DOCUMENT RESUME

ED 234 77_ IR 010 846

AUTHOR McCann, Patrick H.
TITLE Methods for Improving the User-Computer Interface. Technical Report.
INSTITUTION Navy Personnel Research and Development Center, San Diego, Calif.
REPORT NO NPRDC-TR-83-29
PUB DATE Aug 83
NOTE 18p.
PUB TYPE Information Analyses (070)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Automation; Behavioral Science Research; *Computer Programs; *Design Requirements; *Human Factors Engineering; Individual Needs; Literature Reviews; *Man Machine Systems; Models; Training Methods; *Work Environment
IDENTIFIERS *Shipboard Computers

ABSTRACT

This summary of methods for improving the user-computer interface is based on a review of the pertinent literature. Requirements of the personal computer user are identified and contrasted with computer designer perspectives towards the user. The user's psychological needs are described, so that the design of the user-computer interface may be designed to accommodate them. Development of the user-computer interface is discussed in terms of the user's physical, perceptual, and conceptual contacts with the system, and the ideals of the system design--transparency and visibility to the user--are described. Twenty-one dialogue principles identified by a review of dialogue design studies are listed. Additional topics include work station design guidelines and some relevant variables that should be considered in the operator's physical environment. Further research is suggested that will explore the characteristics of efficacious menu selection, develop a theory of the operator, determine the best locus of control for dialogue features, provide guidelines for improving system documentation, and improve user work station habitability. Twenty-four references are listed. (Author/LMM)

NPRDC TR 83-29 August 1983

METHODS FOR IMPROVING THE USER-COMPUTER INTERFACE

Patrick H. McCann

Reviewed by
Robert E. Blanchard

Released by
J. W. Renard
Commanding Officer

Navy Personnel Research and Development Center
San Diego, California 92152

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: NPRDC TR 83-29
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): METHODS FOR IMPROVING THE USER-COMPUTER INTERFACE
5. TYPE OF REPORT & PERIOD COVERED: Technical Report, Feb. 1981-Dec. 1982
6. PERFORMING ORG. REPORT NUMBER: 17-83-8
7. AUTHOR(s): Patrick H. McCann
8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Navy Personnel Research and Development Center, San Diego, California 92152
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: SF57525001
11. CONTROLLING OFFICE NAME AND ADDRESS: Navy Personnel Research and Development Center, San Diego, California 92152
12. REPORT DATE: August 1983
13. NUMBER OF PAGES: 19
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)

Management information systems
Computer users
System design
Computer dialogue
User-computer models
Shipboard management
Automation; shipboard functions
User interface
Work station design

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

Requirements of the personal computer user are identified and contrasted with computer designer perspectives towards the user. The user's psychological needs are described so that the design of the user-computer interface may be designed to accommodate them. Development of the user-computer interface is discussed in terms of the user's physical, perceptual, and conceptual contacts with the system. The ideals of system design—transparency and visibility to the user--are described.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Further research is suggested that will explore the characteristics of efficacious menu selection, develop a theory of the operator, determine the best locus of control for dialogue features, provide guidelines for improving system documentation, and improve user work station habitability.

FOREWORD

This effort was conducted within exploratory development task area SF57525001 (Human Factors Engineering Support for Nontactical ADP), under the sponsorship of the Naval Sea Systems Command (NSEA-61R2). The objective of the subproject is to provide human factors engineering support to the development of shipboard ADP systems for supply and administrative systems.

Appreciation is expressed to personnel of USS VINSON (CVN-70) and USS TRIPOLI (LPH-10), for providing access to their computer work stations and describing their collective experiences with shipboard computers.

J. W. RENARD
Commanding Officer

JAMES W. TWEEDDALE
Technical Director

SUMMARY

Problem

Numerous examples have been cited of deficiencies in the user-computer interface on Navy computers, both ashore and aboard ship. The computer system designer often overlooks the user's perspective in his desire to provide the user with a system that is a faster and more powerful tool. Some of the problems have been existent in the manual systems that have been automated; others are a result of new gadgetry hereto unknown to the operator.

