1987

AN EXPERIMENTAL STUDY OF USER SATISFACTION WITH MULTIPLE DIALOG MODES

Roderick A. Taylor
Air Force Flight Test Center

Michael S. Y. Wang
The University of Texas at Austin

AN EXPERIMENTAL STUDY OF USER SATISFACTION WITH MULTIPLE DIALOG MODES

Roderick A. Taylor
Major, USAF
Air Force Flight Test Center

Michael S. Y. Wang
Department of Management Science and Information Systems
The University of Texas at Austin

ABSTRACT

This paper investigated a normative theory that says computer users have different dialog needs depending on their level of experience in using a computer. It hypothesizes that the answer to satisfy the needs of a mixed population is to have multiple dialog modes that the user is free to choose from and switch between as required.

The hypotheses that experts and novices would be more satisfied with multiple dialog modes than with just one mode were tested empirically in a controlled laboratory setting. Both novice and expert computer users used one of three types of user-system interfaces (menu, command language, or both modes) to solve the same database problem. Results showed that those with both types of dialog modes were more satisfied and performed better than the command language group. However, they were statistically equal to the menu group, while the menu group's satisfaction rating and performance scores were slightly better. It was concluded that the subject's choice of dialog mode, when both modes were available, and their satisfaction with a dialog mode have more to do with past experience and preference than with the difference in expert and novice problem solving strategies.

INTRODUCTION

All computer systems, no matter what type, require a user-system interface to facilitate communication between the user and the computer system. The interface can range from simple to complex. It can range from setting hardware toggles, to punched cards, to real-time interaction via a terminal. This mandatory interface is necessary for data transfer in or out of the computer, processing instructions, starting/stopping a system or a combination of these plus others. The point is, humans must be provided with a way to communicate with computer systems to accomplish a task and that method is through an interface.

With the advent of interactive systems followed by mini-computers and the micro revolution, emphasis on the user-system interface has grown tremendously. These systems and the current state-of-the-art in hardware has opened the door for a tremendous variety of users who have a wide range of experience, preferences and task requirements. Because of this, providing just one dialog mode for all these different user experiences has the basic underlying problem that all the users, regardless of their experience with computers or the problem task, will have to learn/conform to the selected dialog mode if they want to use the system.

One of the problems with varied user experience is that experts and novices use different problem solving techniques to complete tasks (Simon 1984; Larkin et al. 1980; Card, Moran and Newell 1983). They have different amounts of knowledge and experience to apply to accomplishing the task and approach the task differently; i.e., novices are subgoal oriented versus experts who are goal oriented and can seemingly apply a compiled process to the solution. Card, Moran and Newell (1983), in an in-depth experimental study on experts and novices performing text editing with a word

processor, found that the novices would rather use several basic steps to perform an editing task (use of subgoals) compared to the expert's desire to use one specific command (a compiled process) to perform the same editing task. Applying this to a user-system interface, and dialog modes specifically, it seems that novices would be more inclined to prefer and actually perform better if they used a dialog mode consistent with their problem solving strategy such as can be done with menus. Menus are a form of subgoal processing since each menu is in itself a subgoal that leads the user to the final solution (goal). This same reasoning applies to experts as well except that they could use a command language to accomplish the goal directly. With menus, the experts can become frustrated with the method (a subgoal process oriented to novice style of problem solving) for accomplishing the task seemingly because that is not their problem solving process (Stevens 1983).

This need to accommodate different levels of user experience, especially between experts and novices, has also been consistently written about in a normative manner and tested experimentally by a number of researchers from a variety of disciplines such as cognitive psychology, information systems, and computer science. In general, the normative literature is primarily based on opinions and experience and concerns itself with the problem of novice users having different user-system interface needs than do experts and that these needs are many times opposite. They note that without accommodating the different needs between novices and experts (or experienced versus inexperienced), one or both groups will be faced with having to use interface functions or modes that are inappropriate for their level of experience. This inappropriateness could inhibit the user from maximizing both performance on the problem task and satisfaction with the interface. Experimental research on user-system interfaces have usually dealt with some underlying concept in relationship to novices only or with experts versus novices. Few experiments have tried to collect evidence for the need to provide multiple dialog modes for accommodating the differences in user experiences; those that dealt directly with this issue did not provide any real support for this need (Hauptmann and Green 1983; Whiteside et al. 1985). In each case though, the experiment had some internal validity problems or confounding variables that cause one to be suspicious of the results. There are also a few other experiments that do provide support for multiple dialogs but the data was collected as a by-product of other research and was not the primary focus (Hiltz 1984; Gilfoil 1984; Benbasat, Dexter and Masulis 1981; Mozeico 1982).

The conclusion to be drawn was that a well designed, controlled experiment that alleviates the problems in the previous experiments was still needed to test whether multiple dialog modes are better than a single dialog mode when there is a range of user experience in the population. Thus, the main thrust of this research was to try and provide this empirical evidence. In particular, the research was designed to determine, for the specific experimental setting, (1) whether the use of multiple dialog modes can accommodate the differences in user experience and result in higher user performance and satisfaction than an interface with only one dialog mode and (2) does user experience level make a difference in the user's task performance and satisfaction for different types of dialog modes?

