This is to certify that the

dissertation entitled

the effect of locus of control on

performance with man-machine dialogues

presented by

Richard L. Hartley

has been accepted towards fulfillment
of the requirements for

PhD. degree in Education

Major professor

Date May 18, 1983

THE EFFECT OF LOCUS OF CONTROL ON PERFORMANCE WITH MAN-MACHINE DIALOGUES

by

Richard L. Hartley

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

College of Education

1983

ABSTRACT

THE EFFECT OF LOCUS OF CONTROL ON
PERFORMANCE WITH MAN-MACHINE DIALOGUES

By

Richard L. Hartley

This study focuses on the theory that performance in a man-machine interaction is based not only on cognitive skills and short term memory capacity but is also affected by anxiety. The anxiety of concern is that which might be produced when individuals with specific personality attributes use specific types of computer software that are not compatible with their personal approach to problem-solving. The personality trait selected was based on the importance accorded it in the literature. This attribute is "locus of control" - the degree to which individuals view events as occurring as result of their personal actions.

Two sets of computer programs were used on an IBM personal computer to simulate dialogues that either 1) gives the user very limited control over the interaction or 2) gives the user considerable control. The sample consisted of undergraduate business students. The task performed in the study was limited to a data entry operation.

It was found that locus of control did not have an effect on either time to completion or on the number of steps to completion of either dialogue task.

Hence, the theory that performance in a man-machine dialogue is affected by anxiety (due to a mismatch of personal attitude with software capability), could not be supported. The lack of support for the theory is attributed to the inability to produce anxiety when the incongruence is based on personality factors.

It is recommended that further studies be conducted to determine the validity of the performance theory. Such studies should consider dimensions of the man-machine interaction outside the affective domain to induce the anxiety necessary to test the performance theory. Studies should also investigate the effect of dialogue interactions over time as it is hypothesized that anxiety exists in levels according to the amount of exposure to the man-computer environment.

ACKNOWLEDGEMENTS

I would like to express my appreciation to Dr. Richard Featherstone, chairman of my committee, and to the other committee members, Dr. Gerald Miller, Dr. George Sargent, and Dr. John Vinsonhaler for their contributions to this study.

Additionally, I am especially indebted to Dr. Carole Beere and Dr. Jerald Lounsbury, without whose counsel and encouragement I might not have completed this dissertation.

Finally, I owe a deep debt of gratitude to my wife, Jill, and daughter, Nicole, who gave me the understanding, support, encouragement, and sacrifice so necessary to complete this project.

TABLE OF CONTENTS

APPENDIX PAGE

LIST OF TABLES

TABLE		PAGE

LIST OF FIGURES

CHAPTER I

THE PROBLEM

Computer technology has been advancing at a rapid rate. Computers have become less expensive and more powerful thereby putting information processing capability in the hands of an ever increasing number of individuals.

While computer technology has advanced, our psychological awareness of the man-machine interaction has not increased significantly. Individuals express anxiety and fear as they find computers irresistible but unwieldy, appealing yet threatening. Sackman (1970) and Shneiderman (1980) have indicated that individual differences are greater than computer system differences and suggest the need for further exploration of personality and behavior characteristics. The little research which has been done on the man-machine interface has focused on man's cognitive abilities. Personality variables have been largely ignored.

It is conceivable that future computer systems can be sophisticated enough to adapt man-computer dialogues to specific individuals. Given sufficient knowledge about individuals and their reaction to various man-machine dialogues, it is possible to develop computer software that can take different forms based on the personality traits of the individual using the system. Future systems could draw upon the personality profiles of users as they sign on to systems. The system could then modify the presentation to best accommodate the user. It is not beyond the realm of possibility to even anticipate systems that could modify the man-computer dialogue according to the current mental state of the user, perhaps via bio-feedback technology.

Currently, however, very little is known about the user's reaction or response to various computer dialogues. Terms such as "user-friendly" and "ease-of-use" are frequently used but as yet lack clear definition.

It is the intent of this thesis to determine if a single personality trait can be isolated and be shown to have an effect on the man-machine interface. That effect will be measured in terms of both performance and attitude. The personality trait that has been selected is locus of control. This trait was selected because control is frequently cited as an element of man-machine dialogues (Miller & Thomas, 1976; Dehning, Essig, & Maass, 1981.)

Purpose of the Study

It is the purpose of this thesis to determine if the personality of an individual affects the individual's performance in a man-machine interaction. Furthermore it is hypothesized that such performance is directly related to the manner in which the computer system communicates with the computer user, that is, the form of the dialogue used.

This thesis focuses on a single dimension of personality -- locus of control -- as it affects a user's preference for computer dialogues. The environments of concern in this study are characterized by two commonplace man-machine dialogues. One form of dialogue gives users control over the interaction; the other form of dialogue forces the user to follow a system controlled interaction.

The specific man-machine interaction addressed by this thesis involves the communication between computer users and computer software via personal computers. Computer software in the context of this study refers to those programs that present information to users via terminal screens, receive data via terminal keyboards, and process the data. The communication between the user and computer is termed a dialogue.

Much of the software currently available tends to be based on the premise that the performance of an individual in a man-machine interaction is a function of the individual's cognitive skills and the user's short term memory capacity. The cognitive skills required for a given interaction might include a level of familiarity with the equipment used, and might include knowledge of the task to be performed using the computer system. Short term memory is that memory people possess to "process a sensation and hold interpreted units of information for up to 30 seconds, but this period can be extended by continued rehearsal or repetition" (Shneiderman, 1980, p. 224). The relationship of cognitive skill level and short-term memory capacity to the performance of the individual might be expressed as --

$$PC = \frac{CS_p}{CS_r} + \frac{STM_p}{STM_r}$$

where p = abilities possessed by the individual

r = abilities required for the task

CS = cognitive skills

STM = short term memory

If an individual possesses fewer cognitive skills than are required, performance will diminish. Or, if the short term memory capacity of the individual is not optimum for the task, performance will diminish.

Another dimension to the man-machine interaction is the level of frustration or anxiety imposed by the dialogue between the computer user and the software (programs) that the user is interacting with. Taking the personality of the user into account the equation might be expressed as --

$$PC = \frac{CS_p}{CS_r} + \frac{STM_p}{STM_r} - ANX$$

where the added dimension ANX is the anxiety introduced by the dialogue as a result of the the lack of compatibility between the personality of the user and the presentation of information by the computer software. The anxiety factor can become a liability in terms of the user's processing capability.

Importance of the Study

In 1978 there were approximately 600,000 computers in the United States. The number had grown to 2 million by 1981 and even conservative estimates predict there will be 7 million by 1985 (Conrades, 1982). Due, at least in part, to this phenomenal growth, the relative cost of processing data has rapidly declined since the early days of computer processing. Processing which cost one dollar in 1952 now costs .0076 cents (based on Conrades' (1982) comparison of the IBM 701 against the current IBM 3081).

The nature of interaction with computers has also changed. The earliest systems were "batch-oriented". According to Dock and Essick "under batch processing, the data is collected over a period of time and in a separate step before it is submitted to the computer for processing" (1981, p. 205). The movement during recent years has been from batch processing to an interactive environment in which users communicate with computer systems via terminals. In 1981 there were approximately four terminals per 100 professionals employed in the United States. By 1986, estimates are that there will be more than 16 terminals per 100 professionals (Conrades, 1982).

