of abbreviations and codes used should be available on-line. (1, 9)

On-line access to a list of system capabilities and subsystems should be provided. By showing the system components, options, and structure, the on-line reference capability permits the user to understand the use the system effectively. (1)

* When possible, natural language, rather than an hierarchic menu, should be used to invoke on-line documentation. (7)
Reference: Shapiro & Kwasny, 1975.

* If more details are needed, the user can ask for a continuation. Successive levels of the HELP request can go into greater detail. (1)

5.5 *Computer Aids*

5.5.1 *Debugging Aids*

Value ranges, bounds, and exceptions provided by the programmer in the program should be used to generate test cases for debugging programs. (7)

Editor assists can be used to prevent syntax errors in certain programming languages, such as parentheses balancing in LISP. (7)

5.5.2 Decision Aids

In subjective decision making the computer should inform the user of information that has been overlooked. (2)
References: Katter, Potash, & Halpin, 1978; Anderson & Gillogy, 1976; Waterman & Jenkins, 1977, Waterman, Anderson, Hayes-Roth, Klahr, Martin, & Rosenschein, 1979.

6. SECURITY AND DISASTER PREVENTION

Special dialogue considerations are presented in this section which deal with human-computer transactions that are directed toward the prevention of catastrophic circumstances, such as the inadvertent deletion of files or the premature termination of a computer session. These user considerations are concerned with topics such as the cancellation of data entry and command sequences, the requirement to confirm ambiguous or destructive actions, the control of destructive actions, and the handling of system crashes.

The topics in this section are organized as follows:

6.1 Command Cancellation
6.2 Verification of Ambiguous or Destructive Actions
6.3 Sequence Control
6.4 System Failures

6.1 Command Cancellation

When multiple items are entered as a single transaction, the user should be allowed to restart, cancel, or change any item before taking a final enter-action. (4, 13)

When data entries or changes will be nullified by an abort action, the user should be asked to confirm the abort. (13)

The user should be able to "take back" or undo the effect of at least the immediately preceding command. (7)

Whether or not errors have been committed, escape from a partially completed procedure must not lead to incorrect or accidental modification of stored data or the initiation or modification of other system functions. (1, 6)

6.2 Verification of Ambiguous or Destructive Actions

When a user signals to terminate the interactive session, the computer should check pending transactions to determine if data loss seems probable. If so, the computer should send an advisory message requiring confirmation before any log-off action is effected. (13)

One or more verification inputs should be required of the user to implement any critical action such as erasing a file, permanently modifying data, or changing system operation. (1, 3, 5, 9, 13)

When command entries are subject to misinterpretation (as in the case of voice input), the user should be given an opportunity to review and confirm the computer's interpretation of the command. (5, 13)

The prompt for the confirmation action should be worded in such a way that any potential data loss is clearly stated. (13)

Critical actions should not depend on one keystroke for verification. (3)

6.3 Sequence Control

The sequence control for commands that result in destructive action should be difficult. (13)

No user error should cause a session to be terminated or aborted. (1)

When a user fails to meet the security requirements for part of the data, this should not impede use of the open data and should not discontinue the dialogue. (4)

6.4 System Failures

The system should provide frequent automatic backups in order to restore files in case of a system crash. (5, 6, 7)

During system failures, errors should not be entered into user data-files. (6)

When a system does fail and terminals become inoperative, users should have some other means of dealing with the situations that confront them. (6)

In some cases when only part of the system fails, the user should be able to switch over to another piece of equipment. (6)

7. MULTIPLE USERS

Although most human-computer dialogue considerations are concerned with the design of the single-user interface, a few design considerations have been offered for the multiple-user environment. These considerations are concerned primarily with separating messages and inputs of multiple users, the use of cursors in multi-user displays, and computerized record keeping of inter-user messages.