INTERFACE DEFINITION AND COMPONENTS

To properly discuss the implications of the user's performance (effectiveness and efficiency in solving a problem) and satisfaction (perceived usefulness of an interface to facilitate the solving of a problem) with a user-system interface, it is important to know the different components of an interface so that each piece can be analyzed as to its effects on user performance and satisfaction and provide a foundation for further discussion of user-system interfaces.

Based on the work of Smith (1980) and Benbasat, Dexter and Masulis (1981), the interface can be conceptually viewed as having four components (Figure 1).

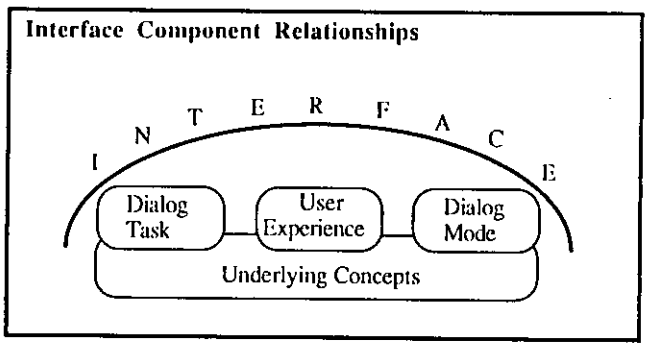

Figure 1. Interface Component Relationships

Dialog Task: This term reflects the generic lower level task operations a computer user would perform or need to communicate while working within an application or problem domain. These generic tasks are such things as data entry, text editing, query retrieval and process control. This is not an exhaustive list but represents the major types of activities that a computer user would perform when interacting with an application to accomplish some task.

User Experience: These are categories in which one must consider the experience of a user.

a. Technology: experience with computers in general regardless of the problem domain.

b. Problem Domain: experience in some field without regard to computer technology.

c. Application Specific: experience with computers in a specific problem domain.

d. General: general knowledge acquired through life's experiences.

Dialog Mode: This is the vehicle for representing the communication between the user and the computer. The representation used must be able to convey the meaning and the intent of both the computer's and the user's requests and actions.

There are several different dialog modes that can be used in an interface. The two most often used, and the two that will be studied in this research, are menus and command languages. Some of the other dialog modes are natural language, question and answer, and forms.

a. Menu: This is probably the most popular form of communication with an application. Selected menu items may perform an action or generate another menu for eliciting additional information. Menus have the advantage of restricting the domain presented to the user which allows a high degree of control by the application especially for error determination and correction. Interfaces that use this type of dialog mode generally tend to control the flow of communications. When this control occurs they are called **computer directed dialogs.**

b. Command Language: This dialog mode generally consists of verb - noun pairs followed by options. This mode usually puts the user in control of the flow of dialog and is an example of **user directed dialog.** While this mode is generally harder to learn and can require extensive training, it provides the user the freedom to control the interaction and not be restricted to a particular subdomain.

Underlying Concepts: This is the foundation on which an interface is based. It consists of functions that implement procedural and conceptual requirements that have been developed through experience and human factors research and tend to be applicable to all user-system interfaces regardless of the type of dialog mode used, the dialog task performed or the user's experience. Some examples are consistency, feedback, help and error messages. While these underlying concepts generally transcend dialog mode, dialog task, and user experience, the other three interface components none the less have an impact on the degree to which the underlying concept is implemented in an interface.

EXPERIMENTAL DESIGN

The selection of the independent and dependent variables for this experiment are supported by the prior work of Benbasat, Dexter and Masulis (1981), who established a model for conducting user-system research by listing a number of independent variables by category. Specifically, the independent variables were the dialog modes (from the interface characteristics category) used to solve a problem and a subject's technology experience (from the human user category).

The dialog mode variable consists of three treatments a) menu only (computer directed), b) command language only (user directed) and c) both menu and command language together (called "both"). The technology experience variable has two classes determined by a pre-survey of potential subjects a) novice and b) experienced (hereafter called experts in this research). To ensure internal validity for this experiment, the other three categories of user experience (discussed earlier) were also controlled. Problem domain experience was held constant by selecting an easily understood problem with which all subjects would be familiar (updating an address book). General experience was assumed to be constant since all subjects were university students and college graduates. Application specific experience could not be assumed constant or ignored as with the other two cate-

gories without posing possible internal validity problems. Thus, technology was selected as the independent variable but all subjects were measured on their application experience to be investigated as a covariant in the experiment. Application experience was also used to further divide the sample population for random assignment to the treatment groups.

The dependent variables for this experiment are performance effectiveness/efficiency and user satisfaction with the dialog mode(s) used to do the problems. Performance was measured by: (a) time to complete the problem (efficiency), (b) quantitative score reflecting correctness of required actions (effectiveness), and (c) percent menu used versus command language when a choice was available. The satisfaction measurement was accomplished with a semantic differential post-survey given to each subject at the conclusion of the experiment. The survey is based on similar survey instruments developed and tested by Zoltan and Chapanis (1982), Magers (1983), Hauptmann and Green (1983), and Kerber (1983).

The experimental design consisted of three treatment groups with a blocking factor of two based on experience for a total of six different experimental cells. Each subject was assigned to only one cell and was required to perform two specific problems consisting of multiple steps or items.

To control for extraneous and confounding variables: a) the same two problems were used for every treatment group, b) the problems were accomplished using a software package that has the facility to provide multiple dialog modes (this eliminated discrepancies in application performance and response times that could occur if different packages were used), c) every group used the same type of hardware and operating system, and d) every group performed the problems in the same setting. The following section describes the details of the experiment.