The relative performance of computers has also increased. The IBM 3081 delivers nearly 1700 times the processing capabilities which were produced by the IBM 701 in 1952 (Conrades, 1982).

A computer must be instructed what to do; this direction is provided by a computer program. Traditionally, programs have been developed by trained technical personnel called programmers. Programming "is the execution of tasks necessary to create information from data by providing the proper instruction to a computer" (Dock & Essick, 1982).

A computer user is one who benefits from the processing performed by the computer. As most users have not been trained in computer programming, they have had to rely upon professional system analysts and computer programmers to develop the necessary software (computer programs) for their computers.

As computer hardware continually becomes less expensive and more powerful, there will be an increased shortage of trained personnel to instruct, or program, computers. At the current rate of expansion, and assuming the current methods of programming computers, the bulk of the work force in the United States would actually have to be writing programs. Programming will not remain solely in the hands of the computer professional. There will be too many systems available for users to continue to require the training of computer professionals for the programming of every computer or terminal available.

Many individuals heretofore have not utilized computers because of their fear of computers or perhaps because of frustration resulting from bad experiences with dialogues that were not conducive to the personality of the individual. These individuals nevertheless will find the computer to be a necessary tool in the future if they are to perform the tasks expected of them in our modern technological society. It is therefore important to expand our knowledge of the man-machine interaction so that more individuals can use computers in a productive manner without unnecessary anxiety that could reduce their processing capacity.

Statement of Research Questions

Following is a set of questions to delineate the scope of this thesis.

1. How do people with differing personalities respond to specific man-machine dialogues? The concern here is very broad. Perhaps different people perceive software products or computing environments differently. These different perceptions may be caused by differences in educational experiences, differences in cognitive abilities, or possibly, affective differences. Many times the term "user-friendly" is used by software vendors. There are certainly many connotations to the phrase. One connotation might be contingent upon the personality makeup of the individual.
2. Do people perceive themselves to be in control in certain man-machine interactions and not in

others? When people desire to be in control and are not, their level of anxiety may well rise. When a software product or computing environment gives people a feeling that they are not in control, will this situation affect their performance and hence productivity? Will users be more effective with tools that match their personality?

3. Is a person's attitude (negative or positive) toward a software product or computing environment due to the lack of cognitive skills requisite to the tasks to be performed or possibly due to man-machine interactions that are not congruent with the personality makeup of certain individuals?
4. Do software products have to be easy to use to be user-friendly?
5. What features or attributes of software cause frustration or anxiety for users? Do the same characteristics bother all people in a similar manner? Can features be identified that affect different personalities differently?
6. What causes fear or apprehension when utilizing computer systems? It would be desirable to be able to identify facets of products which might be deemed as threatening or detrimental to use by individuals who are not themselves assertive.

CHAPTER II

A SURVEY OF THE LITERATURE

Man interacts with computers in many ways for different purposes. Some are involved with the computer as a tool to resolve problems or retrieve data using software created by others. Some computer users (typically professional programmers) are involved in creating the software necessary for the problem solving efforts of other users. Still others interact with computers in developing software for there own problem-solving or information retrieval needs (user-programming).

Regardless of the task being performed on the computer a man-machine interaction exists. Further knowledge of the interaction can contribute to the development of software that will make users of computer software more effective during a given task, whether that task be programming or information processing. More complete knowledge of the user of a computer system can also lead to software that may cause less anxiety on behalf of users who henceforth may have suffered undue anguish while working with computer systems that were not conducive to the affective or cognitive nature of the user.

The languages currently designed for users include a class of software called query languages. A review of research conducted with query languages is included in the survey of related literature. Most research on the use of computer software has been in the realm of high-level languages such as COBOL, FORTRAN, and BASIC. Research has also been conducted using text editors.

While users typically would not use high-level languages such as COBOL or text editors, a review of this literature seems warranted because much of the work done can provide insight to the interaction of man and machine via computer languages. A review of literature relating to high-level languages and text editors is, therefore, included.

A portion of the review of the literature is directed at what is currently known about software development both by professional programmers and by end users. The other portion of the literature review is directed at gaining insight into why the performance of individuals varies with different software products.

Finally, the main intent of this thesis is to explore the psychology of the user. Shneiderman suggests that "the impact of personality differences should be investigated to see if different languages, programming environments, or interactive tools might aid specific personality types" (1980, p. 62). Both the cognitive and the affective realm are explored through the available studies of user-oriented query languages and through the studies of high-level languages and text editors.

Generations of Computers

It has been stated by many authors that computers have evolved through four generations (Dock, 1982; Gore & Stubbe, 1979; Martin, 1982). The Japanese Information Processing Development Center (JIPDEC) has put forth a national Japanese plan for a fifth-generation system by 1990 (Manuel, 1982, p. 141). To date, computer manufacturing and software development has been dominated by the United States.

The goal of the Japanese is to "have a higher performance level at lower cost, be able to handle many more general problem solving tasks, ... and to be as natural for people to use as it is for them to speak" (Manuel, 1982, p. 142). This first function is referred to as the intelligent-interface machine.

A second function will be the "system's ability to learn, associate, and infer, just as people do" (Manuel, 1982, p. 142). This function is to be implemented on a separate machine and is termed the problem-solving and inference system. The last capability planned will be the ability to use stored information. The base of data will constitute a knowledge base as opposed to a data base. Such a base is necessary to feed the problem-solving function. This last function will be known as the knowledge-based management system.

Should the Japanese succeed in their endeavor, the traditional methods will no longer be barriers to effective use of computers by users. Users will themselves be interacting via a 'natural language', that is, a language that closely resembles their native tongue.

Computer languages can be better designed with human factors considered, as more is learned about people, both in the cognitive and in the affective domains.

Types of Programs

Most computer environments require programs to perform six basic functions (Gore & Stubbe, 1979). First, computer instructions are necessary to create or load data files. Second, programs are necessary to update or process user transactions. Third, files need to be backed-up for protection against the inadvertent loss of data, e.g., hardware failure. Fourth, the maintenance function requires programs allowing additions, changes, and deletions to the data files. The fifth function, input data editing, is necessary to ensure that only valid, correct data is entered into the system (as determined jointly by the user, analyst, and programmer). The sixth function, report generation, is the most encompassing program function. Within this realm lies all programs that produce output from a computer system in a human readable form. The output may be on paper, a terminal screen, microfilm, or by synthesized voice. The nature of the 'reports' generated may range from conventional listings of data to simulations performed at a terminal. In recent years much software development effort has been directed toward assisting users in the retrieval of information via queries to data bases. The necessary programming for report information ranges from simple to highly complex.

The first five program functions have been traditionally programmed by professional programmers. In recent years the last function, report generation, has come into the domain of the user. (Some user-oriented languages also allow users to perform some or all of the first five functions with a minimum of training (Martin, 1982).

<u>Language</u>.

Programming languages may be classified in several ways. Early computers only utilized what has become known as low-level languages (Martin, 1982). These are languages which are very close to machine language. A great deal of training is necessary to write in low-level languages. The benefit of using such languages is in the speed with which low-level language programs execute instructions and their generally efficient use of memory.