Objective

The objective of this effort was to identify methods for improving the user-computer interface. This was done by reviewing the pertinent literature.

Results

1. Requirements of the personal computer user are identified and contrasted with the computer designer's viewpoint of the user.

2. The user's psychological needs are described so that the user-computer interface may be developed to accommodate them.

3. Two ideals of system design, transparency and visibility, are established to provide a reference for developing desirable dialogue principles.

4. Twenty-one dialogue principles, which were identified by surveying dialogue design studies, are listed.

5. Sources for work station design guidelines are discussed as well as some relevant variables that should be considered in the operator's physical environment.

Recommendations

1. Future study needs to be conducted to determine how to (a) aid the user instructionally, (b) use attentional devices to maintain operator alertness, and (c) develop compensational mechanisms for limited user short-term memory.

2. Various facets of menu selection methods (e.g., display formats, informational load per option, and the amount of user control over entry and exit from the menu display) need further explication.

3. The implementation process must be carefully planned, paying particular attention to pre-installation and initial operational activities.

CONTENTS

7

INTRODUCTION

Problem and Background

Deficiencies in User-Computer Interface

Numerous examples have been cited of deficiencies in the user-computer interface on Navy computers, both ashore and aboard ship (c.f. Moran, 1981). In each case, the user-computer interaction could have been designed differently to facilitate the user in accomplishing his task. However, to do this, the computer system designer must (1) know the user's goal and the task structure (i.e. the range of actions needed to reach the goal), (2) be aware of the user's knowledge of the task structure so that this knowledge can be exploited in reaching the goal via the computer, and (3) consider the user's information processing capabilities (i.e., memory and error propensity) in highly proceduralized repetitive tasks. The human factors engineer needs to influence the design of the interface either by changing the task structure or increasing the user's knowledge of it. The user's limitations may be compensated for by giving embedded training, providing efficient error recovery routines, or by breaking down user goals into easier, more attainable subgoals.

Designer Attitudes Versus User Psychology

The technological approach to computer science is that real progress is made by using smaller and faster computers without sacrificing storage capacity. Although numerous advanced technologies and devices have enhanced the user's capabilities, they have been accompanied by unique interactional problems. Some of the problems existed in the manual systems that have been automated. For example, if the manual system's documentation or operating instructions were inadequate or unclear from the beginning, automation will not make system operation any clearer or easier. Other problems result from new gadgetry hereto unknown to the user. For example, if information formerly contained on a teletypewriter printout is displayed on a CRT, new problems arise, such as how to format the information, how fast to present it, and determining who should control display refreshment (formerly page turning). Also, there are problems of screen glare, position, viewing angle, and height (i.e., work station design).

The designers' approach to solving these problems may be to try to be more considerate of the user in developing the system hardware and software. In doing this, they rely on their intuition in predicting what system features are necessary for operator efficiency, as well as on an anecdotal collection of experiences that may or may not lead to a well configured user-computer interface. However, this approach overlooks the user's perspective or behavior, which needs to be analyzed empirically if not systematically to improve the user-machine interface. This will assure a more reliable user interaction, just as hardware reliability has been studied and improved with new technology.

Moran (1981) defines the user's interface as any part of the computer system that the user comes in contact with--either physically, perceptually, or conceptually. Physically, the elements of the user's work environment impinge on him/her as mentioned previously--these are the factors of good ergonomic design. Perceptually, the user reacts to work station design features and informational content. Conceptually, the user may begin to function on the system at the procedural level and then gradually develop a model of the system, which may be refined as the user interacts with the system and experiences successes and failures. The conceptual model must be taught to the user and reinforced by the behavior of the system so that the user can achieve his/her goals. In his definition of the user interface, Moran views the computer as a tool--responsive, easy to wield, reliable, and capable of doing a bigger job. Control remains with the user. In contrast, Robertson, McCracken, and Newell (1981) view the computer as an intelligent agent, a problem analyzer, that produces results and explains them to the user.