Task Design: The specific application used for the experiment was dBASE III. It was selected over other applications because of its rich command language, the ability to develop menus and the ability to surreptitiously generate a log of all console actions and system responses. The two dialog modes tested were a restricted subset of the dBASE III command language and a menu system that was developed using the dBASE III macro and screen generation facility. The menu system was built using proven design techniques (Foley and Sibert 1983) and contained a sufficient set of the underlying concepts described earlier as necessary for any type of interface dialog mode. Extensive care was taken in developing the menu interface to ensure that it was not excessively better or worse (in terms of human factors issues and underlying concepts) than the built-in dBASE III command language. The capabilities of the two dialog modes were made equivalent so that anything that could be done with the restricted subset of commands from the command language could be done with the menu system and vice versa. To align the two modes even more, the menu system presented and asked for information using terms and formats consistent with the command language. Only a few of the top level menus were task specific. All of this was done to ensure that there would not be an inherent bias toward one or the other of the modes. Pilot testing of the system indicated that this was successfully accomplished.

Subjects: Subjects were primarily undergraduate and graduate students from the University of Texas College and Graduate School of Business. All subjects that volunteered completed a Computer Technology Experience Survey that was used to categorize potential subjects as experts or novices in computer technology and as expert, novice or no experience with the dBASE III system. The primary factors used to determine a subject's experience level were: a) course work, b) work experience, c) programming languages used, d) hardware used, and e) types of applications used. Based on the computer technology survey, subjects were assigned to either the novice or expert group and a dBASE experience level for a total of six groups. They were then randomly selected from these two groups to one of the three treatment groups without knowledge of their experience rating or treatment to be received. Figure 2 summarizes the specifics on the distribution of subjects to each treatment based on their technology and dBASE experience level.

At the beginning of each experimental session, subjects were given a thirty minute training session on the particular dialog mode(s) that they would be using. This training was done in a group setting using an overhead projector and handouts with a short hands-on session prior to beginning the first problem.

Subject Assignment

User Experience Level: technology - application	Dialog Mode: Menu	Cmnd Lang	Menu & Cmnd Lang	Total
Expert-E	6	3	6	15
Expert-N	3	4	5	12
Expert-Z	4	4	3	11
Total	13	11	14	38
Novice-E	1	1	1	3
Novice-N	13	13	12	38
Novice-Z	6	7	6	19
Total	20	21	19	60
			Total	98

Legend:
E=Expert in dBASE N=Novice in dBASE Z=No experience in dBASE

Figure 2. Subject Assignment

Problem Domain: The problem task the subjects were asked to perform consisted of updating an address book and an associated Christmas card list. The problems themselves consisted of fourteen numbered narrative items (eight in problem one and six in problem two) that required the use of different dialog tasks such as adding, deleting and modifying data records, querying database, and generating reports. This mixture of dialog tasks ensured that the problems were not biased toward a particular dialog mode because only one dialog task was used. This type of problem was selected to ensure that all of the subjects would have the same problem domain knowledge and thus eliminate this variability. The experimental problems were split into two 45 minute parts so that the experiment would not be perceived as too long and so that there would be a sense of closure (Shneiderman 1980) on finishing one part before going on to the next. The time limit was imposed so subjects would have some pressure to perform the items not only correctly but in a timely manner.

When each subject finished the second problem or when the time limit was reached, they were given a Satisfaction Survey that contained seven semantic differential pairs of words from which a satisfaction score was derived. The first six pairs are easy-to-use/hard-to-use, frustrating/comfortable, simple/complicated, hard-to-learn/easy-to-learn, confusing/obvious, satisfying/dissatisfying. The last pair of contrasting words were in a question form that asked for their satisfaction level with the interface mode used in the problem: satisfactory/unsatisfactory. Other questions also were asked to gather information on their thoughts on the experiment itself and interface dialog modes in general.

Data Gathering: To determine the performance measurement variable, each subject's actions and the system's responses were collected in a log file where a quantitative score was determined for each problem by analyzing each item in the problems and assigning it a score based on a scale of 0 to 4 where a 0 was given for not even attempting to do an item and a 4 was awarded for a completely correct action/response.

The efficiency variable was determined by time stamps that were put into the log by the recording function. There was a time score generated for each problem (though not for each item) with the maximum value being the 45 minute problem time limit. The time score was then divided by the number of items in the problem with a score greater than 1 to generate an average time per problem.

The satisfaction variable was generated by adding up the score for each of the seven items used to elicit the subject's satisfaction then dividing by seven to get a score that ranged from 1 (extremely satisfied) to 7 (extremely dissatisfied).

Statistics: The performance and satisfaction variables were analyzed using an analysis of variance for unbalanced cells. Both the performance and satisfaction models were also analyzed using dBASE experience as a covariant to determine the effects of application specific experience on these variables. Other statistics such as comparison tests, correlations and means were used as needed to investigate each of the hypotheses.