Today low-level languages are typically called assembly languages and are used in writing compilers for higher level languages, operating systems, data communication programs, and data base management languages. Assembler or low-level programming is invariably performed by highly trained technical personnel. Computer Science curricula prepare students well for this type of work (Dock & Essick, 1982).

In contrast to the low-level languages are the high-level languages. Examples of high-level languages are COBOL, PL/1, FORTRAN, BASIC, and PASCAL. Whereas high-level programs take relatively less time to write, they are also less efficient to execute and require more computer memory for the same task. However as computer hardware costs drop and personnel costs rise, most organizations choose to develop their applications in high-level languages. In addition to the time saved in programming, the high-level languages offer portability. A program written in one of the high-level languages can be run on a computer of a different manufacturer with minor modification.

The demand for new applications in most companies is rising faster than data processing personnel can supply the application software. In fact, the known applications are only a part of the problem. Because many users are aware of long waiting lists for application development, many system needs are never revealed to computer center personnel.

Need to Focus on the User

There are two solutions to meeting the impending crisis in software development. One focuses upon the professional programmer and systems analyst, the other focuses on the user.

Programmers. With conventional methodologies, system analysts meet with users and determine their needs then proceed to define a solution. When formulated the solution is then given to applications programmers who write the solution in a high-level language, most frequently in COBOL. Productivity under the aforementioned methodology can be increased by the use of what have become known as programmer productivity tools or by gaining insight into how to become more efficient with high-level languages such as COBOL. Martin has the following observations on programming productivity. "First, the best programmers achieve a much higher productivity than poor programmers. Second, productivity is higher with small programs than with large ones. It can be improved by modular design, especially with structured data base facilities in which no one module becomes too large. Third, productivity falls with highly complex programs. Good design can reduce complexity" (1982, p. 39).

Another approach to increasing productivity is via application development tools such as IBM's Application Development Facility (ADF) or IBM's Development Management System (DMS). These tools compliment program development in high-level languages by reducing the time to develop certain types of applications. For example, DMS is a tool for generating interactive applications. DMS allows professionals to define terminal screens, access data bases, validate data, etc. It is designed not for end users but for the computer professional. While many application development aids are available, the movement toward such tools is slow. Martin suggests that one reason might be that development tools are considered as a threat to the knowledge of systems analysts and programmers (1982).

Users. The second solution to the impending crisis in software development is addressed to the end user, the person for whom systems are developed. Many users have become disenchanted with computer professionals and the time it takes to get even the most rudimentary data from the data processing department via the analyst/programmer route. It is not uncommon to wait weeks or months for changes to a report or to wait even years for totally new applications.

During the past decade many computer manufacturers and software companies have developed languages especially for users. Which products are suitable for end users is open to debate. Martin offers the following '2 day test'. To pass the two day test a product should have the following properties:

> Most end users can learn to use it effectively in a 2-day course. Some can learn it much faster.
>
> At the end of this course they are comfortable with it and can use it on their own.
>
> At the end of the course they can start getting useful work out of it. This emphasis is on useful work is important. There is no point in learning gimmicks that have little relevance.
>
> After the course the end users can leave the product for a week and still be able to use it. (Most users would forget mnemonics or fixed entry sequences in a week.)
>
> End users will not necessarily have to return to another class on the product.
>
> Users can expand or refresh their knowledge of the product at the terminal

> by using HELP features and computer-aided instruction (Martin, 1982, p.107).

(In some cases the 2-day course may cover a useful subset of the product's features and more can be learned later when the users have had experience with it.) The tasks that might be performed by users is also open to debate. Many computer professionals firmly state that users should not perform any tasks themselves. Others, such as Martin suggest a broad array of activities utilizing both data bases and personal files (Figure 1).

The problem with defining a list of activities that might be 'programmed' by users is that very little is known about the 'psychology of computer programming' or about the 'psychology of the user'. As more is learned through research about users and programmers, and as more is learned about the use of available software, it will become possible to state what skills or attributes are necessary for performing the many tasks outlined in Figure 1.

Experimental Design

Certain problems arise in the conduct of experiments in general and in programming experiments specifically. Constantly, empirical methods must balance between the conflicting goals of attaining results that may be generalized and of ensuring reliability.

As Sheil's has stated, "real-world programming performance varies in too many ways to allow reliable

