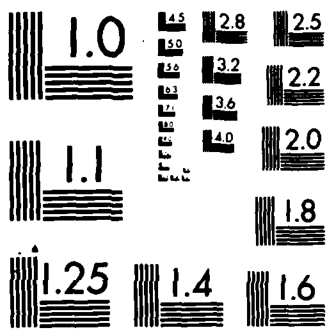
AD-A159 265 DESIGN CONSIDERATIONS FOR HUMAN-COMPUTER DIALOGUES(U) NAVY PERSONNEL RESEARCH AND DEVELOPMENT CENTER SAN DIEGO CA L R WILSON APR 77 NPRDC-TN-77-10 1/1

UNCLASSIFIED F/G 5/8 NL

END
FILMED
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(2)

NAVY PERSONNEL RESEARCH AND DEVELOPMENT CENTER SAN DIEGO, CALIFORNIA 92152

NPRDC TN 77-10

APRIL 1977

DESIGN CONSIDERATIONS FOR HUMAN-COMPUTER DIALOGUES

Technical Note 77-10 April 1977

DESIGN CONSIDERATIONS FOR HUMAN-COMPUTER DIALOGUES

Lance R. Wilson

Reviewed by
Joe Silverman

Navy Personnel Research and Development Center
San Diego, California 92152

FOREWORD

This research and development was conducted in support of Exploratory Development Task Area ZF55-521-010 (Manpower Management Decision Technology) under the sponsorship of the Bureau of Naval Personnel, Active Enlisted Plans Branch (Pers-212). This is one of a series of reports relating to Work Unit ZF55-521-010-03-11 (Management Decision Models).

J. J. CLARKIN
Commanding Officer

SUMMARY

Problem

Major policy and programming decisions in the area of manpower and personnel management are made under severe time constraints and with very limited amounts and kinds of information. Interactive management systems (IMSs) are being developed that provide a computational vehicle that may enable managers to make better use of their time and to consider problems in novel ways. However, the design of IMSs is a difficult and complex task due to problems in identifying, measuring, and relating diverse design considerations.

Objective

Most of the factors affecting the design of IMSs may be classified into one or more of the following categories: Environment, User, Hardware, Software, Models of the Problem, Data Base, Situations, and User-System Interface. The objective of this report is to reduce the scope of the design problem to a manageable proportion by focusing upon the user-system interface and presenting select criteria for the design and evaluation of an operational IMS.

Approach

The approach is to distill, from the available literature on man-computer dialogues, those criteria that will facilitate the design, implementation, utilization, and evaluation of conversational software for the user-system interface. The criteria refer to "rules of thumb," which focus upon the ideal design; that is, where cost, time, and programming efforts are not a major consideration. The criteria suggest the construction of the "best" possible user-system dialogue given the current state-of-the-art (hardware, software, and knowledge of dialogue requirements).

Findings

Thirty design criteria are presented as general normative statements which serve as guides to determine more specific and measurable design factors. These criteria are organized into six categories: (1) integration with current work habits, (2) training, (3) user input to the system, (4) user errors, (5) system output to the user, and (6) system processing.

Plans

1. Additions and revisions to the criteria will be accomplished through the design of a conversational software package for a model used in forecasting the basic pay of enlisted personnel.

2. Refine present criteria in order to obtain operational definitions and facilitate measurement.

3. Explore the utility of the criteria for comparing and evaluating dialogues.

CONTENTS

	Page
INTRODUCTION	1
Problem	1
Factors Affecting IMS Design	1
Assumptions Underlying IMS Design	3
Objective	4
APPROACH	5
DESIGN CRITERIA SELECTED	7
Integration with Current Work Habits	7
Training	8
User Input to the System	8
User Errors	10
System Output to the User	11
System Processing	12
PLANS	15
REFERENCES	17

INTRODUCTION

Problem

The effectiveness of Navy managers depends on the efficient use of information under conditions where time is limited. Interactive management systems (IMSs) provide a computational vehicle that may enable managers to make better use of their time and to consider problems in novel ways. Generally, IMSs consist of mathematical and statistical models of management processes, coupled with large data bases, and computer software that enables the manager to access the data base and manipulate models interactively.