RESULTS

The expected result from this experiment was that the subject population would be more satisfied with the user-system interface that had multiple dialog modes where there was a specific case of a user directed mode (the command language) and other for a computer directed mode (the menus). This was based on the theory that the experts and novices would be able to use the dialog mode that matches their problem solving method and not be mismatched (thus causing dissatisfaction). The specific

hypotheses developed to test for this result are listed below. H1 is used to see if there is an effect on satisfaction when a user-system interface has multiple dialog modes. H2 through H4 test whether or not multiple dialog modes cause greater satisfaction for different segments of the population.

Hypothesis 1

$H1_0$: An interface's dialog modes has no effect on a user population's satisfaction with the interface.

$H1_A$: An interface's dialog modes has an effect on a user population's satisfaction with the interface.

Hypothesis 2

$H2_0$: Subjects who have multiple dialog modes were equally or less satisfied with the user-system interface than users who only have either a computer directed or user directed dialog mode.

$H2_A$: Subjects who have multiple dialog modes are more satisfied with the user-system interface than users who only have either a computer directed or user directed dialog mode.

Hypothesis 3

$H3_0$: Experts who have multiple dialog modes were equally or less satisfied with the user-system interface than experts who only have either a computer directed or user directed dialog mode.

$H3_A$: Experts who have multiple dialog modes are more satisfied with the user-system interface than experts who only have either a computer directed or user directed dialog mode.

Hypothesis 4

$H4_0$: Novices who have multiple dialog modes were equally or less satisfied with the user-system interface than novices who only have either a computer directed or user directed dialog mode.

$H4_A$: Novices who have multiple dialog modes are more satisfied with the user-system interface than experts who only have either a computer directed or user directed dialog mode.

In all of the charts and figures presented in this section, the lower the mean score for a population the more satisfied the population is with a dialog mode interface. The scale is based on a seven point differential with 1 = extremely satisfied, 4 = neutral and 7 = extremely dissatisfied.

The mean scores obtained for satisfaction are shown in Figure 3. The data is presented so that bars that go up from the neutral response (a mean of 4) represent increasing satisfaction while bars that go down represent decreasing satisfaction (dissatisfaction). The data clearly show a difference in satisfaction levels between the treatments. In general, the command language satisfaction is in the dissatisfaction direction while the other modes are in the satisfaction direction.

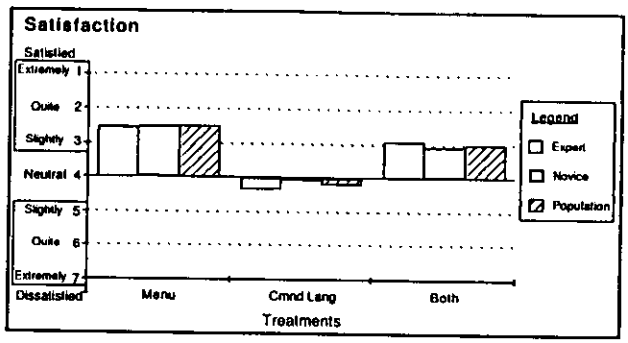

Figure 3. Satisfaction

Testing of H1: The results of the analysis (see Table 1) show that the only significance ($p < .0001$) was on treatment. The null hypothesis can then be rejected in favor of the alternate; dialog mode did make a difference in satisfaction.

Testing of H2: A comparison test of the population means (see Table 2) shows a very significant difference in the satisfaction level between command language and the other two ($p < .001$), even though the population satisfaction mean for command language, 4.169, is just barely in the dissatisfaction direction (only .169). Subjects in the other two treatments indicated satisfaction with the interface from "slightly satisfied" for "both" to almost "quite satisfied" for menu. Since subjects were more satisfied with menus, the null hypothesis

cannot be rejected. A comparison test to see if the difference was significant in favor of menu resulted in a weak significance at only the .1 level ($p < .10$).

Table 1. Dialog Mode Satisfaction Model

Dialog Mode Satisfaction Model				
Source	Df	SS	F Value	p>F
Model	5	50.44	8.64	<.0001
Error	92	107.40	N/A	N/A
TECHEXP	1	0.08	0.07	<.7875
TREAT	2	48.05	20.58	<.0001
TECHEXPxTREAT	2	0.57	0.24	<.7833
MSE = 1.167 R^2 = .32				

Table 2. Dialog Mode Satisfaction Treatment Comparisons

Dialog Mode SatisfactionTreatmentComparisons						
	Comparisons (a)	(b)	Means (a)	(b)	F Value	p>F
Total Pop.	"Both"	Menu	2.960	2.465	3.11	<.10
	"Both"	Cmnd Lang	2.960	4.169	20.69	<.001
	Menu	Cmnd Lang	2.465	4.169	38.92	<.001
Experts	"Both"	Menu	2.881	2.506	0.810	>.10.
	"Both"	Cmnd Lang	2.881	4.333	11.131	<.005
	Menu	Cmnd Lang	2.506	4.333	17.036	<.001
Novices	"Both"	Menu	3.018	2.438	2.808	<.10
	"Both"	Cmnd Lang	3.018	4.083	9.706	<.005
	Menu	Cmnd Lang	2.438	4.083	23.770	<.001
MSE = 1.167 F(.01) = 6.96 F(.05) = 3.96 Df = 1/92						

Testing of H3: The means for experts, as depicted in Figure 3 shows that experts were more satisfied with menus than either the command language dialog mode or the interface with both types of dialog modes. This allows for the acceptance of the null hypothesis without any further analysis. Experts were not more satisfied with multiple dialog modes than with just one dialog mode: menu. The analysis in Table 2 shows that while menus were more satisfying than "both," there was no statistical difference between them. It is also surprising to see that experts on the whole were basically neutral (just barely not satisfied) with the command language interface. This is not the expected result based on the literature.