This part organizes the pertinent design considerations into the following categories:

7.1 Separating Messages/Inputs

When two or more users must interact with the system simultaneously, control inputs by one user should not interfere with those of another. (13)

In on-line communication among users, the input from each speaker should be buffered to prevent any interference. (7)

The transmitter of each message in inter-user communications should be identifed, and separate areas of the display screen should be provided for each communicator. (7)

7.2 Separating Work Areas

With multi-user displays, multiple cursors, one for each communicator, should be provided. The active cursor for each user should be indicated. (7)
References: Chapanis, 1971; Ochsman & Chapanis, 1974.

In multi-user situations, each user should be provided with an individual work area for personal files as well as access to the shared work area. (7)

7.3 Communications Record

A permanent record (file) of inter-user messages should be made. (7)

REFERENCES

Anderson, R. H. and Gillogy, J. J. Rand intelligent terminal agent (RITA): Design philosophy (R-1809-ARPA). Santa Monica, California: Rand Corporation, February 1976.

Baker, J. D. and Goldstein, I. Batch vs. sequential displays: Effects on human problem solving. *Human Factors*, 1966, *8*, 225-235.

Barmack, J. E. and Sinaiko, H. W. Human factors problems in computer-generated graphic displays. Washington, D. C. : Institute for Defense Analyses, 1966. (AD-63617C)

Boies, S. J. User behavior on an interactive computer system. *IBM Systems Journal*, 1974, *13*, 2-18.

Brown, C. M., Burkleo, H. V., Mangelsdorf, J. E., Olsen, R. A., and Williams, A. R., Jr. Human factors engineering criteria for information processing systems. Sunnyvale, California: Lockheed, October 1980.

Burton, R. R. A semantically centered parsing system for mixed-initiative CAI systems. Paper presented at the Association for Computational Linguistics Conference, Amherst, Massachussetts, July 1974.

Card, S. K., English, W. K., and Burr, B. J. Evaluation of mouse, rate-controlled isometric joystick, step keys, and text keys for text selection on a CRT. *Ergonomics*, 1978, *21*, 601-613.

Carlisle, J. H. Man-computer interactive problem solving: Relationships between user characteristics and interface complexity. Doctoral Dissertation, Yale University, 1974.

Chapanis, A. Prelude to 2001: Explorations in human communication. *American Psychologist*, 1971, *26*, 949-961.

Christ, R. E. Review and analysis of color coding research for visual displays. *Human Factors*, 1975, *17*, 542-570.

Codd, E. F. Seven steps to rendezvous with the casual user (Tech. Rep. RJ-1333). San Jose, California: IBM Research Laboratory, January 1974.

Coffey, J. L. A comparison of vertical and horizontal arrangements of alpha-numeric material. *Human Factors*, 1961, *3*, 93-98.

Cropper, A. G. and Evans, S. J. W. Ergonomics and computer display design. *The Computer Bulletin*, 1968, *12*, 94-98.

Donzeau-Gouge, V., Huet, G., Kahn, G., Lang, B., and Levy, J. J. A structure oriented program editor: a first step towards computer assisted programming (Res. Rep. No. 114). Paris, France: Laboratoire de recherche en informatique et automatique, 1975.

Durding, B. M., Becker, C. A., and Gould, J. D. Data organization. *Human Factors*, 1977, *19*, 1-14.

Earl, W. K. and Goff, J. D. Comparison of two data entry methods. *Perceptual and Motor Skills*, 1965, *20*, 369-384.

Eason, K. D. A task-tool analysis of manager-computer interaction. Paper presented at the NATO Advanced Study Institute on Man-Computer Interaction, Mati, Greece, September 1976.

Ehrenreich, S. L. Query languages: Design recommendations derived from the human factors literature. *Human Factors*, in press. Also: Design recommendations for query languages (Tech. Rep. 484). Alexandria, Virginia: U. S. Army Research Institute for the Behavioral and Social Sciences, 1981.

Engel, S. E. and Granda, R. E. Guidelines for man/display interfaces (TR 00.2720). Poughkeepsie, New York: IBM, December 1975.

Fields, A. F., Maisano, R. E., and Marshall, C. F. A comparative analysis of methods for tactical data inputting (Tech. Paper 327). Alexandria, Virginia: Army Research Institute for the Behavioral and Social Sciences, September 1978. (AD A060 562)

Foley, J. D. and Wallace, V. L. The art of natural graphic man-machine conversation. *Proceedings of the IEEE*, 1974, *62*, 462-471.