Factors Affecting IMS Design

The design of interactive management systems is a difficult and complex task due to problems in identifying, measuring, and relating diverse design considerations. Most of the factors affecting the design of IMS may be classified into one or more of the following eight categories (see Figure 1):

1. Environment. The architecture of organizations, organizational behavior, organizational communication, and other factors that locate the IMS within an organization and, therefore, imply constraints on its design.

2. User. Psychological differences such as cognitive and perceptual attributes, information processing capabilities, and other variables relating to individuals (or a class of individuals) using the IMS.

3. Hardware. The capability, power, configuration, and operating characteristics of computers and peripheral equipment (including terminals, display devices, data storage media, etc.).

4. Software. The capability and characteristics of operating systems, general-purpose interactive software, data base management software, and other "enabling" programs.

5. Models of the problem. Statistical or mathematical models of management problems, particularly the form of the model and its underlying assumptions.

6. Data base. The structure, organization, and content of data relevant to the problem.

7. Situations. Related to traditional notions of problem or task, situations may or may not have solutions. There are different ways to proceed with a situation in order to understand its underlying structure. The IMS is created in order that the user may better understand and control the situation. Thus, the situation and its quantitative representation as a model are integral parts of the IMS.

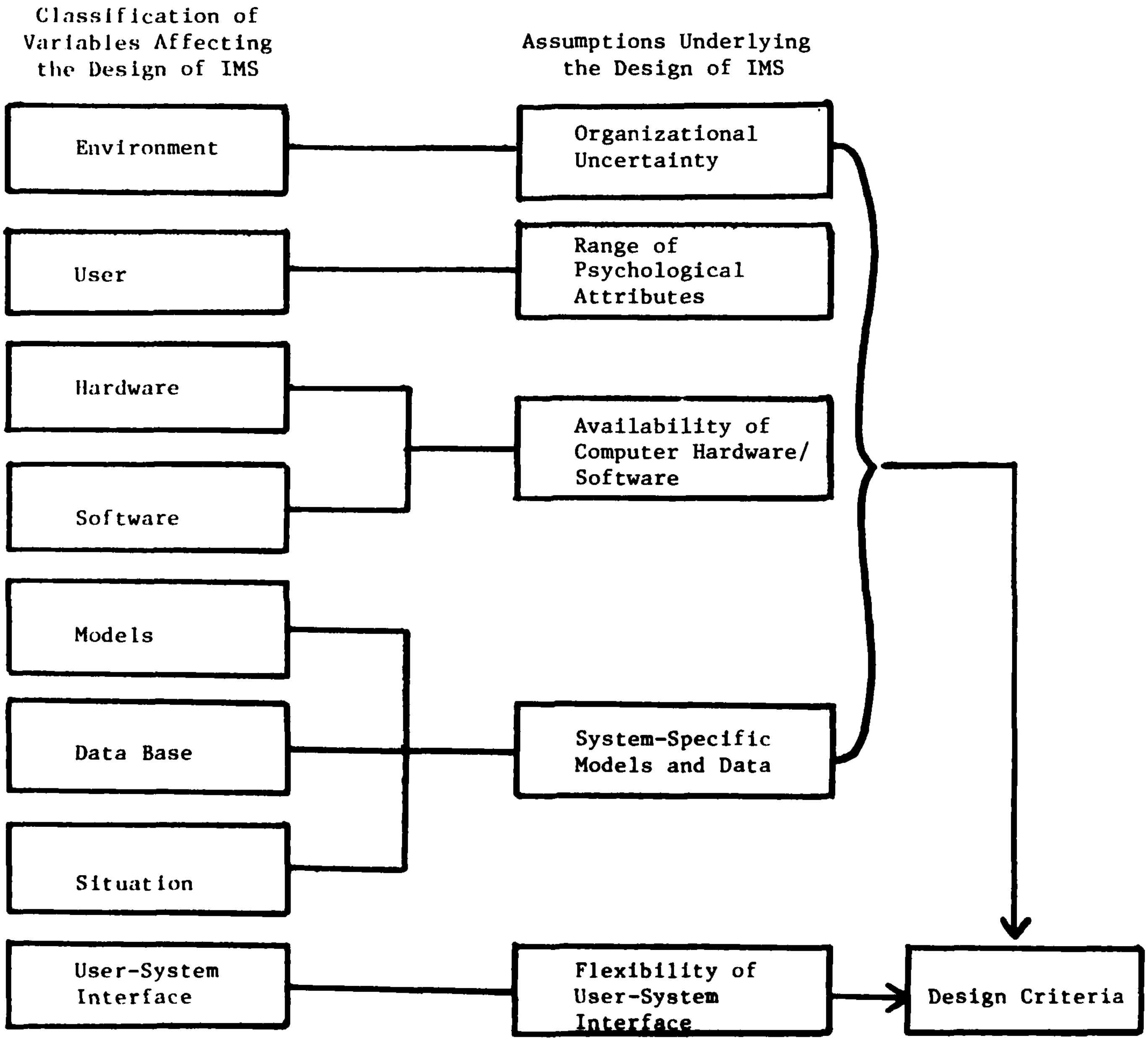


Figure 1. Variables and assumptions in the design of IMS.

8. User-system interface. The communication link between the user and hardware, software, models, and data bases. The user-system interface is the means by which user needs are conveyed to the system and the system communicates to the user. Of primary concern is the dialogue which occurs between the user and the other elements of the IMS.

Assumptions Underlying IMS Design

Due to the complexity of relationships and large number of variables in the eight categories, the development of operational systems requires some limitation on the number of variables explicitly considered in system design. To assist in this process of "scoping" interactive management systems, the assumptions described below have been considered. These assumptions also appear in Figure 1.