Testing of H4: It was predicted that even though novices are supposed to prefer menus, the interface with both menus and command language would still be more satisfying since the novice was provided a choice and not forced to use either mode. This turned out not to be the case (see Table 2). For novices, the menu dialog mode was more satisfying than the other two although weakly for "both" ($p < .10$); thus, the analysis requires the acceptance of the null hypothesis. In the case of novices, as with experts, "both" was significantly more satisfying than command language.

DISCUSSION

In the next few paragraphs, a number of the important findings derived from the data are presented. Conclusions that can be drawn from the findings are presented in the last section.

Novice versus Expert: None of the groups tested, experts, novices or the combined population, statistically showed the "both" method to be better than either of the other two although the subjects in the "both" treatment indicated satisfaction with the mode. It also showed that within treatment, there was very little difference between the experts' and the novices' satisfaction. Novices held true to the literature in regard to menu versus command language but experts did not. The experts were also more satisfied with menus than command language.

To see if this result was true for all classes of experts and novices, the satisfaction scores were graphed by dBASE experience and treatment as shown in Figure 4 and a co-variant analysis of satisfaction using dBASE experience was done. The co-variant analysis (SS = 2.234, F = 1.914, $p > .10$) had no significance. The graph of satisfaction by dBASE experience showed that for menu and "both," all classes of dBASE experienced users were indeed in concert.

For command language, an unusual phenomenon occurred. Those experts and novices who had never used dBASE before (N/Z and E/Z) were actually satisfied with the interface while those who had used dBASE before were not, especially the E/N group. A review of the comments written in the post-survey by subjects in the three dissatisfied

groups revealed that they were bothered not only by the syntax but also by not knowing when to use which command. In other words, they had a problem remembering the language. It was observed during the experiment that the E/N and E/E subjects tended to rely on trying to "remember" the syntax and command use rather than take the time to use the handouts. On the other hand, the E/Z and N/Z subjects, who only had a brief introduction to the commands, used the handouts regularly (presumably because they did not know the syntax).

When expert versus novice problem solving is considered, subjects in the "both" treatment do not select the dialog mode predicted by their problem solving strategies. Novices in technology selected both menu and command language, when given their choice, without regard to previous experience with dBASE. The same held true for experts. It was also found that, even when experts used the menu user-system interface (which favors a novice mode of problem solving), they performed better than those who used the command language. The experimental results did not provide evidence to support the difference in expert and novice problem solving techniques as the basis for why experts and novices need different types of user-system interfaces.

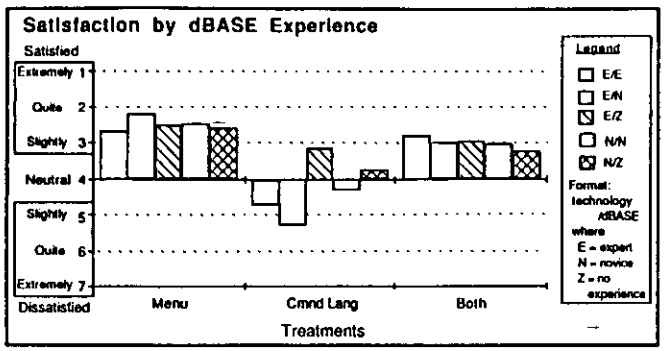

Figure 4. Satisfaction by dBASE Experience

Satisfaction versus Preference: One underlying assumption of the satisfaction measure in this experiment is that it is somewhat of a surrogate for preference. That is, if subjects prefer menus to command language, their satisfaction should be higher in the menu and "both" treatment where they can choose menus than those subjects in the command language treatment that did not have access to the menus. To investigate this, subjects were asked in the post survey to state their preference between using a menu or a command language dialog mode. Figure 5 shows the results with 1=strongly prefer menu, 4=neutral, and 7=strongly prefer command language. In comparing satisfaction (see Figure 4) to preference, all of the groups in the menu treatment were satisfied with menus and all groups preferred menus except E/N. Note that the E/E and E/Z type of experts in the menu treatment preferred menus for a dialog mode interface which is contrary to the literature. For the "both" treatment, every group except E/E preferred menus and were able to choose them if desired. The command language treatment group had both E/E and E/N dissatisfied with the command language but they also preferred to use a command language for the dialog mode interface. Thus, while they were unhappy with the dBASE language, they still wanted a command language (presumably with some other characteristics than were available in dBASE). In summary, subject satisfaction with a dialog mode did not necessarily follow what was predicted based on the literature and theory. It was also found that satisfaction and dissatisfaction with a dialog mode does not imply preference.