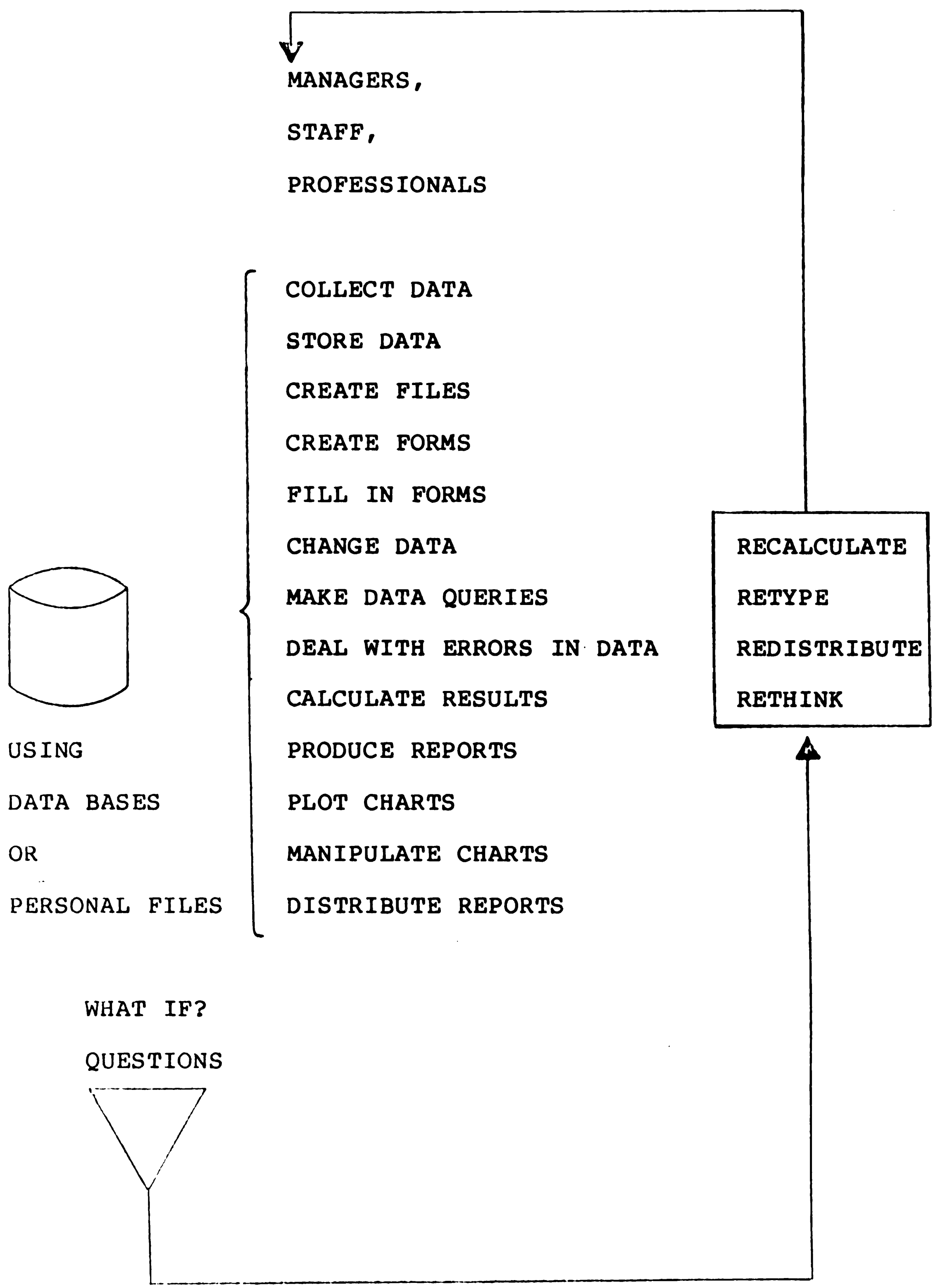

Figure 1. End-User Tasks

Source - Martin, 1982, p.105

interpretation of its causes" (1981, p. 113). The very act of introducing some technical innovation for the purposes of evaluating it may cause changes of behavior that are unrelated to any properties of the innovation (the Hawthorne effect).

An alternative to real-world experimentation is to construct artificial experimental situations in which extraneous influences can be either eliminated or controlled. However, the use of non-programmers requires training, thereby introducing the learning process. The danger here is that activity of learning might dominate the results.

Furthermore, if one wishes to approximate real-world situations one should use complete programming tasks. However, this is difficult because of the introduction of a large number of extraneous variables. Sheils noted that "this suggests a focus on either isolated aspects of the programming task or the psychological claims that implicitly underlie different programming techniques" (1981, p. 103).

The problem then is that the distance of the isolated task from real-world programming makes generalization difficult. It should be noted that the aforementioned concerns with reliability versus generalizibility are greater in the study of professional programmers with large tasks than with users involved with 'user-oriented' software.

Much attention has been given to programming notation, practices and tasks. Computer vendors have introduced aids to programmers and many authors have advanced several methodologies to programming. These aids and methodologies have not necessarily improved actual productivity in new program development nor demonstrated improvements in the maintenance of programs.

Conventional Programming Literature

Notation Since Dijkstra's article (1968) on the detriment of using GOTO statements in conventional programming languages, many studies have been conducted to examine the use of structured programming. Of particular concern has been the IF-THEN-ELSE construct compared to explicit transfers of control. Statements are constructed as follows in a nested manner:

```
IF condition A
    procedure A1
ELSE
    IF condition B
        procedure B1
    END IF
END IF
```

The alternative method (test and jump) utilizing the GOTO statement might be written as follows:

```
IF condition A GOTO procedure A1.
IF condition B GOTO procedure B1.
```

When reaching procedure A1 or B1, additional tests and jumps might well exist. By having the procedures distant from the conditions, programs are more difficult to follow logically than with the IF-THEN-ELSE construct where the procedures are coded adjacent to the condition being tested.

In testing the above constructs, experiments are designed such that students are either instructed to write solutions to problems utilizing one of the notation methods or they are asked to explain the meaning of code already written. In a study conducted by Sime, Green, and Guest (1973) and replicated later (1977) it was found that the nested approach resulted in a significant reduction in semantic errors ($p > .01$) over the explicit transfer of control (test and jump). However, syntactic errors were significantly fewer for the test and jump group. Sime, et al., also measured the number of additional attempts after an initial error and found that the 'error lifetimes' were greater for the test and jump. This study gives ample evidence of the superiority of nested conditional statements over the use of the GOTO to jump over program statements.

In the 1977 study, Sime, Green, and Guest included another notation termed the repeated predicate form. By embedding the actions within the conditional statements semantic and syntactic errors were both reduced significantly ($p < .01$, $.05$ respectively) and the error lifetime was .09 compared to 1.06 for the jump and test and 1.60 for the nested notation ($p < .02$). Unfortunately, the statement formats chosen differed significantly between the nested and the repeated predicate. In the nested each action was preceded by the word BEGIN and terminated by the word END. In the repeated predicate BEGIN and END were not used. This difference in syntax alone could have caused significant differences in results (Shneiderman, 1981).

Practice Effect. Several studies (Green, 1977; Sime, 1973; Shneiderman, 1976; Winer, 1971) have shown large practice effects, that is, marked improvement in performance with each session. Many times the practice effect was so great as to overshadow the effects of the experiment being measured. In Sheil's review (1981) of Greeley's 1977 comparison of structured and unstructured languages, Sheil states "whereas the effect of the different languages is to change the mean reaction time by amounts which range from 4 to 15 percent, the effect of a single session of practice ranges from 13 to 27 percent" (p. 106).

Whenever conducting an experiment with experienced and inexperienced subjects one should be cognizant of such practice effects.

The research of structure versus unstructured programming, aside from the conditional, has yielded mixed results. The results obtained do not adequately support the benefit of structured programming. This is not to say that such a benefit does not exist. Sheils states "the evidence suggests only that deliberately chaotic control structure degrades performance". One reason cited for inclusive results is that structured programming is "a discipline, a way of thinking" (Sheil, 1981, p. 107) and hence is simply not easy to simulate in an experimental situation. Another problem with experiments in structured programming is that simply prohibiting "the use of a program feature without providing any motivation or alternative strategies . . . is hardly likely to produce much else other than resentment" (Sheil, 1981, p. 107).

In his summary of research on programming notation Sheils states, "given the small sizes of and inconsistencies among the reported effects, it is not even clear that notation is a major factor in the difficulty of programming . . . it comes as a shock how little empirical evidence there is for their importance . . . many of these effects tend to disappear with practice or experience. This raises some doubt as to whether these results reflect stable differences between notations or merely learning effects and other transients that would not be significant factors in actual programming performance" (1981, p. 108).

Flowcharting. There is a lack of evidence as to the value of flowcharting prior to writing programs. Shneiderman conducted "five successive experiments, with different tasks and measures (and) failed to reveal any reliable advantage of flowchart use" (Sheil, 1981, p. 109). However, the experiments themselves may have caused the inconclusive results. Two of the experiments suffered from 'ceiling effects'. Both the control and the experimental groups achieved scores near the maximum leaving little room for measured differences. In other experiments Sheils points out that the lack of results might be explained by the "choice of materials, language, and/or participants" (1981, p. 109).

Indenting (Prettyprinting). Instructors frequently encourage students to make their programs neat and perhaps even conform to a particular style. For example, in COBOL programming instruction common practices are to limit students to one statement per line, to require all Data Division entries of the same level to be in the same column, and to have the word PICTURE begin in the same column throughout the Data Division. Within the Procedure Division it is common practice to have IF-THEN-ELSE sentences indented for clarity as illustrated in Figure 2.