1. While the most immediate use of IMSs is in large organizations, the design of such systems can proceed without explicit consideration of specific variables related to organizational environment. The reason for this, at least in the Navy context, is that many of the organizational variables are unstable. For example, an IMS may be designed for a particular Navy organization but, by the time it is in operation, there may be changes in leadership, membership, organizational structure, goals, procedures, reporting requirements, etc. Further, an organizational subunit could be totally eliminated, with its functions performed by other new or existing organizations. Thus, it is desirable to develop an IMS that is robust in terms of possible organizational variation.

2. An IMS is usually required to function in ways that are relatively insensitive to individual differences among users. Psychological attributes of the user are variant and often unknown. Due to the large personnel turnover in Navy organizations, one cannot easily predict the psychological characteristics (cognitive, information processing, search behavior, or perceptual abilities) of the users. Thus, IMS must be designed to be insensitive to a broad range of individual differences in psychological characteristics.

3. In order to consider nontrivial problems in an operational environment, the availability of third-generation computing equipment, peripheral devices, time-sharing systems, and other computational accoutrements must be available. Similarly, the availability of relatively recent developments in data base management software, operating systems, and other software relevant to the construction and operation of IMS is assumed.

4. While most of the design criteria and some of the interactive modules can be "exported" to other applications, an operational IMS must be devoted to specific models and data bases to be effective. Accordingly, problem-specific capabilities are necessarily pursued at the expense of generality, when such design tradeoffs must be made.

5. Another limitation concerns the capabilities of the user-system interface or communication link. The interface must be capable of serving single or multiple users who may operate in serial or parallel manner. Use may be for extremely brief or extended periods of time. IMS capabilities

may be exhausted or barely exercised by individual users. User representational preferences may be graphic or numerical; computation preferences may be intuitional or analytic.

The purpose underlying the design of IMS in this report is that of problem revelation, as distinct from problem recognition, problem solution, or decision making. Thus, the IMSs are intended as systems for manipulating the structure of problems in order to gain understanding and insight and to enhance intuition. Unlike some experimental IMSs, the management task addressed here is not that of choice or decision, and IMSs are not intended as decision-aiding systems in the conventional sense.