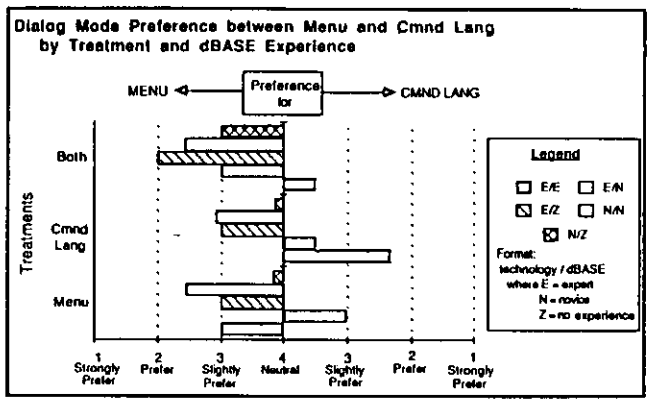

Figure 5. Dialog Mode Preference between Menu and Cmnd Lang by Treatment and dBASE Experience

Menu versus Command Language -- Preference of Choice: The subjects in the "both" treatment were analyzed further by how much of each dialog mode they used in accomplishing the problems. To determine this, the subjects in the "both" treatment were broken out by how much of each dialog mode they used to solve the problems. This was done by determining what percent of the problem was done

using command language by looking at how each item in the problems was completed.

The subjects in the "both" treatment were divided into expert and novice and then again subdivided into those who used the command language less than 25% of the time to do the problems (e.g., used menus more that 75% of the time) and those who used command language more than 47% of the time (there were no subjects between 25% and 47%). The latter group is labeled "both"/command language (BC) and the former "both"/menu (BM). Once this separation was done, each classification of groups of subjects in the "both" treatment, from the total population down to the technology/dBASE experience grouping, were looked at to see what percentage of each group fell into the BM and BC category (see Figure 6). From this analysis, one sees that the groups were all basically around a 60/40 split of BM to BC respectfully except for E/Z all of whom used the menus exclusively. Of the total population of 21 subjects that used menus more than 75% of the time, half of them used menus exclusively. For the BC population, three of the twelve used the command language exclusively. The 33 subjects that had the user-system interface with both menus and command language had a dialog mode usage that ranged from 100% menu to 100% command language with all sorts of combinations between.

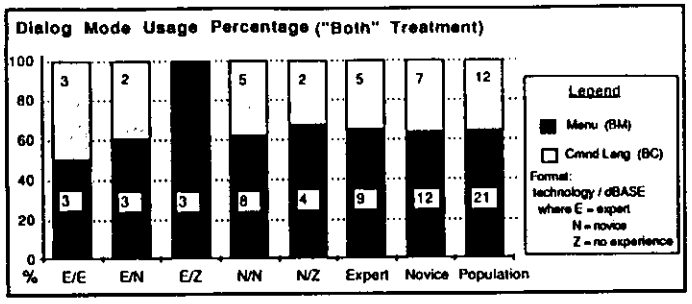

Figure 6. Dialog Mode Usage Percentage ("Both" Treatment)

What must be carefully pointed out, though, is that the command language users were not exclusively the dBASE or technology experienced subjects. The choice of what mode to use did not seem to be based on expert/novice differences as the theory and literature suggested. The choice seemed to be more of a preference based on not only experience, but ease of using the dialog mode, the specific item to be done, curiosity (one E/E subject said that she initially used menus just to see how good they are and not necessarily because of preference or satisfaction requirements), the mental effort required to do the problem, learning and especially remembering (memory and recall).

Voluntary versus Mandatory Use of Dialog Mode: A more detailed look at satisfaction shows, in Figure 7, that the satisfaction for menu users in the menu and the "both" treatments had comparable results with no significant differences. For command language (Figure 8) there was a large difference in experts' satisfaction level with those who used the command language in the "both" treatment being much more satisfied than those who used it in the command language treatment. Thus for command language, satisfaction was improved in all cases, except N/Z, when the subject chose command language rather than being forced to use it. In Figure 8, there were no E/Z subjects in the BC population thus no score is listed.

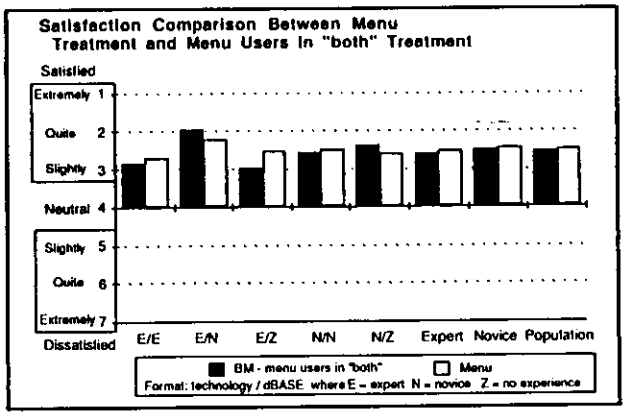

Figure 7. Satisfaction Comparison between Menu Treatment and Menu Users in "both" Treatment

User Response Time: Analysis of the experiment suggests that one reason for this has to do with the response time of the menu system. An exceptionally fast, dedicated micro system that displayed the full screen menus was used for each subject. This speed of processing is one of the activities prized by many users, especially experienced users. Speed of processing is also one of the main advantages of command language, e.g., one does not have to progress through many levels of menus to perform an action. Thus, for experts, the speed of

the menu seemed to have been sufficiently fast that they were satisfied. If this same system had been implemented on a mainframe computer where there was a definite wait time for each menu to be displayed (especially on a terminal using 1200 baud), the experts' satisfaction (and maybe even the novices') may have been much lower and narrowed the satisfaction difference between command language and menus. It might also have caused more of the experts in the "both" treatment to use the command language. Further analysis in this area is needed before any specific conclusion can be generalized beyond the micros used in this experiment but it does seem that micros do not support the theory. It may also be that experts are not necessarily dissatisfied with computer-directed interfaces, just dissatisfied with poor design and response times for menus. On mainframes, one way for experts to correct for slow response with menus is to use command language, since no delay is generated waiting for the menu to be presented.