Gebhardt, F. and Stellmacher, I. Design criteria for documentation retrieval languages. *Journal of the American Society for Information Science*, 1978, *29*, 191-199.

Gilb, T. and Weinberg, G. M. *Humanized Input*. Cambridge, Massachusetts: Winthrop, 1977.

Goodwin, N. C. Cursor positioning on an electronic display using lightpen, lightgun, or keyboard for three basic tasks. *Human Factors*, 1975, *17*, 289-295.

Gould, J. D. and Ascher, R. N. Use of an IQF-like query language by non-programmers (Res. Rep. No. RC-5279). Yorktown Heights, New York: IBM Watson Research Center, February 1975.

Gould, J. D., Lewis, C., and Becker, C. A. Writing and following procedural, descriptive, and restricted syntax language instructions (Res. Rep. No. RC-5943). Yorktown Heights, New York: IBM Watson Research Center, April 1976.

Green, B. F. The time required to search for numbers on large visual displays (Tech. Rep. 36). Lexington, Massachusetts: MIT Lincoln Laboratory, 1953.

Grether, W. F. and Baker, C. A. Visual presentation of information. In H. P. Van Cott and R. G. Kinkade (Eds.) *Human engineering guide to equipment design*. (rev. ed.) Washington, D. C.: U.S. Government Printing Office, 1972, 41-121.

Heafner, J. F. Protocol analysis of man-computer languages: Design and preliminary findings (Rep. No. ISI/RR-75-34). Marina del Rey, California: University of Southern California, Information Sciences Institute, July 1975.

Hiltz, S. R. and Turoff, M. *The network nation*. Reading, Massachusetts: Addison-Wesley, 1978.

Irving, G. W., Horinek, J. J., Walsh, D. H., and Chan, P. Y. ODA pilot study II: Selection of an interactive graphics control device for continuous subjective functions applications (Rep. No. 215-2). Santa Monica, California: Integrated Sciences Corporation, April 1976.

Katter, R. V., Potash, L. M., and Halpin, S. M. MIQSTURE: Design for a mixed initiative structure with task and user related elements. *Proceedings of the 22nd annual meeting of the Human Factors Society*, Detroit, Michigan, October 1978.

Kelly, M. J. Limited vocabulary natural language dialogue. In *Proceedings of the 19th annual meeting of the Human Factors Society*, Santa Monica, California: Human Factors Society, 1975, 296-300.

Kennedy, T. C. S. The design of interactive procedures for man-machine communication. *International Journal of Man-Machine Studies*, 1974, *6*, 309-334.

Kruskal, V. J. An editor for parametric programs (Rep. RC 6070). Yorktown Heights, New York: IBM Research Laboratory, 1976.

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

Miller, L. A. and Thomas, J. C., Jr. Behavioral issues in the use of interactive systems (RC 6326). Yorktown Heights, New York: IBM, December 1976.

Mittman, B. and Borman, L. *Personalized data base systems*. Los Angeles: Melville Publishing, 1975.

Moses, F. L. and Potash, L. M. Assessment of abbreviation methods for automated tactical systems (Tech. Rep. 398). Alexandria, Virginia: Army Research Institute for the Behavioral and Social Sciences, August 1979.

Newman, W. M. and Sproull, R. F. *Principles of interactive computer graphics*. New York: McGraw-Hill, 1979.

Ochsman, R. B. and Chapanis, A. The effects of ten communication modes on the behavior of teams during cooperative problem-solving. *International Journal of Man-Machine Studies*, 1974, *6*, 579-619.

Palme, J. Interactive software for humans (Rep. C 10029-M3(E5)). Stockholm: Swedish National Defense Research Institute, Planning and Operations Research July 1975.

Palme, J. A human-computer interface for non-computer specialists. *Software--practice and experience*. 1979, *9*, 741-747.