Objective

The objective of this effort was to reduce the scope of the problem to manageable dimensions by focusing upon one aspect of system design--user-system interface. Other design considerations, such as user attributes, hardware configuration, software capability, etc. are excluded. This will enable us to focus attention on those criteria that facilitate the design, implementation, utilization, and evaluation of conversational software for the user-system interface.

APPROACH

The approach used in compiling and presenting these criteria involved a brief review of literature on the design of man-computer dialogues. The review considered the following sources dated from 1970 to the present: (1) IEEE Proceedings, (2) IEEE Transactions, Systems Cybernetics and Society, (3) IEEE Transactions, Man-Machine Systems, (4) Management Science, (5) Journal of Man-Machine Studies, (6) Annual Review of Information Sciences and Technology, and (7) James Martin's book, Design of Man-Computer Dialogues.

The criteria refer to "rules-of-thumb," which focus on the ideal design; that is, where cost, time, and programming efforts are not a major consideration. Thus, the criteria suggest the construction of the "best" possible user-system dialogue given the current state-of-the-art (hardware, software, and knowledge of dialogue requirements).

The criteria are derived from several sources, which reference both theory and practical experience (see, for example, Gaines (1975) and Kennedy (1974)). However, due to the lack of consistent or widely accepted theoretical models concerning man-system interaction, most of the considerations arise from practical experience and common sense. Moreover, since many ideas occur in several places in the literature, no specific citations are given regarding their source.

Extensive bibliographies for the user-system interface are presented in Bennett (1972), Martin (1973), and Rouse (1975). The latter is the most complete and up-to-date. These bibliographies will give the reader a good introduction to the content and direction of the field of man-computer interaction. Many other works could be cited, but they present duplicate or technological criteria which are obsolete.

The list of criteria is a fairly exhaustive summary of the information contained in the bibliography and is intended to be used as a baseline to add, delete, or revise based on experimental evidence.

Further work is necessary, not only in revising the content of the list, but also in operationalizing each criteria to permit measurement. The criteria are presented as general normative statements that can serve as guides to determine more specific and measureable design factors.

There are 30 criteria, which are loosely collected into six categories: (1) integration with current work habits, (2) training, (3) user input to the system, (4) user errors, (5) system output to the user, and (6) system processing. The categories represent no underlying reality; they were selected merely for purposes of organizing the presentation of the criteria.

Throughout this report, the terms "systems" or "interactive system" will refer to the hardware, software, models, and data base components of an IMS.

DESIGN CRITERIA SELECTED

Integration with Current Work Habits

The criteria in this category refer to the manner in which the system is presented to the user and the user's subsequent attitudes or beliefs about the system. The introduction of any new tool, whether a new office machine or a sophisticated management information system, requires integration into the user's daily "work life" in ways that are operationally simple and non-disruptive. In this regard, the following criteria apply.

1. Avoid unnecessary complexity in the user's "mental model" of the system. Each user will have a conception or idea of the IMS. The goal here is to keep the user's image of the system as simple and comprehensible as possible. The alternative, namely, the presentation of the system as an extremely complex entity, may negatively affect the user's desire or ability to use the system.

2. Present the system as a useful tool to the user. Although this is not strictly a dialogue design criteria, it is important that users develop positive attitudes (by way of examples, hints, or suggestions) and that the user believes the IMS has utility for his situation and problems.

3. Integrate and adapt the system to the user's work habits and work environment, The utilization of time by managers is often dictated by situational constraints (meetings, phone calls, and other interruptions). In the case of interruptions, the system should accommodate these constraints by enabling the user to take up where he left off. Thus, dialogues should be designed so that previous steps in the problem, which led to the present state, are reviewed without breaking the continuity (e.g., by requiring the reentry of all previous commands).

In this regard, the system must be able to store (on drum, disk, or tape) problem parameters needed to recreate the current state should the user leave the terminal for an extended period of time (several hours or days). Further, the user's responses and outputs should be stored for review upon returning to the terminal. This permits the user to refresh his memory concerning the previous work on the problem.

4. Program the human-computer interaction as though it were a conversation between two knowlegeable professionals discussing the problem. At the very least, the dialogue should be based upon terms familiar to the manager, proceeding in a manner consistent with conversation between a manager and a colleague, where the manager is seeking information and advice from the colleague.