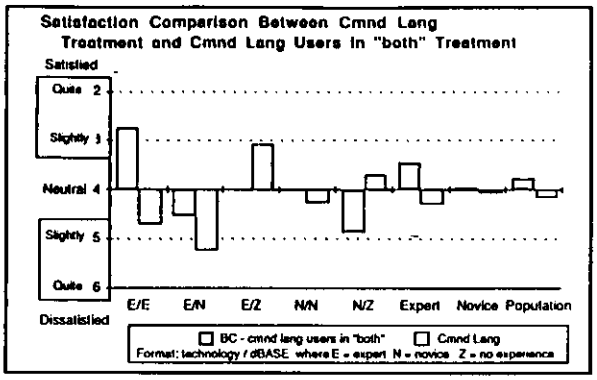

Figure 8. Satisfaction Comparison between Cmnd Lang Treatment and Cmnd Lang Users in "both" Treatment

Preference for Multiple Dialog Modes: This experiment showed that the dialog mode could make a difference in a user's satisfaction. It also showed that the subjects' choice of which dialog mode to use was not dependent on the subjects' technology experience level and associated problem solving strategy but more on a preference for one or the other of the dialog modes for the problem requirements. Experience with dBASE III had both a positive and negative influence on performance and satisfaction for all classes of users for a variety of different reasons. When asked about their preference for only one mode versus having both modes, the vast majority of the subjects expressed a strong preference for a user-system interface with both dialog modes (see Figure 9), even though they were most satisfied with menus and performed best with menus. Those subjects that were in the treatment that provided them this preference for multiple dialog modes and freely made their choice did not exceed the scores for the subjects forced to use menus. Thus, having their preference did not improve satisfaction and performance.

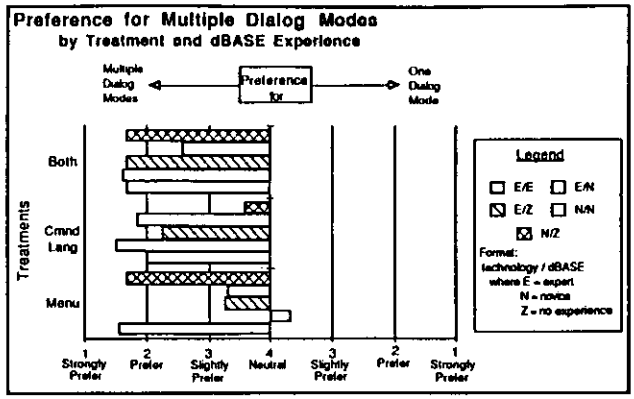

Figure 9. Preference for Multiple Dialog Modes by Treatment and dBASE Experience

Implications: The results of this experiment provide situational information that can have an impact on a user population's satisfaction and performance with a user-system interface. In particular, the results clearly show that, between the dialog modes used in the research, the menu mode is an excellent choice for novices just as has been found with other experiments. Novices performed best and were most satisfied with the menus. More importantly though, it was found that experts in technology also did best with menus. This was especially true for those that were novices to the dBASE command language or had fallen back to casual user status. This indicates that for applications that have periodic database activities and are to be used primarily by casual and novice users to the application (regardless of their technology experience), a well structured menu system that provides all of the necessary functionality and utilizes current human factors concepts will allow the user population to perform better and be more satisfied than with a command language dialog. Having only a command language can actually lead to dissatisfaction and poorer performance with the application.

Thus, under these conditions, there does not seem to be a need for having a user directed as well as a computer directed mode, although it should not be cast aside without due consideration, since there is a strong preference for both modes and the performance and satisfaction were basically equal.

There are two other important items that have implications for user-system interfaces. The first is that having multiple dialog modes does have the advantage of providing the user with alternative methods for accomplishing the same activity. It allows the user to "choose" which method to use under which conditions based not on technology experience but on some other set of experiences and preferences.

The second item is that if the user population is a mix of user experience for an application and a command language is required, then better performance and satisfaction will be obtained from the casual and novice users if a menu system is also included in the user-system interface. To make this true though, the user-system developer must use design principles found in the literature and ensure that the response time of the menus is similar to that achieved with the command language.

CONCLUSIONS

The hypotheses for this experiment were based on a normative theory developed from the literature and past experiments. This normative theory said that computer directed user-system interfaces (e.g., menus) were best for novices and that they would be satisfied and perform well with this type of interface. The theory also said that experienced users would not be satisfied with a computer directed interface because they would want to be more in control and thus would do better if they had a user-directed interface such as command language. The underlying basis for this normative theory was conjectured to be grounded in the theory developed around the differences in expert and novice problem solving strategies. These strategies seemed to be closely aligned to user directed and computer directed styles of dialog modes. From all of this information, it was hypothesized that the reason a user-system interface with a user directed dialog mode seemed best for experts was because it lent itself to the expert's problem solving strategy. The same is true for novices with a computer directed dialog mode. Thus, if there was a mixed population of expert and novice users, then a user-system interface with both a user and a computer directed dialog mode would provide more satisfaction and facilitate better performance than just one dialog mode that all users would be forced to use.