Sheil noted that research by Weisman (1974) concerning indentation and 'pretty printing' revealed "positive self-evaluations but no performance improvements for either modifying, hand simulating, or answering questions about indented versus unindented versions of programs" (Sheil, 1974, p.109). Another technique widely recommended in COBOL programming is the use of paragraph numbers. In COBOL a paragraph name (similar to a statement number in other languages) may be composed of letters and/or numbers up to 18 characters in length according to ANSI 74 standards. Most COBOL users tend to create meaningful self-documenting program names. Some COBOL users go a step farther and

```
DATA DIVISION.
01  COUNTERS.
    02  MINOR-COUNTER    PICTURE S9(5) VALUE 0.
    02  SENIOR-COUNTER   PICTURE S9(5) VALUE 0.
    02  OTHER-COUNTER    PICTURE S9(5) VALUE 0.

PROCEDURE DIVISION.
.
.
IF AGE < 18
   ADD 1 TO MINOR-COUNTER
ELSE
   IF AGE > 65
     ADD 1 TO SENIOR-COUNTER
   ELSE
     ADD 1 TO OTHER-COUNTER.
```

Figure 2. IF-THEN-ELSE construct.

assign a three or four digit prefix to each paragraph name such that the numbers are in sequence throughout the program. The premise for the use of the number prefixes is that such paragraph names will be easier to find when debugging or modifying someone else's program. No research has been conducted on the benefit of the 'paragraph name with number prefix' method.

Comments. It is generally considered a good programming practice to insert comment lines into a program for the purpose of providing documentation to future readers of the program. Studies that have been conducted (Weisman, 1974; Shneiderman, 1977; and Sheppard, Curtis, Milliman, & Love, 1979) have put forth mixed conclusions as to the benefit of comments. Sheppard, et al. found that comments had no effect on the accuracy of the ability to modify programs (1979), while Shneiderman found that programs were easier to modify with high level comments present (1977).

The choice of language, the subject matter of the programming effort, and the clarity of the code itself could have a great effect on the benefit of comments. For example, in COBOL one can write nearly self-documenting program statements and paragraph names thereby negating the benefit of comments. In some BASIC languages (e.g., CDC BASIC), variable names are restricted to two characters and statements are identified by five digit numbers. Hence, in some BASIC languages, comments might prove very useful.

A further point on comments relates to the real-world programmer. It is very common for a programmer to be required to modify a program unfamiliar to him and frequently involving an unfamiliar application. The type of comments useful to the professional programmer might be of an entirely different type than would be thought of by the inexperienced programmer. For example, in COBOL the reference to a copied table might be commented in such a manner as to define the source of the table and its organization which could reduce the time needed to modify the program by providing a more complete understanding of the data processed.

As Sheil states, "it is equally clear that a comment is only useful if it tells the reader something she either does not already know or cannot infer immediately from the code" (1981, p. 111).

Variable Naming. There have been several studies of variable naming conventions (Weissman, 1974; Shneiderman, 1980; Sheppard, et al., 1979). These studies have attempted to ascertain whether mnemonic variable names were an aid in debugging programs. The experiments were inconclusive in that groups performed no better at debugging programs with mnemonic names than with programs without mnemonic names. The only improvement cited was in self-evaluation by the participants.

It should be noted again that experiments typically involve small programming tasks. In such tasks mnemonic names could well offer little benefit over 'meaningless' names, as the subjects could remember a limited number of variable names without memory aids.

Database Query Language

"A query language is a special-purpose language for constructing queries to retrieve information from a database of information stored in the computer" (Reisner, 1981, p. 14). The intent of a query language is to provide a tool for the retrieval of data that is easy to use. The alternative is to rely on professional programmers to develop programs whenever information is needed by users in a form not already available.

Many computer manufacturers and independent software vendors now offer query languages for the many databases available. Each software firm claims ease of use as a major attribute of their product. Martin offers a 'two-day test' (previously discussed) to determine if a product is really user-oriented as opposed to an orientation more favorable to professional programmers (Martin, 1982).

Ease-of-Use. The main problem encountered in query language studies has been the measurement of ease-of-use. Reisner offers several tasks that have been used in evaluating query languages (Figure 3).

The kinds of tests used to measure ease-of-use vary widely. Some studies use only a single test, others have used several. Reisner summarizes the most common approaches:

1. Final exams of learning. These tests how easy a query language is to learn; they are given at the end of teaching.

2. Immediate comprehension. These help identify why particular learning problems occur. They are given during teaching, immediately after some function has been taught, to determine whether subjects can use the function, given that they know it is the one to use.

3. Reviews. These help identify why particular learning problems occur. They are given during teaching and cover functions taught up

Task	Description
Query writing	Users are given a question stated in English and required to write a query in the given query language
Query reading	Users are given a query written in the query language and asked to write a translation into English.
Query interpretation	Users are given a query in the query language and a printed database with data filled in. They are asked to find the data asked for by the query.
Question Comprehension	Users are given an English question and a printed database and are asked to find the data asked for.
Memorization	Users are asked to memorize and reproduce a database.
Problem solving	Users are given a problem and a database and are asked to generate questions in English that would solve the problem. The questions should be answerable from the database.

Figure 3. Query Language Tasks

(Reisner, 1981, p.16).

to that time. They require that subjects know which function to use.

4. Productivity. These are tests of query language use by 'skilled' users. They test how well the language can be used after some predetermined level of learning has been attained.

5. Retention. These test how easy a query language is to remember: how well it can be used by people who have been away from it for a period of time.

6. Relearning. These test how easy a query language is to relearn by users who have been away from it for a period of time and have forgotten some of it (1981, p. 17).

Some of the basic differences in query languages are their syntactic form, procedurality, and the underlying data model.

Syntactic Form. Syntactic form is the manner in which queries are constructed. Figure 4 illustrates queries in four different query languages to find the names of employees in department 50.

While some studies (Greenblatt & Waxman, 1978; Reisner, 1975) have compared two languages of different syntactic form, their conclusions on ease-of-use should not be attributed purely to the syntactic form of the languages. The studies to date have not been sufficiently controlled to unequivocally state, for example, that a two dimensional syntax as used in QBE is easier to use than a linear syntax as used in SQL.

Language	Example
SQL	SELECT NAME FROM EMP WHERE DEPTNO = 50
QBE	EMP NAME DEPTNO p.Brown 50
SQUARE	EMP ('50') NAME DEPTNO
TABLET	FORM DEPTFIFTY FROM NAME, DEPTNO OF EMP KEEP ROWS WHERE DEPTNO = 50 PRINT NAME

Figure 4. Syntactic Form in Four Query Languages. (Reisner 1981, p.14).

Procedurality. Following is a review of a study conducted by Charles Welty (1979) to determine if difficult queries to a data base are written more easily in a procedural or in a non-procedural query language.

According to Welty, ". . . a language is procedural if it specifies a step-by-step method for achieving a result. Non-procedural languages describe the desired result without specifying how it is to be achieved. (The idea is comparable to the difference between constructive and nonconstructive existence proofs in mathematics)" (1979, p. 16).

Further background on the formation of the definitions is provided by Codd (1971). A procedural query is based on relational algebra. Through relational algebra distinct ordered steps are defined. "An operation on a relation or relations always yields another relation" (Welty, 1979, p.16).

A relational calculus query as defined by Codd, "describes the elements of the desired relation. The query is purely descriptive, containing no method for achieving the desired relation" (Welty, 1979, p. 16).