5. Design the system so that it may grow with the user. The user may go through several stages of development, from the untrained beginner to the sophisticated expert. Further, users who have attained different levels of expertise and familiarity may use the same system. For example at 0900, a manager who has used the system several times and is very familiar with it may use the system for some specific purpose. At 1130, a manager, with a half hour for an uninterrupted lunch, may wish to use the system for the

first, second, or third time. Clearly the frequent and experienced user does not want a detailed description of the problem or an explanation for each step in the system process. Alternatively, the neophyte or infrequent user may desire and need as much explanation and assistance from the system as possible. Therefore, the user should be able to determine, at each major stage in the problem, the level of computer assistance desired (e.g., detailed error messages, textual explanation of each program phase, abbreviated input or output, etc.). In essence, the system should converse with the user at the user's level.

Training

There are several criteria that refer to the training a user must have to obtain the most benefit from the system each time it is used. As a user progresses from naivete to sophistication, he should be able to utilize more and more of the system's capabilities.

1. Avoid the need for a user to learn or use elaborate codes or sets of mnemonics. Codes and mnemonics are usually difficult and time consuming for managers to learn, and easy for them to forget. The interaction should be in as natural a language as possible. Abbreviated forms of commands may be used by experienced users. For example, P, PRT, or PRNT may be used instead of PRINT for a user command. However, the full natural language word should be printed by the computer. That is, the human may abbreviate his conversation but the computer may not abbreviate its communication to the user.

2. Distribute textual and tutorial information throughout the dialogue. Material to explain the use of the system, its algorithms, interpretation of outputs, etc. should be accessible throughout the interaction. If the user needs help or desires an explanation of some aspect of the program, he should be able to call for help (in a manner consistent throughout the interaction), to receive help, and to return to his problem without having to restart the system or retrace his steps.

3. Avoid the need for instruction manuals. To retain conversational fluency, the user should not have to refer to instruction manuals, lists of options, lists of codes, etc. If the user requires instruction, it should be displayed by the terminal, without changing the current problem status.

4. Before a user operates a system for the first time, lead him through a simple example problem demonstrating the system capabilities and familiarizing the user with the procedures and dialogue. The user obtains a "hands on" experience with the system rather than reading or listening to some other source. Thus, a simple example problem should be a part of the system's design, and the user should have the opportunity to go through the example as often as desired.

User Input to the System

Criteria in this section refer to the communication from the user to the system. Included here are the criteria for keyboard entry by the user.

1. Avoid input of long strings of numbers or information by the user. Most managers have neither the time, desire, nor typing skill to input strings of numbers. Manual input of data is usually time consuming, frustrating, and error prone. Much of the data required for a given problem is usually available in one form or another and should be placed on tape, disk, or drum for easy access and manipulation at appropriate times and places during on-line operation.

2. If the user must input short strings of numbers, avoid leading zeros, required blanks, and fixed formats. Fixed format inputs do not come naturally to managers who are unaccustomed to computer programming and usually cause input errors and frustration.

3. Keep entries required from the user short and concise to enable easy correction. The user should be required to reenter the command only in the event of error and that command should be short. Commands should be short one, two, or three word strings, not a line or two. These short commands are easy to read, locate, and correct if there is an error.

4. Use verbs for commands and nouns for labels. This short rule helps commands, arrays, programs, etc. from being confused with each other. Attempt to avoid words that may be interpreted as either nouns or verbs. Commands and labels should be precise and should relate, in a natural language sense, to the function they perform or the substance of the label.

5. Keep terminology and required responses consistent throughout the dialogue. All labels and commands should be applied in the same way throughout the interaction. For example, the request for printed output as PRINT in one routine should be the same throughout the program; not PRINT in one phase and OUTPUT in another. Similarly, required user response should be consistent. For example, the replies YES and NO should be standard throughout; not YES, NO for one phase of the dialogue and AFFIRMATIVE, NEGATIVE for another. If it is necessary to use numeric responses (e.g., 0=Yes, 1=No), these should also be consistent. So, if 0=Yes and 1=No are used in one phase of the dialogue, 0=PRINT, 1=PUNCH, 2=WRITE should not be used in another. One should, of course, try to avoid the latter formulation anyway.