Objective and Approach

The objective of this effort was to identify methods for improving the user-computer interface. This was done by reviewing pertinent literature.

RESULTS AND DISCUSSION

User Behavior and User Tasks

According to Moran (1981), user behavior is determined by the user's knowledge of the task structure and the task structure itself. The range of user knowledge of the task structure extends from that held by the naive user, through that held the novice, to that held by the "expert." The naive/novice user is sensitive to all variations in the task structure, while the expert is not affected by them. The novice finds every task a problem-solving exercise, while the expert finds the same tasks routine fare. The expert handles tasks quickly, while the novice aspires to complete them regardless of time. It is recognized that expertise is not a global concept and that user knowledge varies with the task structure.

The designer of the user-computer interface should not assume that the user possesses programming skills. To the contrary, the best interface may result if it is assumed that the user is naive (totally lacking experience with computers) but has normal (9th grade) reading ability. Shortcuts for the more "advanced" user should then be built into the dialogue to allow him or her to accomplish the task faster. Perhaps the most productive approach to the user interface design is that every system should have an instructional capability to assist the naive or slow-learning user while, at the same time, allowing for the expert to jump ahead in finishing the job.

The enhancement of the user's conceptual model of how the system works will facilitate his effectiveness in achieving his goal. It may be argued that it is only necessary for the user to be able to follow a set of procedures to perform certain jobs. It is true that all uninitiated users begin by using a stepping-through strategy to perform a task. However, as they become aware of the data storage entities and the internal movement of data between files, and recognize the physical counterparts that contain the operating software and the program applications, they will become more adept in troubleshooting. The instructional assists embedded in a system are a necessary prerequisite for user acceptance of the system and contribute toward the development of the user's system model.

Conceptual Model

It cannot be assumed that the user is a passive static being to be controlled and directed by the system. All actions of the system should be evaluated in terms of their effects on an actively changing user who is attempting to comprehend the system (Gaines, 1981). According to DuBoulay, O'Shea, and Monk (1981), the computer is an idealized, conceptual, "notional" machine whose properties are implied by the constructs in the programming language employed. The notional machine should be conceptually simple. Methods should be provided for the novice to observe certain of its workings in action. DuBoulay's notional machine is functionally simple--with simplicity being achieved by having a complicated program interpret the user's inputs. Robertson et al. (1981) emphasize that the system should be "transparent" to the user. The user should know why the system is doing what it is doing, and how to obtain more information from it or to get it to do something. He should feel that the system is completely controllable and nonmysterious. The user's conception of the system's transparency determines how he reacts to it. The Robertson et al. (1981) specifications for meeting the transparency requirement include the following features: menu selection, rapid response, large networking, and simple displays. These features create a structure that is simple in concept and completely under the user's control.

The transparency concept of the user interface is parallel to the Duboulay et al. (1981) notion of "visibility" where the hidden actions, such as storing a procedure, are concluded with a written comment from the system. Visibility means being able to see selected parts and processes of the computer system in action. System visibility can be increased by the use of mode lights, examinable code of standard subroutines, a series of steps to accomplish a procedure, and command language buttons to display the contents of the program counter, as well as by improving error message.

The user's conceptual model of the system, which tells the user how the system works and how it can be used to meet his goals, is an integral part of the user interface. The conceptual model must be developed for the user so that he can use it and be reinforced by the behavior of the system. The user's training and documentation should be keyed to development of a conceptual model of the system. Furthermore, the design of the user interface should be built around a conceptual model of the system. Codd (1974), as reported by Ehrenreich (in press), regarded the user's perception of the data base to be crucial in developing a query language system. He posits that the user's view of the data affects how he conceives and formulates queries and other types of transactions. The user's data model needs to be monolithic and should not have a multiplicity of structural alternatives for representing the data.

