# THE DESIGN, SIMULATION, AND EVALUATION OF A MENU DRIVEN USER INTERFACE

Ricky E. Savage, James K. Habinek, and Thomas W. Barnhart

IBM System Products Division
Rochester, Minnesota 55901

## INTRODUCTION

The design and evaluation of a menu-driven user interface for a general purpose system are described. Analysis of errors made by participants in simulation studies of the interface led to the development of hypotheses concerning user choice behavior. For example, novice users had difficulty selecting menu options based on job titles rather than tasks and functions. Redesign of the interface to reflect these hypotheses resulted in significantly improved performance. The current version of the interface appears to accommodate both the novice user, through an extensive hierarchy of menus, and the experienced user, through a variety of shortcuts to system functions.

Many practical alternatives for user-computer communication exist (Martin [5] Ramsey & Atwood [7], Shneiderman [8]). All have advantages and drawbacks, usually as a function of the characteristics of the typical operator. At one extreme, programming languages with precise syntax and vocabularies may be used for dialog between person and computer. These languages have the advantage that complex concepts can be communicated unambiguously by the operator. The associated drawback, is that the use of the language requires extensive training and strict adherence to rules.

At the other extreme, a hierarchy of detailed menus may be used for person-computer interaction. Since this technique relies heavily on the user's recognition memory and passive response to computer prompts, little formal training is required. As might be expected, menus have their drawbacks, too. For example, the creator of the menu must have a perception of all the possible or desirable options to include. Furthermore, use of the menus may become tedious if the choices are too finely detailed or if the user has extensive experience.

As the number of system users increases, the degree of formal training of the typical user

declines. Techniques such as menu selection, which can best accommodate the novice user, almost necessarily must be included in a strategy for person-computer communication. Yet care must be taken that the experienced or sophisticated user is not encumbered with an interface that involves frustratingly slow entry of commands or procedures. This paper details the process and techniques required to develop and test an interface that would satisfy the needs of a broad spectrum of users. Two design and evaluation iterations are described.

## DESIGN OF THE INTERFACE (PHASE I)

### Initial Human Factors Considerations

The primary consideration for designing this user interface was to provide an easy access to the entire system without constant reference to manuals. The design was targeted to assist the novice user, but at the same time not to penalize the experienced user. This second point is important, and to accomplish this goal, systems may have to be designed with multiple levels of user interface. General purpose systems normally have all levels of users. For experienced users, there would be a highly abbreviated and quick access to system functions. The novice user, on the other hand, would need a very specific step-by-step interface to lead him to the required system function. Both Shneiderman [8] and Martin [5] have stated that novice users normally require a menu driven interface.

A consideration in designing the interface was the structure of the menus. The hierarchical or tree structure was used (see Shneiderman [8] Chapter 7 for a brief discussion on data structure modes) because of the experimental evidence favoring such a structure (Brosey and Shneiderman [1]), and because of the natural hierarchical structure of this interface (Durding, Becker, and Gould [3]).

Another initial consideration was consistent screen design. Based on the work of Engel and Granda [4] and Peterson [6], screen standards were developed to ensure consistency. It was also determined that menus should have no more than nine options, and that the paths or levels of menus to a function should be relatively short, generally three or four levels.

For the less novice user, several shortcuts to commands and procedures exist. For example, any given menu may be displayed simply by entering its name. In addition, the user may directly enter a known command or procedure name to obtain a prompt screen for that command or procedure. Finally, the command or procedure name may be entered with its appropriate positional parameters to most directly accomplish a desired function.

## Implementation of the Simulation

Very little supporting software was required since most of the support was already available on the IBM System/34. The Source Entry Utility (SEU) and Screen Design Aid (SDA) were used extensively to create the menus and prompts. System/34 had some restrictions which prohibited running some commands from a menu. These restrictions were removed and code was added to process the command keys and chaining of menus. These operating system changes were primarily modifications to existing System/34 Assembler routines.

## EVALUATION OF THE USER INTERFACE (PHASE I)

## Tasks

With help from individuals with field experience, tasks were developed for programmers, system operators, and work station operators. Careful consideration was given to importance, frequency of use, and difficulty of the many possible functions before they were selected for use. Some examples of tasks were building libraries, copying files from diskette, changing printer IDs, printing information, and modifying source. Each participant had at least ten tasks to perform.

## Participants

Twelve participants performed the programmer tasks. Nine participants performed the system operator tasks and ten participants performed the work station operator tasks. These participants' experience ranged from total novice to operators with less than six months experience. All participants were employees of IBM Rochester.

## Equipment

The simulation of the user interface was developed to run on the System/34 which, in turn, became the vehicle for running the study. The system captured response and time on each menu for the protocol analysis. Video recorders and cameras were used as a backup and to record comments by the participants. Finally, an attitude questionnaire was administered.