6. Give the user a choice of responses appropriate for a given query and provide a means for him to request exact information. For example, in the following sequence, the user was unsure of the term "Title" and requested that the term by explained:

```
NAME:  JOHN DOE

TITLE:  ?

YOUR OPTIONS FOR TITLE ARE AS FOLLOWS:  MR, MRS, MISS,
  MS, ADM, CAPT, ETC

PLEASE ENTER TITLE FROM ABOVE LIST

TITLE:  CAPT
```

7. Print old values and new values before executing a change. The user should be shown the current values of the parameters to be changed so that he can compare them to the new ones. The sequence may occur as follows: (User response is underlined).

DO YOU WISH TO CHANGE BUDGET CONSTRAINTS FOR THE THREE PROGRAMS?

YES

WHICH PROGRAMS (1, 2, 3, ALL)?

ALL

CURRENT BUDGET CONSTRAINT VALUES ARE AS FOLLOWS:

PROGRAM 1	PROGRAM 2	PROGRAM 3
$520,000	$1,570,000	$875,000

PLEASE ENTER CHANGES

PROGRAM 1: 500,000

PROGRAM 2: 1,500,000

PROGRAM 3: 1,000,000

THE PREVIOUS AND CURRENT VALUES ARE PRESENTED BELOW

	PROGRAM 1	PROGRAM 2	PROGRAM 3
PREVIOUS	$520,000	$1,570,000	$ 875,000
CURRENT	$500,000	$1,500,000	$1,000,000

DO YOU WISH TO EXECUTE THESE CHANGES?

YES

CHANGES COMPLETED.

This sequence helps to ensure that the user clearly understands the values being changed, their relation to the old values, and the accuracy of the input.

User Errors

Most users will make either syntactical or logical errors while using the system. There are at least two criteria that suggest ways in which the system should respond to such errors.

1. Allow error correction without reentering the command string. When input errors occur, allow the user to recover by changing only the command in error. If one were required to reenter six commands just to correct one, the process is subject to further error (six chances to err rather than one), and it becomes frustrating and time consuming to reenter several commands.

2. Provide explanations of errors and guidance that are polite, detailed, and yet concise. In addition, the error messages should pinpoint the error

accurately and suggest appropriate corrective measures. Once again the user should be able to control the level of detail involved in error explanation and correction.

System Output to the User

Criteria listed here refer to the system's presentation of output to the user. Output may include tables, graphs, textual material, solicitations for user response, etc.

1. Display the user inputs and commands before execution. After the user has generated a series of inputs or commands for a given execution, the information should be displayed so that the user can validate the input. Further, the user should be allowed to change any information at this time without reentering the entire command or data string.

2. Provide immediate feedback or response to the user. Frustration and negative user attitudes usually result from situations where the user has to wait long periods of time for the computer to respond. The computer should respond immediately (within 1 to 3 seconds) when a command has been received and accepted for execution. If the execution of a command requires more than a few seconds of wait time, the system should periodically indicate that the computer is in execution, everything is going fine, and results will be along shortly. It has been observed that terminal speeds of 30 characters per second and greater are sufficient to eliminate much of the frustration due to slow printing. At greater speeds (e.g., 90 to 120 characters per second), some users may feel pressured by the computer.

3. Ensure that output terminology is consistent throughout the dialogue. All labels or other output material should retain the same designation throughout the interaction. For example, all files or processes should retain the same names throughout the interaction. This output consistency facilitates an understanding and ease of operation, especially for beginning users.

4. Do not introduce seemingly random phenomena. Make all changes in the system output or system action a direct, obvious, and visible consequence of a user action. If this is neither possible nor desirable, give a brief explanation for the change. For example, data may be displayed in an histograph form for one series of output and as a time graph for another. The reason for this change should be clearly stated (e.g., more data points, space constraints, or interpretability). Programmers often assume that several program changes or actions may arise from one user command, whereas the user may not. Therefore, all changes that are a consequence of a user or system command should be briefly explained. While the system may appear to be under user control at all times, this does not necessarily have to be the case.