Query languages are meant to be used by persons who are not computer professionals, i.e., professional programmers. The use of such a language is ancillary to the users main work and hence the language must be easy to use or it will not be utilized.

While papers have been presented (Codd, 1971; Date, 1977) on the superiority of non-procedural languages over procedural languages, the papers have not been the result of research but reflect the opinions of the authors.

Welty hypothesized that "people more often write difficult queries correctly using a procedural query language than they do using a non-procedural query language" (1979, p. 15).

The languages chosen to test the hypothesis were SQL and TABLET. Both languages use a relational model as opposed to a network or hierarchical model. Both are relationally complete. Both have similar language levels as measured by the Halstead method (Halstead, 1977). And both languages utilize the same terminal equipment. Their difference lies in that SQL is a non-procedural or descriptive language while TABLET is a procedural or constructive language. The experiment participants were attracted by the offering of a one credit course offered by the Accounting

Department of the University of Massachusetts at Amherst. The instructional material for the two languages was prepared by using the SQL training manual and then rewriting the TABLET manual to be similar in format and content. The representation in the 72 participants was primarily business undergraduates. However, the background of the individuals varied considerably as can be seen from Table 1.

The group tended toward Freshmen and Sophomores which would imply very little business exposure. Hence, being a business undergraduate only indicates a business interest not a business background or exposure through coursework.

Nearly half of each experimental group had previous programming experience. Sixty-five of the sample had calculus while the remaining 17 had at least a pre-calculus background. One might question whether such a mathematical and programming background might be representative of actual users of query languages in a business organization. The familiarity with symbols displays knowledge that might be typical of business school graduates if they were required to take a business statistics course.

The subjects were divided into four groups. Two groups learned SQL, two learned TABLET. For each language two groups were identified as experienced in computers (having taken a course in FORTRAN or BASIC) or inexperienced.

Instruction was via the prepared manuals. At the class sessions, a question and answer period was conducted and then a quiz administered. Any question asked in one class

Table 1

Subjects of SQL and TABLET Study

Class	SQL	TABLET
Senior	5	4
Junior	5	5
Sophomore	14	6
Freshmen	15	22
Major		
Business Admin	17	19
Accounting	5	7
Fashion Mktg	1	2
Marketing	1	1
Other	11	8
Computer Experience		
None	17	20
BASIC	10	10
FORTRAN	5	6
Other	3	1
Math Background		
Calculus	28	27
Pre-Calculus	7	10
Familiarity with Math Symbols		
>	34	36
<	34	36
=	34	36
u	25	27
n	26	27

Source - Welty, 1979, p.366

was covered in the other classes. The experiment was pre-tested at another college.

Success in each language was measured by a paper and pencil final exam and a retention exam given three weeks later. The tests were scored by counting errors. The test responses were graded as essentially correct or incorrect according to a classification of errors developed by Reisner (1976).

As query languages are designed for the casual user, the retention test is most significant. According to Welty the results of the retention test support the hypothesis that subjects using a procedural language (TABLET) would have significantly more correct responses than those using a non-procedural language (SQL). However, the tests were comprised of "easy" and "hard" queries which did not show the same results. No difference was shown between the languages for the easy queries. The procedural language (TABLET) showed significant advantage over the non-procedural language (SQL) for the hard queries. The results were significant at the .05 level.

Welty also stated that he found that experienced subjects performed better than inexperienced with a procedural language ($p > .005$) but there was no significant difference when using the non-procedural language. Experience difference was based on programming exposure to either FORTRAN or BASIC. However a complete description of the backgrounds by group was not provided. The difference could be attributed to factors other than programming experience, for example, mathematical background (1979).

It is reasonable to conclude from this study that procedurality in a language does enhance its ease of use and the ease of learning with difficult or hard to write queries. It is also fair to conclude that subjects with a semantic reference structure built into their long term memory will perform better when using a procedural language than those who have not developed such a structure via a programming language. As most business schools offer, and frequently require, an introductory data processing course which requires minimal study of FORTRAN or BASIC, future employees of business organizations should be somewhat prepared for procedural query languages. The background necessary to enhance learning of a non-procedural language is yet to be determined.

Data Models. "In most query languages the user is assumed to have a conceptual view of how the data are stored in the computer. Three well known data models are the relational model in which data are assumed to be stored in the form of tables; the hierarchical model, in which data are assumed to be stored in the form of tree structures; and the network model, in which data are assumed to be stored in the form of general graph structures" (Reisner, 1981, p.15).

Lochovsky and Tsichrizis conducted studies comparing the three data models (Lochovsky & Tsichritzis, 1977; Lochovsky, 1978). The subjects were computer science and business administration students. They were classified as "more experienced users" if they had six months or more of programming experience, otherwise they were classified as "less experienced users".

The students were given instruction in the APL language and then given user manuals and programming problems to study for a week. The subjects were given tests containing query writing tasks. They were then given the task of debugging and running the examination queries. Another set of queries was given after the subjects had worked with the on-line system.

The results of the experiments clearly showed the relational model to be superior. For the less experienced users, the relational model was significantly better $p < .01$). Experienced users did significantly better with all three models ($p < .05$) than the inexperienced users. The experienced user, however, did better with the relational model only before their on-line experience.

Two problems exist with these studies. First, the models used were not products in commercial use but were developed for the purpose of the experiment, hence the results can only suggest that other software products incorporating these models might yield the same results.
Second, as Lochovsky points out, "it is difficult to attribute the differences in performance to either the data model or the data language, since they were not separated" (Lochovsky, 1978, p. 22).

Brosey and Shneiderman (1978) conducted a study to determine if the data model alone caused differences in ease-of-use. Their study included only the relational and hierarchical models. The subjects were undergraduate students grouped by experience in programming. The beginners had two or three terms of programming. The advanced group had six terms of programming. The researchers used a question comprehension, a memorization, and a problem-solving task. On the comprehension task the hierarchical model was easier for beginners to use (significant at $p < .05$) but not for the advanced group. Schneiderman, in a review of his own experiment concludes that "although the relational model is . . . possibly a convenient notation in general, there exist circumstances in which the tree model is easier to use" (1980, p. 167).

Reisner offers a word of caution in using the tasks of question comprehension and memorization. These tasks are less related to real-world tasks than are query writing tasks (1981).

Other studies of how people organize data show that people do have structures like the data models previously discussed. Reisner noted that in the studies by Durding, Becker, and Gould subjects "were able to organize words into these structures based on the semantics of words, and had difficulty on a task that required them to use words in inappropriate structures" (Reisner, 1981, p. 22).

Broadbent and Broadbent (1978) studied database structures in a non-database query mode and found individual differences dependent upon educational background.

Affective Dimensions of Man-Machine Interfaces

This section focuses on general problems in man-machine interfaces, particularly those in the affective domain of psychology. The discussion excludes hardware factors such as keyboard or video display design, as well as excluding software topics such as menu selection or command languages.

Attitude and anxiety. Users attitudes can dramatically affect their performance. According to studies by Walter and O'Neil "novices with negative attitudes towards computers learned editing tasks more slowly and made more errors". They also suggest that anxiety (fear of failure) "may reduce short-term memory capacity and inhibit performance" (Shneiderman, 1979, p. 225). Anxiety can be caused by the unknown as in a timesharing environment. The novice user lacks complete knowledge and hence may fear the loss of files (invisible to the user in some remote location) or possibly concern for destroying the computer system which he is attempting to interact with.