In this experimental problem setting of database update, query, and report generation, the results of this experiment did not support the normative theory or the expert versus novice problem solving theoretical foundation; thus, most of the hypotheses were rejected. Specifically, experts were not dissatisfied with a menu user-system interface and performed quite well with menus. Many experts were in fact dissatisfied with the command language, even those with extensive dBASE experience, while some novices were not dissatisfied. Most significantly, when the experts and novices were given the opportunity to choose their dialog mode, as in the "both" treatment, 60% chose menus with the remaining 40% choosing command language in both the expert and novice populations. What this experiment indicates is that there are other reasons for a user's choice and satisfaction in using a particular type of interface than those presented in prior literature and that expert versus novice problem solving strategies may not be the underlying basis for this choice.

A key result in this experiment that casts doubt on the developed normative theory is the universal lack of dissatisfaction among the subjects with the computer directed menu dialog mode. Experts, even those with dBASE experience, and novices were both satisfied with the menu system. Lack of experimental support for this normative theory indicates that perhaps the normative theory no longer applies especially for settings similar to this experiment (e.g., database activities).

Overall, the research indicates that the differences in expert and novice problem solving strategies did not seem to impact dialog mode usage and satisfaction as predicted by the theory and literature. Some reasons for this may be that the dialog mode was possibly at too low a level of thought processing to impact on the problem solving tasks required in the experiment or there just isn't a correlation between expert versus novice problem solving strategies and user-system computer interfaces. The important considerations in using a dialog mode seem to be based more on previous experience with dialog modes, ease of using the dialog mode for the problem, speed, and a bias

either for or against a specific dialog mode or dialog style based on a subject's own unique set of preferences and general experiences.

There are many other research topics that can build on this and other research. The issue of which type of user-system interface is best for what type of user in what type of situation has not been solved but progress is being made. Use of good experimental design and techniques and the use of established frameworks (such as the one presented by Benbasat, Dexter and Masulis 1981) to guide the research will definitely enhance our ability to further our knowledge of this extremely important facet of human computer interaction.

REFERENCES

Benbasat, I.; Dexter, A. S.; and Masulis, P. "An Experimental Study of the Human/Computer Interface." *Communications of the ACM*, Vol. 24, No. 11, November 1981, pp. 752-762.

Card, S. K.; Moran, T. P.; and Newell, A. *The Psychology of Human-Computer Interaction*, Lawrence Erlbaum Associates, Publishers, Hillsdale, NJ, 1983.

Chapanis, A. "Computers and the Common Man." *Information Technology and Psychology*, Praeger, 1982, pp. 106-132.

Foley, J., and Sibert, J. "How to Design User-Computer Interfaces." Tutorial 1, *Proceedings CHI '83 Human Factors in Computing Systems*, Boston, December 1983.

Gilfoil, D. M. "Tracking Cognitive Learning and Dialogue Preference in Naive Computer Users A Longitudinal Study." Unpublished dissertation, Stevens Institute of Technology, 1984.

Hauptmann, A. G., and Green, B. F. "A Comparison of Command, Menu-Selection and Natural-Language Computer Programs." *Behavior & Information Technology*, Vol. 2, No. 2, April-June 1983, pp. 163-178.

Hiltz, S. R. *Online Communities*. Ablex Publishing Co., Norwood, NJ, 1984.

Kerber, K. W. "Attitudes Towards Specific Users of the Computer. Quantitative, Decision Making and Record-Keeping Applications." *Behavior & Information Technology*, Vol. 2, No. 2, April-June 1983, pp. 187-209.

Larkin, J.; McDermott, J.; Simon, D. P.; and Simon, H. A. "Expert and Novice Performance in Solving Physics Problems." *Science*, Vol. 208, January 1980, pp. 1335-1342.

Magers, C. S. "An Experimental Evaluation of On-line Help for Non-Programmers." *Proceedings CHI '83 Human Factors in Computing Systems*, Boston, December 1983, pp. 277-281.

Mozeico, H. "A Human/Computer Interface to Accommodate User Learning Stages." *Communications of the ACM*, Vol. 25, No. 2, February 1982, pp. 100-104.

Shneiderman, B. *Software Psychology: Human Factors in Computers and Information Systems*. Winthrop Publishers, Inc., 1980.

Simon, H. A. *The Sciences of the Artificial*. The MIT Press, Cambridge, MA, 1984.

Smith, H. T. "Human-Computer Communication." In H. T. Smithland and T. R. G. Green (eds), *Human Interaction with Computers*, Academic Press, London, 1980, pp. 2-38.

Stevens, G. C. "User-Friendly Computer Systems? A Critical Examination of the Concept." *Behavior & Information Technology*, Vol. 2, No. 1, January-March 1983, pp. 3-16.

Whiteside, J.; Jones, S.; Levy, P. S.; and Wixon, D. "User Performance with Command, Menu, and Iconic Interfaces." *Proceedings CHI '85 Human Factors in Computing Systems*, San Francisco, April 1985, pp. 185-191.

Zoltan, E., and Chapanis A., "What Do Professional Persons Think About Computers?" *Behavior and Information Technology*, Vol. 1, No. 1, January-March 1982, pp. 55-68.