Parrish, R. N., Gates, J. L., Munger, S. J., and Sidorsky, R. C. Development of design guidelines and criteria for user/operator transactions with battlefield automated systems. Volume IV: Provisional guidelines and criteria for the design of user/operator transactions (Draft Final Report, Phase I). Alexandria, Virginia: U. S. Army Research Institute, 1981.

Pew, R. W. and Rollins, A. M. Dialog specification procedures (rev. ed.) (Rep. No. 3129). Cambridge, Massachusetts: Bolt Beranek and Newman, Inc., September 1975.

Potash, L. M. Effects of retrieval term specificity on information retrieval from computer-based intelligence systems (Tech. Rep. 379). Alexandria, Virginia: Army Research Institute for the Behavioral and Social Sciences, July 1979.

Poulton, E. C. and Brown, C. H. Rate of comprehension of an existing teleprinter output and of possible alternatives. *Journal of Applied Psychology*, 1968, *52*, 16-21.

Prince, M. D. *Interactive graphics for computer-aided design*. Reading, Massachusetts: Addison-Wesley, 1971.

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

Reisner, P. Use of psychological experimentation as an aid to development of a query language. *IEEE Transactions on Software Engineering*, 1977, *SE-3*, 218-229.

Sauvain, R. W. Structural communication in a personal information storage and retrieval system. Ann Arbor, Michigan: University Microfilms, MS. No. 70-21 782, 1971.

Schoonard, J. W. and Boies, S. J. Short-type: A behavioral analysis of typing and text entry (Res. Rep. RC 4434). Yorktown Heights, New York: IBM Research Laboratory, 1973.

Schutz, H. G. An evaluation of methods for presentation of graphic multiple trends: Experiment III. *Human Factors*, 1961, *3*, 108-119.

Shapiro, S. and Kwasny, S. Interactive consulting via natural language. *Communications of the ACM*, 1975, *18*, 459-462.

Shields, N. Spacelab display design and command usage guidelines (Tech. Rep. MFSC-PROC-711A). Moffett Field, California: NASA AMES, April 1980.

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

Skinner, V. L., Jr. Text processing applications (Tech. Rep. FSC 72-6014). Gaithersburg, Marland: IBM, Federal System Division, 1972.

Smith, S. L. Angular estimation. *Journal of Applied Psychology*, 1962, *46*, 240-246.

Smith, S. L. Man-machine interface (MMI) requirements definition and design guidelines: A progress report (ESD-TR-81-113). Bedford, Massachusetts: MITRE, February 1981. (AD A096 705)

Smith, S. L. and Goodwin, N. C. Blink coding for information display. *Human Factors*, 1971, *13*, 283-290. (a)

Smith, S. L. and Goodwin, N. C. Alphabetic data entry via the Touch-Tone pad: A comment. *Human Factors*, 1971, *13*, 189-190. (b)

Stewart, T. F. M. Ergonomic aspects of man-computer problem solving. *Applied Ergonomics*, 1974, *5*, 209-212.

Stewart, T. F. M. Displays and the software interface. *Applied Ergonomics*, 1976, *7*, 137-146.

Stone and Webster Engineering. Reported on in Engineering, editing, and problem solving. *IBM Computing Report*, 1973, *Fall issue*, 8-9.

Teichner, W. H., Christ, R.E., and Corso, G. M. Color research for visual displays (Rep. No. ONR-CR213-102-4F). Las Cruces, New Mexico: New Mexico State University, Department of Psychology, June 1977.

Thomas, J. C. Quantifiers and question-asking (Rep. RC5866). Yorktown Heights, New York: IBM Research Laboratory, 1976.

Vartabedian, A. G. Effects of parameters of symbol formation on legibility. *Information Display*, 1970, *7* (5), 23-26.

Walther, G. H. The on-line user-computer interface: The effects of interface flexibility, experience, and terminal-type on user-satisfaction and performance. Colorado Springs, Colorado: U.S. Air Force Academy, Department of Astronautics and Computer Science, August 1973

Walther, G. H. and O'Neil, H. F., Jr. On-line user-computer interface: The effects of interface flexibility, terminal type, and experience on performance. *AFIPS Conference Proceedings*, 1974, *43*, 379-384.