Shneiderman suggests that "every attempt should be made to make the user at ease without being patronizing or too obvious" and that "the user will feel best if the instructions are lucid, in familiar terms and easy to follow". "Diagnostic messages should be understandable, non-threatening and low-key . . . avoid meaningless, condemning messages such as "SYNTAX ERROR" where a constructive informative message can be displayed such as "UNMATCHED RIGHT PARENTHESIS". He suggests that "constructive messages and positive reinforcement produce faster learning and increase user acceptance" (Shneiderman, 1980, p. 226).

Control. Individuals may be classified by their desire to control or be controlled (external versus internal locus of control) (Rotter, 1966). Shneiderman asserts that individuals desire to be in control, and that "with respect to computers, the desire for control apparently increases with experience" (Shneiderman, 1980, p. 226). This is an untested hypothesis. Users may resent messages which imply that the computer is in charge, for example, the authoritarian phrase "ENTER NEXT COMMAND" compared to the servile "READY FOR NEXT COMMAND". The previous example is representative of changes made by the Library of Congress in their interactive systems.

The Equitable Life Assurance Society has the following set of guidelines for developing interactive systems.

> "Nothing can contribute more to satisfactory system performance that the conviction on the part of the terminal operators that they are in control of the system and not the system in control of them. Equally, nothing can be more damaging to satisfactory system operation, regardless of how well all other aspects of the implementation have been handled, than the operator's conviction that the terminal and thus the system are in control, have 'a mind of their own', or are tugging against the operator's wishes". (Shneiderman, 1980, p. 227)

Word processing systems have gained widespread acceptance perhaps in large part because they give users a sense of control. Word processors are basically micro or mini computer systems with their own disk drives and printers. Compared to a time sharing system with remote (and invisible to the user) storage facilities and frequently their printers. Shneiderman perceives that users have a greater sense of control with the micro and mini systems and, hence, greater satisfaction for similar tasks.

Shneiderman does not speak to the issue of whether users, in fact, wish to be in control. His assumption is that all users will desire to be in control and will wish greater control as they gain experience with a given system. While the desire to be in control may be considered an admirable, or at least a desirable quality, in users there is ample evidence that individuals do differ in their desire to control (Shneiderman, 1980).

A problem with the casual observation of human behavior is that it is easy to accept the commonplace as the norm for behavior. Certainly, many users of computer systems may verbalize their desire for more control; for example, the desire for shortcuts, as they gain experience. But what of the individual who remains frustrated and anxious even after considerable exposure to computer systems? Perhaps some people should be served by man-machine dialogues that meet their need to be directed or guided.

Closure. Closure is the completion of a task leading to relief, in essence when our limited short-term memory is relieved of information that is no longer needed (Shneiderman, 1980). Such relief is often experienced while working with text editors. As one modifies a program or text with an editor there can be considerable anxiety up until the editing session ends with an EXIT or SAVE command. The implication here is that users might be more comfortable with multiple small tasks than with larger singular tasks. Some software vendors now offer the ability to save modified text before ending the editing session, as is the case with XEDIT distributed by the Control Data Corporation (1981). The only research on closure in man-machine dialogues is that on text editors; and even that research offers only conjectures about closure as observed in experiments on other factors.

Response time. Response time is the moment from which a user submits a command or request until the moment the on-line computer system responds (Dock & Essick, 1981). It is not uncommon to hear computer system users ask for faster response time when working with interactive systems. Shneiderman suggests that "a more informed view is that the acceptable response time is a function of the command type" (Shneiderman, 1980, p. 232). That is, certain commands such as those for response light pens, should be very quick while the response to seemingly complex queries should be longer.

Miller has shown however that variability in response time actually causes poorer performance and lower user satisfaction (1968). Users may well prefer a system with a consistent three second response time to one which varies from one to five seconds, even though the average might be the same (Shneiderman, 1979). Shneiderman conjectures that "by eliminating the variance in response time, service is perceived to be more reliable and one source of anxiety can be reduced" (1980, p. 232).

In one experimental study involving the modification of five parameters with light pens the findings showed that the subjects performance improved as response time improved (Goodman & Spence, 1978). However, in another study of subjects performing calculations on numeric arrays, subject performance increased as response time was slowed (Grossberg, Wiesen, & Yntema, 1976). In this latter case, subjects changed their work habits and became more cautious. The subjects, however, took fewer steps to solution of the problems and frequently completed the tasks in the same time.

There seems to be evidence that users work at the speed of the computer system or at least attempt to. In some cases the decision-making time is short, as in data entry, and the response time needs to be as fast as the user. In other cases where the decision making time might be longer, a fast response time could cause anxiety and mistakes. Possibly the combination of proneness to anxiety and internal versus external locus of control could be related to performance of tasks under various response times. If, in fact, people function better with different response times, perhaps computer systems (like many electronic games) should have variable response times either selected by the user or assigned by the computer system according to the number of errors the system detects.

Time-sharing versus Batch Processing. Time-sharing is "the term used to describe a central processing unit that is shared by several users, usually with the use of terminals" (Stern & Stern, 1982, p. 627). Time-sharing environments are on-line environments, that is the "utilization of data processing equipment that is directly under the control of the main central processing unit" (Stern & Stern, 1982, p. 624). On-line implies a users ability to interact directly with a computer.

Batch processing is "the processing of data in groups or batches, as opposed to the immediate processing of data" (Stern & Stern, 1982, p. 617). Batch processing can be characterized in an academic setting by students keypunching programs, submitting the programs to a computer center, and then picking up the output after some period of time. The period of time from submission of a job to receipt of output is termed turnaround time. Turnaround time might range from a few minutes to a few days.

As time-sharing systems came into active use and users were given a choice between batch and time-sharing several studies appeared on the subject (Schatzoff, Tsao, & Wiig, 1967; Gold, 1969; Sackman, 1970a; Sackman, 1970b; Boillot, 1974; Hansen, 1976). One viewpoint was that waiting for batch output was "annoying, disruptive, and time-consuming". Another viewpoint is that time-sharing "encouraged sloppy and hasty programming, which in turn led to more errors and poorer quality work" (Shneiderman, 1980, p. 232).

Studies by Schatzoff, Tsao and Wiig (1967) and by Gold (1969) indicated a higher cost for time-sharing (50 percent increase) and a greater elapsed time for batch jobs (50 percent longer) with no difference in total computer time used. More compilations of programs were observed in the time-sharing mode which might indicate that individuals were not as thorough in checking their work before compiling their programs.

Gold states that "the user's attitude appears to be one of the variables which may influence the user's immediate behavior and usage of computer systems" (1969, p. 255).

In a review of time-sharing versus batch processing experiments, Shneiderman summarizes, "In all the experimental results, the influence of individual differences apparently played a major role. The high variance in performance and conflicting anecdotal evidence suggests that unmeasured factors such as personality may influence preference and performance" (Shneiderman, 1980, p. 233-4).