Dialogue Principles

Accepting the notions of transparency and visibility as ideals in the design of the user interface, several principles have been suggested for the development of user-system dialogue. These principles, which are listed below, mainly address the conceptual and some of the perceptual aspects of the user's interaction with the system. They deal with the language processing structure and dialogue development from the user's point of view or model of the system. Everything suggested as a dialogue principle in developing the user interface is in keeping with the notions of system transparency or visibility, which are the user's ideal view of the system.

1. Always inform the user of the irrecoverable consequences of a command and request confirmation from him or her. Similarly, ensure that the actual and the apparent penalty of making an error are not excessive. error messges should describe errors in terms of system components known to the user (Jones, 1978).

2. Use the user's model of the activity being undertaken and program the interactive dialogue as if it were a conversation between two users mutually accepting this model (Gaines, 1981).

3. Make the state of the dialogue observable by giving the user feedback--an immediate unambiguous response--to any of the user's inputs that may cause the dialogue to branch. The response should be sufficient to identify the type of activity taking place (Gaines, 1981).

4. Ensure that no selection by a user will produce a change that is irreversible (no "sudden death"). Where this is not possible, require an explicit confirmation from the operator (Robertson et al., 1981).

5. Always inform the user of the cost to him or her if the command will require an excessive amount of either time or money (Jones, 1978). Some way is needed to determine what "excessive" is for a particular user.

6. Avoid acausality by making the activity of the system a clear consequence of the user's actions (Gaines, 1981).

7. Ensure that all terminology and operational procedures are uniformly available and consistently applied.

8. Give users experience with interactive systems by getting them onto a terminal or a related or model system if their own is not yet available (Gaines, 1981).

9. Base user manuals on actual user dialogue. Illustrate the use of the system in action by showing actual dialogue sequences that achieve specific objectives (Gaines, 1981).

10. Ensure that the user is always able to return to known "anchor points" in the interaction. Anchor points should be dynamically determinable (i.e., back, mark, return, etc.) (Robertson et al., 1981). Provide a reset command that cleanly aborts the current activity back to a convenient checkpoint. The user should be able, at any stage in a transaction, to abort it cleanly with a system command that takes him back to a well-defined checkpoint as if the transaction had never been initiated (Gaines, 1981).

11. Provide a backtrack facility that allows a user to return through the dialogue sequence in reverse (Gaines, 1981).

12. Provide a set of standard options with standard names (edit, help, back, next, return, etc.) that are available in all displays (Robertson et al., 1981).

13. Allow the user maximum flexibility to make responses holistically (in parallel) or serially (in sequence) as desired (Gaines, 1981).

14. Distribute instructional aid appropriately throughout the dialogue system to be accessed by the user through a simple uniform mechanism (Gaines, 1981) or give aid whenever the system perceives that the user is in difficulty (Kennedy, 1974).

15. Ensure that the user can control the length of cues or error messages to suit his or her requirements (Kennedy, 1974).

16. Where entry commands require arguments, ensure that the user can enter them, either individually or in a string, depending on his or her level of ability (Kennedy, 1974). A program editor should be able to deal with individual lines within a data set (Miller & Thomas, Jr., 1977).

17. For novices, use "qualificational" languages (e.g., "Put the black ball in the box") rather than "conditional" languages (e.g., "If a ball is black, then put it in the box"). Miller (1975), as reported by Duboulay et al. (1981), found that novices were more at home with qualificational languages.

18. For inexperienced users, use functional, computer-oriented words (what the computer does) rather than operational words (common usage, no reference to computer) words for issuing commands (Scapin, 1981).

19. Use a keyword command argument format with permutable strings of special words since it has been found to be superior to a positional argument format. Weinberg (1971) found that user memory load was higher, as reflected by increased user error rates.