Waterman, D. A. and Jenkins, B. M. Heuristic modeling using rule-based computer systems (Rep. P-5811). Santa Monica, California: Rand Corporation, March 1977.

Waterman, D. A., Anderson, R. H., Hayes-Roth, F., Klahr, P., Martin, G., and Rosenschein, S. J. Design of a rule-oriented system for implementing expertise (N-1158-1-ARPA). Santa Monica, California: Rand Corporation, May 1979.

Weinberg, G. M. *The psychology of computer programming*. New York: Van Nostrand Reinhold, 1971

Wilks, Y. Semantic considerations in text processing. In E. E. Gloye and R. J. Marcus (Eds.) *Proceedings of computer text processing and scientific research*. Pasadena, California, Office of Naval Research, March 1973, 39-54.

442:MAT:716:maf
81u442-390

OFFICE OF NAVAL RESEARCH

Code 442

TECHNICAL REPORTS DISTRIBUTION LIST

OSD

CDR Paul R. Chatelier
Office of the Deputy Under Secretary of Defense
OUSDRE (E&LS)
Pentagon, Room 3D129
Washington, D.C. 20301

Department of the Navy

Leader
Engineering Psychology Programs
Code 442
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217 (5 cys)

Leader
Communication & Computer Technology
Code 240
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Leader
Manpower, Personnel and Training
Code 270
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Dr. A. Meyrowitz
Information Systems Program
Code 411-IS
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Leader
Physiology Program
Code 441
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Department of the Navy

Special Assistant for Marine Corps Matters
Code 100M
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Commanding Officer
ONR Eastern/Central Regional Office
ATTN: Dr. J. Lester
Building 114, Section D
666 Summer Street
Boston, MA 02210

Commanding Officer
ONR Branch Office
ATTN: Dr. C. Davis
536 South Clark Street
Chicago, IL 60605

Commanding Officer
ONR Western Regional Office
ATTN: Mr. R. Lawson
1030 East Green Street
Pasadena, CA 91106

Commanding Officer
ONR Western Regional Office
ATTN: Dr. E. Gloye
1030 East Green Street
Pasadena, CA 91106

Office of Naval Research
Scientific Liaison Group
American Embassy, Room A-407
APO San Francisco, CA 96503

Director
Naval Research Laboratory
Technical Information Division
Code 2627
Washington, D.C. 20375 (6 cys)

442:MAT:716:maf
81u442-390

Department of the Navy

Dr. L. Chmura
Code 7503
Naval Research Laboratory
Washington, DC 20375

Dr. Michael Melich
Communications Sciences Division
Code 7500
Naval Research Laboratory
Washington, D.C. 20375

Dr. Robert G. Smith
Office of the Chief of Naval
Operations, OP987H
Personnel Logistics Plans
Washington, D.C. 20350

Dr. Jerry C. Lamb
Combat Control Systems
Naval Underwater Systems Center
Newport, RI 02840

Naval Training Equipment Center
ATTN: Technical Library
Orlando, FL 32813

Human Factors Department
Code N215
Naval Training Equipment Center
Orlando, FL 32813

Dr. Alfred F. Smode
Training Analysis and Evaluation
Group
Naval Training Equipment Center
Code N-OOT
Orlando, FL 32813

Dr. R. Neetz
Code 1226
Naval Missile Test Center
Pt. Mugu, CA 93042

Dr. Albert Colella
Combat Control Systems
Naval Underwater Systems Center
Newport, RI 02840

Department of the Navy

Dr. Gary Poock
Operations Research Department
Naval Postgraduate School
Monterey, CA 93940

Dean of Research Administration
Naval Postgraduate School
Monterey, CA 93940

Mr. Warren Lewis
Human Engineering Branch
Code 8231
Naval Ocean Systems Center
San Diego, CA 92152