## Procedure

Each participant operated the system individually for up to a two-hour session. General instructions were given and the participants were allowed to ask questions before continuing without experimenter assistance. Participants read the task instructions and used the menus to accomplish their tasks. Manuals were available through the session. The participants chose options from a series of menus to lead them to a command, procedure, or utility. Parameters were then entered on a prompt screen from which the function would execute and accomplish the task. If participants could not accomplish a particular task, they were allowed to continue with the next task.

## Data Analysis

The analysis of the user interface posed special problems because the evaluation was not an experimental comparison of alternatives. The analysis procedures had to be sensitive in finding problems, such as the wording of the menus, the hierarchical structure of the menus, and organization of a specific menu path or a specific menu, and to be sensitive in describing user behavior, particularly user errors.

Time was recorded for each menu and each task. The primary benefit of measuring time was to serve as a baseline for future evaluation (as a result of modifications to the interface) and to determine if an unusually large amount of time was spent on a particular menu or task.

The protocol analysis consisted of mapping each subject's responses on paper for a comparison with the optimal path, which was defined as the shortest route to the desired function. The error analysis consisted of categorizing user errors (defined as an incorrect menu option chosen or an incorrect parameter specified) into types of errors. A probability analysis was performed by regarding each menu as a decision point, and determining the probability of a correct decision. The probability analysis pointed to specific menus which were not communicating adequately.

## Results and Discussion

The error analysis found four general categories of errors. The first category was called an "inconvenience error" and resulted from three different actions: 1) the user taking the wrong path but ending up at the correct function; 2) the user searching or exploring various menu options and paths and eventually taking the correct path to the correction function; and 3) the user searching or exploring and eventually taking the wrong path but ending up at the correct function. These inconvenience errors were not serious errors, but the user took a less than optimal path to the correct function.

The second category was called a "path error." Three different actions could cause this error: 1) the user taking the wrong path to the wrong function; 2) the user searching or exploring and taking the wrong path to the wrong function; and 3) the user taking the correct path but on the last menu selecting the incorrect option which led to the wrong function. The path

errors were serious because the user ended up using the incorrect command or procedure.

A "function error" was the third category and resulted when the user filled in the wrong parameters on the prompt screen when they had successfully reached the correct function. This error did not indicate problems with the menus, but did point out problems the user had with the prompts.

Table 1 shows the percentage of errors for each category and for each type of task. The inconvenience errors were relatively consistent for all three types of tasks and they seemed to be due to unclear wording of the menu options and some misinterpretation of how the task should be accomplished. The path errors were rather high for system and work station operator tasks. Some of these errors will be explained in the discussion of the probability analysis. The function errors for system and work station operator tasks seemed to be due to a lack of experience with the command and procedure prompts and a lack of help text explaining the parameters. Finally, searching and exploring various menu options by the users accounted for most of the inconvenience and path errors. The absence of detailed help-text for each menu option in the simulation may have been responsible for much of the user confusion.

Table 1--The Percentage of Errors for Each Category

|  | Inconv errors | Path errors | Function errors |
|---|---|---|---|
| Programming tasks | 61.1 | 29.6 | 9.3 |
| System operator tasks | 33.1 | 36.5 | 30.3 |
| Work station operator tasks | 25.8 | 61.6 | 12.6 |

The probability analysis was successful in pointing to menus with a low probability of correct option selection. In general, these problem menus were of two types. In the first type, one or possibly two incorrect alternatives had a high probability of selection relative to the correct alternative. These errors appeared to result from a discrimination problem. In the other type, participants used a shotgun approach: a large variety of incorrect alternatives was selected by the participants in lieu of the correct option. In this case, the users seemed to have little notion as to which alternative was correct. The path errors for the system and work station operator tasks illustrate this. These two groups of participants were constantly confusing each other's options from the first menu. At the next level of menus, the shotgun approach resulted.

The results of the questionnaire administered to the participants showed that the wording of the menus was the primary problem. Generally, the participants felt that the interface was easy to use, helped them to learn about the system, and would aid in their productivity.

One of the primary results from this evaluation was the high success rate: programmer tasks, 92%; system operator tasks, 79%; and work station operator tasks, 81%. Another major result was that the participants did not use the manuals. These results suggested that a menu driven user interface offers a viable method of assisting users in their work. On the other hand, the fact that failures did occur, and that frequently many false starts and backtracking were required before acceptable solutions were found, suggested that the interface needed some redesign.