20. Be sure that a program editor provides for the following (Miller & Thomas, Jr., 1977):

a. Establishment of fields and for moving from field to field (e.g., via tab control).

b. Easy entry of full length records by the use of delineators.

c. Movement of groups of one or more lines or blocks of lines.

d. Line numbering, so there can be communication between processors (e.g. . . "ERROR IN LINE 43"), as well as local line-oriented editing

e. Special features (e.g., checking for parentheses balancing).

f. Formatting capabilities (e.g., indent, font, headings, margins, line-lengths, etc.).

g. Defaults between commands as a characteristic of the operating system (not as a special-purpose user program or macro).

h. Spacing by breaking up the text into logical segments. For example, space could be allocated by use of white horizontal bars produced by line feeds to separate segments of white vertical bars produced by indentation and tabulation to hold each segment together as exemplified by most newspapers and magazines.

21. Ensure that the user feels that his or her data are in safe hands (Jones, 1978)

Physical Aspect of User Interface

The physical aspect of the user interface is equally as important as the conceptual model. Traditionally, human engineers have studied the work place in terms of equipment and environmental design. Much of what has been written in classical human engineering guides is applicable to visual display terminal (VDT) design today. Cakir, Hart, and Stewart's Visual Display Terminals (1980), which covers the ergonomics, health, safety, and organizational aspects of working with VDTs, is one of the most recent and comprehensive manuals in this area. It contains a complete checklist that includes the specifications for the design of VDT equipment, work stations, and environmental conditions desirable for worksites.

Stammerjohn, Smith, and Cohen (1981), in a survey of five VDT work establishments, found excessive keyboard heights and screen positioning that required undesirable inclination of the head and neck for screen viewing. The majority of the operators they interviewed reported that screen readability, reflected glare, screen brightness, and flicker were bothersome factors. McCann (1978), who experimented with a number of graphical marker devices (i.e., the rolling ball, mouse, joystick, lightpen, knee control, and a touch display), found that very little human factors data existed for these devices. Smith (1981) and Gade, Fields, Maisano, Marshall, and Alderman (1981) concluded that light pen selection or entry methods of the "point-at" type, which are more accurate than the "type-in" variety, are good examples of spatial compatability. Parallax problems and definition of light sensitive areas for the light pen may make the touch display most attractive. These physical aspects of the user interface are very important. If they are not judiciously considered, their ill-effects will retard the conceptual/perceptual development of the user interface.

FUTURE INITIATIVES

Several issues need further investigation in the development of the user-computer interface. Some of these are more amenable to empirical study than are others. Research conducted in the laboratory is usually less generalizable to user settings than is research actually conducted in the natural or working environment. Conversely, it is difficult to find answers to questions studied in the real-world setting because of the lack of control over interfering situational variables. Once such question that may be raised is whether user acceptance of a computer system is the result of a well-designed user interface, or is it a prerequisite for a fair test and continued use of a well-designed user interface (Robinson, Malone & Obermayer, 1982). Whichever is the case, it is true that design never ceases. The interactive capabilities of the system close the adaptive loop between system and user through the designer (Gaines, 1981).

To build "user acceptance" of the system, the user should be provided with some form of computer-assisted learning involving the user's own language and his or her own current problem that is context-sensitive (Tagg, 1981). Miller (1979), as noted by Tagg (1981), says that what is required is "a tutor which gathers and maintains state information about each user and uses this information to both determine an optimal interface for a given user, and also to invoke any CAI, Help, or tutoring with adequate contextual knowledge." An interactive system should be capable of perceiving where help is required by the user. Errors need to be pinpointed, their causes diagnosed accurately, and corrective actions given promptly (Kennedy, 1974). The system needs to be response-sensitive (Atkinson, 1972) in establishing a trial-by-trial user history (Gade et al., 1981). When instructional aiding and system tutoring takes place as a secondary task to that which the user is attempting to accomplish, the training resources are said to be "embedded." This concept of CAI contrasts with the traditional notion of CAI as being the end result of the user interaction. Embedded training is a resource that aids the perceptual and conceptual development of the user interface addressed earlier.