Dr. Robert French
Naval Ocean Systems Center
San Diego, CA 92152

Dr. Ross L. Pepper
Naval Ocean Systems Center
Hawaii Laboratory
P.O. Box 997
Kailua, HI 96734

Dr. A. L. Slafkosky
Scientific Advisor
Commandant of the Marine Corps
Code RD-1
Washington, D.C. 20380

Dr. Thomas McAndrew
Code 32
NUSC-New London
New London, CT 06320

HQS, U.S. Marine Corps
ATTN: CCA40 (MAJOR Pennell)
Washington, D.C. 20380

Commanding Officer
MCTSSA
Marine Corps Base
Camp Pendleton, CA 92055

Chief, C^3 Division
Development Center
MCDEC
Quantico, VA 22134

442:MAT:716:maf
81u442-390

Department of the Navy

Mr. Arnold Rubinstein
Naval Material Command
NAVMAT 0722 - Rm. 508
800 North Quincy Street
Arlington, VA 22217

Commander
Naval Air Systems Command
Human Factors Programs
NAVAIR 340F
Washington, D.C. 20361

Commander
Naval Air Systems Command
Crew Station Design,
NAVAIR 5313
Washington, D.C. 20361

Mr. Phillip Andrews
Naval Sea Systems Command
NAVSEA 0341
Washington, D.C. 20362

Commander
Naval Electronics Systems Command
Human Factors Engineering Branch
Code 4701
Washington, D.C. 20360

CAPT Darrell D. Dempster, SC, USN
PMA 270
Naval Air Systems Command
Washington, DC 20361

CDR Robert Biersner
Naval Medical R&D Command
Code 44
Naval Medical Center
Bethesda, MD 20014

Dr. Arthur Bachrach
Behavioral Sciences Department
Naval Medical Research Institute
Bethesda, MD 20014

Dr. George Moeller
Human Factors Engineering Branch
Submarine Medical Research Lab
Naval Submarine Base
Groton, CT 06340

Department of the Navy

Head
Aerospace Psychology Department
Code L5
Naval Aerospace Medical Research Lab
Pensacola, FL 32508

Dr. James McGrath, Code 302
Navy Personnel Research and
Development Center
San Diego, CA 92152

Dr. Mel C. Moy
Code 302
NPRDC
San Diego, CA 92152

Mr. Ramon L. Hershman
Code 302
NPRDC
San Diego, CA 92152

Navy Personnel Research and
Development Center
Planning & Appraisal
Code 04
San Diego, CA 92152

Navy Personnel Research and
Development Center
Management Systems, Code 303
San Diego, CA 92152

Navy Personnel Research and
Development Center
Performance Measurement &
Enhancement
Code 309
San Diego, CA 92152

LCDR Stephen D. Harris
Code 6021
Naval Air Development Center
Warminster, PA 18974

Dr. Julie Hopson
Human Factors Engineering Division
Naval Air Development Center
Warminster, PA 18974

Department of the Navy

Mr. Jeffrey Grossman
Human Factors Branch
Code 3152
Naval Weapons Center
China Lake, CA 93555

Human Factors Engineering Branch
Code 1226
Pacific Missile Test Center
Point Mugu, CA 93042

Dean of the Academic Departments
U.S. Naval Academy
Annapolis, MD 21402

Mr. John Impagliazzo
Code 101
NUSC - Newport
Newport, RI 02840

Human Factors Section
Systems Engineering Test
Directorate
U.S. Naval Air Test Center
Patuxent River, MD 20670

Human Factor Engineering Branch
Naval Ship Research and Development
Center, Annapolis Division
Annapolis, MD 21402

CDR W. Moroney
Code 55MP
Naval Postgraduate School
Monterey, CA 93940

Mr. Merlin Malehorn
Office of the Chief of Naval
Operations (OP-115)
Washington, D.C. 20350

Walter P. Warner
Code K02
Strategic Systems Dept.
Naval Surface Weapons Center
Dahlgren, VA 22448