REDESIGN AND REEVALUATION (PHASE II)

Many of the changes to the interface involved breaking up and rewording complex and cluttered menus to make them simpler in appearance and to reduce the information overload. As discussed earlier, two menu paths had to be redesigned which affected some of the structure of the menu hierarchy. These paths required users to choose options which corresponded to their job: "work station tasks" or "system console tasks." The results from Phase I clearly showed that users were not inclined to use the menus in this fashion. They looked for task oriented options rather than job classification options. The redesign eliminated the two job classification options and developed four task oriented options. The purpose of the Phase II evaluation was to test the success of these changes.

Method

Twenty people participated in the second phase of testing. Six performed the programmer tasks, and seven performed the system console operator and the work station operator tasks. The tasks, procedures, and equipment were identical to Phase I.

Results and Discussions

The second phase of testing the interface was completed with significant improvements in user performance. Table 2 shows the time and success rate for the three groups of participants, comparing the result of the first and second phases of testing. As can be seen, significant improvements were obtained for both types of operators in terms of the average time to complete a task and the rate of successful completion of the tasks.

The results of the error analysis are presented in Table 3. Clearly the path errors are the most serious, and they were significantly reduced during the second phase of testing for both types of operators. These two tables demonstrate the success of the changes that were

made to help the operators. Finally, the attitudinal questionnaire produced similar results to Phase I, with the wording of the menus again being the major problem.

Table 2--Average time to complete each task and the success rate of both phases of testing

| Task Type | Test Phase | Time (Mins) | Success |
|---|---|---|---|
| Programmer | 1 | 11.53 | 92% |
|  | 2 | 12.85 | 100% |
| System console | 1 | 5.41 | 79% |
|  | 2 | 1.69 | 95% |
| Work station | 1 | 6.30 | 81% |
|  | 2 | 2.92 | 95% |

Table 3--Average errors per participant by error category for both phases of testing

| Task Type | Test Phase | Inconvenience | Path | Function |
|---|---|---|---|---|
| Programmer | 1 | 6.00 | 1.60 | 2.20 |
|  | 2 | 7.30 | 2.00 | .83 |
| System console | 1 | 4.80 | 5.30 | 4.40 |
|  | 2 | 3.71 | .71 | .29 |
| Work station | 1 | 4.33 | 10.33 | 2.11 |
|  | 2 | 5.71 | .43 | 1.14 |

The probability analysis found several problems, but they were minor compared to those in the first test phase. These problems consisted of some menu options being misleading, missing functions, wording problems, and discrimination problems. Generally, all of these problems had obvious solutions and were easily correctable.

CONCLUSION

Many of the problems found in these evaluations were due to ambiguous terminology. Participants did not know the difference between work station, display station, and device, nor did they understand the difference between saving and copying a file, or removing and deleting a file. This problem needs to be solved not only with simpler terminology, but also with help text for each menu to explain in more detail what each particular menu option means.

Two results from these evaluations contributed significantly to a better understanding of user behavior with menus. First, it was evident from these studies that a user's job title was not important, whereas tasks and functions were important when developing menus. This was demonstrated by the success of the changes made from Phase I. Second, it appears that users in these studies preferred shorter menus with more levels than the opposite case. Many of the changes from Phase I consisted of breaking up a complex menu into two menus. The second study showed the success of these changes. This result is in apparent conflict with a recent study by Dray, Ogden, and Vestewig [2]. These differences may be attributed to the realism of this interface (in that it dealt with an actual system) and to the fact that menu options were word phrases as opposed to one or two words in the Dray, et al [2] study.

The basic concept of this menu driven user interface is clearly sound. Comments received from participants were generally favorable and performance reasonably successful. Any future changes made to the interface will be more "fine-tuning" than "major overhaul."

REFERENCES

1 Brosey, M. K. and Shneiderman, B. Two experimental comparisons of relational and hierarchical data base models. International Journal of Man-Machine Studies, 1978, 10, 625-637.

2 Dray, S. M., Ogden, W. G., and Vestewig, R. E. Measuring performance with a menu-selection human-computer interface. Proceedings of the 25th Annual Meeting of the Human Factors Society, 1981.

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

4 Engel, S. E. and Granda, R. E. Guidelines for man/display interfaces. IBM Poughkeepsie Laboratory Technical Report TR00.2720, December 19, 1975.

5   Martin, J.   Design of Man-Computer Dialogues.
        Englewood Cliffs, NJ: Prentice Hall, 1973.

6   Peterson, D. E.   Screen design guidelines.
        Small Systems World, February  1979, 19-21,
        34-37.

7   Ramsey, H. R. and Atwood, M. E.   Human factors
        in   computer   systems:   a   review   of   the
        literature.   Englewood, CO:   Science
        Application,  Inc.,  Technical  Report SAI-
        79-111-DEN, September, 1979.

8   Shneiderman, B.   Software Psychology.
        Cambridge, MA: Winthrop Publishers, Inc.,
        1980.