Robertson et al. (1981) point to some major behavioral issues needing investigation that they discovered with their large network interactive system known as ZOG. First, they point out that users readily get lost. Often they do not know where they are, how to get where they want to go, or what to do. They feel lost and may take excessively long to respond. It may be that the users have not developed an accurate conception (theory) of how the system works and, if they have, it hasn't been confirmed or discounted. Not getting lost is a function of the system's properties of transparency and visibility discussed earlier. Also, it may be due to the user's inadequate conceptual development of how the system works. Both facets of the user interface--the provision of system visibility or transparency and the provision of an adequate user system concept--offer good areas for empirical research.

Second, users fail to read information on displays. Even though such information is in exactly the right form, users often miss it. The problem may be one of maintaining user attention to display after display, determining the best display formatting method, or changing the rate of information presentation--all of or a combination of which could be studied in a laboratory setting.

A third user limitation, according to Robertson et al. (1981), is the user's limited short-term memory. This problem may be related to the amount of information per display and the presentation rate. Bevan (1981) found that 10-15 characters per second (cps) was the most effective frame rate, and that 15 cps was the optimum speed. More realistically, system response (or presentation) rate should be variable. Kennedy (1974) found that a response within 2 seconds was acceptable. However, at the end of a sequence, the task is completed and a delay is satisfactory or even desirable to "savor the satisfaction derived from task closure." R. B. Miller (1968), noted by Miller and Thomas, Jr. (1977), suggests that maximums for system response times are a function of the type of user input (e.g., light per entries or a request for next page). He sees system response times increasing as a direct function of task complexity. "Locking out" the user for variable time periods may be useful for inducing concentration and compensating for short-term user memory. Embedded training with more summary and overview statements with overlapping from display to display should be investigated as a means for increasing user recall.

The menu selection method used as a central anchor point from which the user determines his courses of action on the system has been extolled by many (Gade et al., 1981; Robertson et al., 1981) as a user aid that compensates for limitations of user recall and as an orientation device to prevent the user from getting lost. Gade et al. (1981) found that providing a menu reduced input errors by 20 percent over the typing in of entries with an error correction capability. His investigators hypothesize that the menu not only aids the cognitive encoding of information but also reduces typographical entry errors. Their data led to the conclusion that menus are cognitively and behaviorally simpler than typing in entries. Robertson et al. (1981), from their ZOG experiences, point out that menu selection not only serves as a decision aid by eliminating search but also slows the sophisticated user by forcing him to read interposed explanatory text options. The ZOG people conclude that experts need "short-circuits" in their user interface; and novices, "long-circuits." The literature discusses menu selection in the user system dialogue in very global terms. One can get the impression that menus are an "open sesame" solution to most user interface problems. It would be interesting to study various characteristics of menu selection techniques such as the perceptions of display formats, presentation options, branching mechanisms, the information "chunk" size per option, and the amount of user control versus program control in entry to and exit from the menu display.

Having decided to use menu selection in the user-computer dialogue, numerous guidelines have been suggested in the literature regarding the use of this technique (Williges & Williges, 1981; Smith, 1982). For example, menu selections should be ordered in a list according to a logical structure. Options that are mutually exclusive should be grouped separately from options that are dependent upon one another. Related options should appear before specific options; however, if the list has no logical structure, then items should be ordered according to a ranking of their expected frequency of use. These researchers say the same rule applies to subunits of selection options. If frequency of use cannot be predicted for a list of items, then selections should be placed in alphabetical order.