Dr. Thomas Fitzgerald
Code 101
NUSC - Newport
Newport, RI 02840

Department of the Army

Mr. J. Barber
HQS, Department of the Army
DAPE-MBR
Washington, D.C. 20310

Dr. Joseph Zeidner
Technical Director
U.S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Director, Organizations and
Systems Research Laboratory
U.S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Technical Director
U.S. Army Human Engineering Labs
Aberdeen Proving Ground, MD 21005

ARI Field Unit-USAREUR
ATTN: Library
C/O ODCSPER
HQ USAREUR & 7th Army
APO New York 09403

Department of the Air Force

U.S. Air Force Office of Scientific
Research
Life Sciences Directorate, NL
Bolling Air Force Base
Washington, D.C. 20332

Chief, Systems Engineering Branch
Human Engineering Division
USAF AMRL/HES
Wright-Patterson AFB, OH 45433

Air University Library
Maxwell Air Force Base, AL 36112

Dr. Earl Alluisi
Chief Scientist
AFHRL/CCN
Brooks AFB, TX 78235

442:MAT:716:maf
81u442-390

Foreign Addressees

North East London Polytechnic
The Charles Myers Library
Livingstone Road
Stratford
London E15 2LJ
ENGLAND

Dr. Kenneth Gardner
Applied Psychology Unit
Admiralty Marine Technology
Establishment
Teddington, Middlesex TW11 OLN
ENGLAND

Director, Human Factors Wing
Defence & Civil Institute of
Environmental Medicine
Post Office Box 2000
Downsview, Ontario M3M 3B9
CANADA

Dr. A. D. Baddeley
Director, Applied Psychology Unit
Medical Research Council
15 Chaucer Road
Cambridge, CB2 2EF
ENGLAND

Professor B. Shackel
Department of Human Sciences
University of Technology
Loughborough, LEICS. LE11 3TU
ENGLAND

Other Government Agencies

Defense Technical Information Center
Cameron Station, Bldg. 5
Alexandria, VA 22314 (12 cys)

Dr. Craig Fields
Director, Cybernetics Technology
Office
Defense Advanced Research Projects
Agency
1400 Wilson Blvd
Arlington, VA 22209

Other Government Agencies

Dr. Lloyd Hitchcock
Federal Aviation Administration
ACT 200
Atlantic City Airport, NJ 08405

Dr. M. Montemerlo
Human Factors & Simulation
Technology, RTE-6
NASA HQS
Washington, D.C. 20546

Other Organizations

Dr. Jesse Orlansky
Institute for Defense Analyses
400 Army-Navy Drive
Arlington, VA 22202

Dr. T. B. Sheridan
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139

Dr. Arthur I. Siegel
Applied Psychological Services, Inc.
404 East Lancaster Street
Wayne, PA 19087

Dr. Robert T. Hennessy
NAS - National Research Council
JH #819
2101 Constitution Ave., N.W.
Washington, DC 20418

Mr. John Hemingway
Man-Vehicle Systems Research
MS# 239-21
NASA Ames Research Center
Moffett Field, CA 94035

Dr. M. G. Samet
Perceptronics, Inc.
6271 Variel Avenue
Woodland Hills, CA 91364

Dr. Elizabeth Kruesi
General Electric Company
Information Systems Programs
1755 Jefferson Davis Highway
Arlington, VA 22202

442:MAT:716:maf
81u442-390

Other Organizations

Journal Supplement Abstract Service
American Psychological Association
1200 17th Street, N.W.
Washington, D.C. 20036 (3 cys)

Mr. Edward M. Connelly
Performance Measurement
Associates, Inc.
410 Pine Street, S.E.
Suite 300
Vienna, VA 22180

Dr. Babur M. Pula
Department of Industrial Engineering
North Carolina A&T State University
Greensboro, NC 27411

Dr. Richard W. Pew
Information Sciences Division
Bolt Beranek & Newman, Inc.
50 Moulton Street
Cambridge, MA 02138

Dr. David J. Getty
Bolt Beranek & Newman, Inc.
50 Moulton Street
Cambridge, MA 02138

Dr. Douglas Towne
University of Southern California
Behavioral Technology Laboratory
3716 S. Hope Street
Los Angeles, CA 90007