5. Ensure that displays are user controlled. In using graphic output, the user should be able to define the level of aggregation or detail displayed. Since the output on a CRT is limited to a specific number of characters, the user may wish to review the same data series, graphic, or tabular output from different viewpoints or levels of levels of aggregation.

6. Enable user to interrupt long messages or printouts. Many times, especially in a tutorial or textual explanation, the user will understand the communication and not desire any more help or text. Also, the user may only be interested in a portion of the output for a given run. In each case, the user should be able to abort the unwanted remainder of the output and return control to the terminal. In this event, the total output would be saved and only the display would be aborted.

System Processing

The criteria presented in this section involve the processing capabilities of the system and the user's communication with these processes.

1. Use the computer to monitor and record the actions of the user and the system. User responses should be stored for at least two purposes: (a) research concerning user behavior can only be accomplished with accurate and precise information regarding user responses, and (b) to reiterate, the user may need to leave the system for an extended period and wish to restart where he left off, or at least review some of his previous actions and computer responses. He may also wish to review a decision sequence or results executed earlier, without using his own memory for recall. It may also be necessary to record system responses and output to avoid the expense and delays in recomputing system actions.

2. Enable the user to change the sequence of operations. If the user is allowed to temporarily skip some operation, the computer should remind the user of the skipped step and ask for the information before finally computing a solution. The change of sequence option may cause confusion and should be limited to users who are familiar with the system.

3. Enable the system to make default decisions. The major function of the system is to assemble and present relevant problem information to the user. In many cases, the user may not desire to make specific inputs and would be willing to accept default conditions. For example, the user may want the system to analyze time series data and to select a smoothing constant, or an appropriate rotation technique for factor analysis. The system should display the various options, state the one it will use, and ask the user if that is acceptable. In addition, the system should allow the user to override the default.

4. Determine where "lockouts" or delays should occur within the system. Bennett (1972) reports that, for some types of problems, significantly better results are obtained if the user is forced to "think about" or "look at" a solution or output before going on to another step. Whether this is desirable depends on the specific problem or situation under consideration. Perhaps an experienced or time-pressured user should have an option to override the delays and lockouts.

5. User should be able to abort system execution. If the user changes his mind or discovers a prior mistake while the system is executing a command, he should be able to tell the system to stop current action and perform an alternate action (e.g., restart the problem, change parameters,

etc.). The system should be able to display several recovery actions to the user, allowing him to reenter the system where he wishes, thus avoiding superfluous retracing of steps.

6. Enable user to generate, modify, create, or merge data arrays on line. The user should have easy on-line access to all relevant data and algorithms. Also, he should be able to avoid the system job control languages and be permitted to use natural, simple, and short commands to manipulate available data, as well as subroutines in some cases.

PLANS

Based on the criteria discussed above, preliminary plans for development of an IMS have been formulated. This application centers on the design of a conversational software package for a model used in forecasting the basic pay of enlisted personnel in the Navy. In the course of this application, an attempt will be made to add to, and revise, the IMS design criteria discussed previously. In addition, it will be necessary to refine present criteria to obtain operational definitions and facilitate measurement. Of particular interest, the utility of the criteria for comparing and evaluating dialogues will be explored. The intent of this research and development effort is to test the significance of many of the criteria for dialogue design under experimental conditions given specific problems.

REFERENCES

Bennett, J. L. The user interface in interactive systems. In C. Cuadra (Ed.), Annual review of information science and technology, 1972, Vol. 7.

Gaines, B. R. Some experience in interactive system development and application. Proceedings of the IEEE, June 1975, 63(6), 894-911.

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

Martin, J. Design of man-computer dialogues. Englewood Cliffs, NJ: Prentice-Hall Publishing Company, 1973.

Rouse, W. Design of man-computer interfaces for on-line interactive systems. Proceedings of the IEEE, June 1975, 63(6), 847-857.

END

FILMED

10-85

DTIC