The Williges' and Smith both point out that, if options are selected by entry codes, rather than by touch or voice, then the code associated with each option should be included on the display in a consistently identifiable manner. If menu selections are to be made by keyboard entry, usually the initial letter or first few letters of the displayed lable should be used rather than numeric codes. For example, "m" for move or "d" for delete or "del" for delete if the option list is very long. Smith (1982) notes that numbers, not letters or bullets, should be used to list all selectable options. Futhermore, menu numbers should begin with one, not zero, and a period should follow the number and the descriptor sentence. Finally, selection numbers should be justified from either right or left and at least one space should be used between the number and item descriptor.

Users may need to select their own dialogue features to function effectively on a system. Depending on the expertise of the user, he or she could select the appropriate dialogue features. An interesting study would be to compare a totally nonadaptive system to one where the operator selects dialogue features based on his/her perceived skill level. Another condition of the study could be the use of automatic program selection of dialogue features after user skill level has been assessed by the system. A fourth condition could be to compare nonadaptive user selection and automatic feature selection with a condition where the user and the program collaborate on whether user or system will decide who selects the dialogue features, thereby allowing the feature selection to be shifted between user and program. Task errors, time-on-task, and user preference could be measured to determine where dialogue feature selection should reside.

Also of promise for improving the user interface is the system's user documentation. Gaines (1981) and Robinson et al. (1982) stressed the importance of illustrating the use of the system in action by showing actual dialogue sequences that achieve specific objectives. User manuals are often cumbersome at best, let alone when written incorporating proven pedagogical techniques. Witness the dearth of well-written manuals for many consumer products including personal computing systems. Effective programmed text found useful in so many training applications could be developed for operator orientation. Not only are initial training documents frequently lacking, but follow-on reference aids are also in short supply. The technology of job performance aids (JPAs), portably packaged and amply illustrated, has found use in many of the settings. Fully proceduralized JPAs and partially proceduralized JPAs with judicious enrichment both may have potential in the development of the user interface.

Robertson (1981) has suggested that there is a relationship between the prescriptive instructional strategies of the component display theory (CDT), originated by Merrill (1981), for teaching a procedure and the ability of novice computer users to learn from embedded training within a computer system. Robinson (1981) points out that the literature (Merrill, 1981; Merrill, Reigeluth & Faust, 1979; Merril & Tennyson, 1978) indicates that using CDT instructional prescriptions result in significantly superior performance at the remember-level. Merrill proposes three performance levels: remember, use, and find. <u>Remember</u> is performance that requires searching of one's memory to reproduce or recognize some item of information that was previously stored. <u>Use</u> is performance that requires one to apply some abstraction to a specific case. <u>Find</u> is performance that requires one to derive or invent a new abstraction. In addition to performance levels, CDT prescribes instructional treatments based on various content categories and presentation forms. The important thing for user/system interface development is the potential that CDT holds as a user training resource, either embedded in the dialogues itself or used in adjunct user manuals or documentation.

Malone, Obermayer, Robinson, and Funk (1982), in the study of a data entry personnel system, concluded that, "a common but nevertheless important finding was that user acceptance is an enabling requirement for the design of human-computer systems. Without user acceptance, excellence of design in other areas can be a largely wasted effort." Many user interface designers would maintain that user acceptance is a result of a well designed human-computer system rather than a prerequisite for implementation success. A more useful research question may be what installation procedures should be practiced to ensure that a well designed human-computer system is operationally successful. Malone et al. (1982) assert that the operational environment should not be used to test marginal designs. As part of the total user interface development, the implementation process needs to be studied more carefully. It is quite likely that the best designed operator-computer system will not be accepted by users unless it is implemented properly. The implementation process needs to be studied in terms of the organizational dynamics of the system setting, the job(s) or tasks to be accomplished by the system, selection of operating personnel, personnel orientation and training, and follow-up troubleshooting of subsequent operations.

It is apparent that a well designed user-computer system cannot be dropped on the operating group without paying attention to preinstallation and initial operation activity. A very useful product needed by user interface designers would be an implementation guide for system installation. Many lists of criteria for designing the user-computer interface now exist, but guidelines for successful implementation are lacking. Lessons-learned, stated in terms of "pitfalls to be avoided," would be a welcome addition to the user interface designer's repertoire.

REFERENCES

Atkinson, R. C. Ingredients for a theory of instruction. American Psychologist, 1972, 27, 921-932.

Bevan, N. Is there an optimum speed for presenting text on a VDU? International Journal of Man-Machine Studies, 1981, 14, 59-76.

Cakir, A., Hart, D. J., and Stewart, T. F. M. Visual display terminals. Germany: John Wiley & Sons Ltd., 1980.

Dubouláy, B., O'Shea, and Monk, J. The black box inside the glass box: presenting computing concepts to novices. International Journal of Man-Machine Studies, 1981, 14, 237-249.

Ehrenreich, S. L. Query languages: Design recommendations derived from the human factors literature (ARI Tech. Rep. 484). Alexandria, VA: U.S. Army Research Institute (in press).

Gade, P. A., Field, A. F., Maisano, R. E., Marshall, C. F. and Alderman, I. N. Data entry performance as a function of method and instructional strategy. Human Factors, 1981, 23, 199-210.

Gaines, B. R. The technology of interaction dial...e programming rules. International Journal of Man-Machine Studies, 1981, 14, 133-15...

Jones, P. F. Four principles of man-computer dialog. IEEE Transactions on Professional Communication, 1978, PC-21, 154-159.

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

Malone, J. S., Obermayer, R. W., Robinson, E.R.N., and Funk, F. H. Accuracy, timeliness, and usability of experimental source data modules (NPRDC TR 83-1). San Diego: Navy Personnel Research and Development Center, November, 1982. (AD-A121 788)

McCann, C. Graphic display interaction, Part I (DCIEM Tech. Rep. 78X4). Downsview, Ontario, Canada: Defence and Civil Institute of Environmental Medicine, January 1978.

Merrill, M. D. Component Display Theory. In C. M. Reigeluth (Ed.), Instructional design theories and models. Hillsdale, N.J.: Lawrence Erlbaum, 1981.

Merrill, M. D., Reigeluth, C. M. and Faust, G. W. The instructional quality profile: curriculum evaluation and design tool. In H. F. O'Neal, Jr. (Ed.), Procedures for instructional systems development. New York: Academic Press, 1979.

Miller, L. A. and Thomas, Jr., J. C. Behavioral issues in the use of interactive systems. International Journal of Man-Machine Studies, 1977, 9, 509-536.

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

Robertson, G. McCracken, D. and Newell, A. The ZOG approach to man-machine communication. International Journal of Man-Machine Studies, 1981, 14, 461-488.

Robinson, E. R. N. A research design: Report #3 (Dissertation Proposal). Unpublished manuscript. Los Angeles: University of Southern California, School of Education, 1981.

Robinson, E. R. N., Malone, J. S. and Obermayer, R. W. Operator performance on two office data entry system testbeds: Preliminary analyses (NPRDC Spec. Rep. 82-16). San Diego: Navy Personnel Research and Development Center, February 1982. (AD A111 535)

Scapin, D. L. Computer commands in restricted natural language: Some aspects of memory and experience. Human Factors, 1981, 23, 365-375.

Smith, S. L. Exploring compatibility with words and pictures. Human Factors, 1981, 23, 305-315.

Smith, S. L. User-system interface design for computer-based information systems (ESD Tech. Rep. 82-132). Electronics Systems Division, Air Force Systems Command, United States Air Force, April 1982.

Stammerjohn, Jr., L. W., Smith, M. J. and Cohen, b. F. Evaluation of work station design factors in VDT operations. Human Factors, 1981, 23, 401-412.

Tagg, S. K. The user interface of the data analysis package: some lines of development. International Journal of Man-Machine Studies, 1981, 14, 297-316.

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

18