Lee and Shneiderman conducted studies into locus of control and assertiveness regarding batch versus time-sharing preference (1978). "Locus of control focuses on the perception individuals have of their influence over events. Internally controlled individuals perceive an event as contingent upon their own action, whereas externally controlled people perceive a reinforcement for an action as being more a result of luck, chance, or fate; under the control of other powerful people; or unpredictable" (Rotter, 1966, p. 1).

Assertive behavior "allows an individual expression in a manner that fully communicates his personal desires without infringing upon the right of others". (Winship & Kelly, 1976, p. 215). Weinburg conjectures that "humble programmers perform better in batch environments and assertive ones will be more likely to shine on-line". (Weinberg, 1971, p. 235).

Subjects in the study by Lee and Shneiderman (1978) were professional programmers who had available and worked in both a batch environment and a time-sharing environment utilizing Control Data Corporation equipment. The subjects completed questionnaires to ascertain their assertiveness, locus of control, and preference for batch or time-sharing. The groups by preference did not differ on either personality dimension; but, when grouped by internal locus/high assertive and external locus/low assertive, there were significant differences in mean preference scores. Shneiderman suggests further study with a wider variety of programming environments.

Text Editor Usage. A text editor is a software product that enables users to add, delete, or change lines of text. The text is typically a program but could, also, be data or a written document. Text editors have a command language including instructions to move forward and backward in a file of text and providing an ability to print lines of the text and move text lines from one location to another within the text file. Many of the characteristics of text editors are found in word processing packages.

Walther and O'Neil (1974) conducted an experiment using both an inflexible text editor and a flexible text editor. The flexible version, according to Shneiderman, "permitted abbreviations, default values, user declaration of synonyms, a variety of delimiters, and other features" (Shneiderman, 1980, p. 236). Other variables included attitude towards computers and anxiety, experience with on-line systems, and type of terminal (cathode ray tube versus hardcopy terminal). The subjects were evaluated in terms of errors made and time to completion of a task.

> Experienced users worked faster with the flexible version, but inexperienced users were overwhelmed by the flexible version . . . inexperienced users made fewer errors and worked faster with the inflexible version ... hardcopy terminal users worked faster and made fewer errors suggesting that the feedback from the hardcopy terminal may facilitate performance . . . those with negative attitudes made more errors. (Shneiderman, 1980, p. 236).

Sondheimer (1979) conducted experiments with professional programmers focusing on five features chosen for addition to an existing text editor. Sondheimer concluded that the results of the experiment seemed to indicate the persistence of individual usage habits. The implication from this experiment is that text editing is a skill which once learned is difficult to change.

Experiments in the use of text editors by highly trained individuals have been conducted by Card, Moran and Newell (1978). Their experiments were restricted in scope to present a cognitive model based on 'goals, operators, methods, and selection rules'. The model is meant to represent the performance of expert users. The experimental results cannot be generalized beyond the scope of the experiment. That is, the "user must be an expert, the task must be a routine unit task; the method must be specified in detail; and the performance must be error-free". The results indicate a speed advantage of display editors over line editors and that there are "speed and accuracy advantages of a mouse for selection text, when compared with a joystick, step keys, or text keys" (Shneiderman, 1980, p. 238).

As word-processing systems closely approximate text editors it should be a worthwhile endeavor to replicate the previously discussed experiments with current word-processing software. It should be noted that these experiments did not explore characteristics of the subjects (cognitive or affective dimensions) but only classified subjects as expert users.

Shneiderman offers the following guidelines in developing on-line systems; they may well be considered in experimental designs for research involving such systems,

- do not violate the bounds of human performance imposed by short term memory capacity

- design interactions in a modular fashion to facilitate closure

- be sensitive to user anxiety and desire for control
- provide novice users with a sense of accomplishment, but avoid patronizing comments

- consider response time requirements

- accept the personality and cognitive style differences among individuals and do not attempt to make everyone behave as you do
- make error messages constructive and give guidance for using the system in a courteous non-threatening way

- give users control over what kind of and how much information they wish at every point in the interaction

- have HELP facilities available for every command (Shneiderman, 1979, p. 243).

CHAPTER III

RESEARCH DESIGN AND METHODOLOGY

This chapter presents the design of the study including the sample, the instruments used, the treatment applied, the hypotheses, and the statistical analysis used.

The study was conducted over a two month period. During the first month two dialogues were administered to students of one instructor, David Wilson, hereafter referred to as Instructor A. During the second month the dialogues were administered to the students of Richard Hartley, hereafter referred to as Instructor B. Each instructor had two sections of the same course, "Introduction to Data Processing". The first section of each instructor was given Dialogue I (the system-directed software) first. The second section of each instructor was given Dialogue II (the user-directed software) first. The Rotter Locus of Control instrument was administered prior to the dialogue treatments to determine whether each individual had an Internal Locus of Control or an External Locus of Control.

Sample

The sample for the study included undergraduate students enrolled in sections of ISA 221, "Introduction to Data Processing" at Central Michigan University. This course is required of all students on the Business Administration curriculum and is generally the student's first academic exposure to computer systems. The course is open to students with at least sophomore standing.

All students had instruction on the IBM PC and an assignment to write a BASIC program, hence all students had the same level of exposure to the equipment used in the study. (Knowledge of the BASIC language was not necessary but familiarity with the operation of the IBM configuration was relevant.)

Students are assigned to specific sections of ISA 221 by a computer assisted scheduling system. Each student indicates their course preferences through a pre-registration process. The course requests are processed in an order determined by the student's classification (freshman, sophomore, etc.) and number of earned credits. The scheduling program is designed to balance the number of students in each section of a multi-section course. Hence, there is no predetermination by the students as to the section in which they will be placed.

The course "Introduction to Data Processing" is 16 weeks long and covers three primary topics: hardware, systems analysis, and programming. The students are viewed as end users of computer systems as opposed to computer professionals.

This study utilized an instrument to measure locus of control, two sets of programs to simulate user-directed and system-directed dialogues, and a set of independent questions to ascertain the subject's opinions about the dialogue treatments.

Locus of Control Measure

To measure locus of control, Rotter's Internal-External Locus of Control scale was used (See Appendix A for a copy of the scale.) The scale is a 29 item, forced-choice questionaire including six filler items. Each of the other 23 items offer a choice between an internal and external belief statement (Rotter, 1966).

An internal consistency analysis for reliability yielded r = .70 for males and females. Test-retest reliability coefficients computed after one month were r = .60 for males and r = .83 for females (Ritchie, 1970). Correlations between Rotter's Scale and the Marlowe-Crowne Social Desirability Scale (Crowne, 1964) range from -.07 to -.35.

Dialogues

The dialogues presented to the subjects were developed such that the same task was utilized in both treatments. The task was the entry of payroll data into a computer system. The subjects were provided handouts (See Appendix B for a copy of the hand-outs provided to the students) representing a weekly payroll worksheet on which was recorded the employee number, name, dependents, hours worked, hours sick, and hours on vacation during the week. The subjects were informed that the first ten records had already been entered into the system but errors were made as indicated by the circled items. The employees below the double line had not yet been entered. The subjects were instructed to correct the errors previously made and add the data not already on file. When the corrections and additions were completed the subjects were to produce a listing of the data and the accompanying totals. If the totals were correct the task was complete. If the totals were not correct the students were to correct any errors until correct totals were obtained.

The two dialogues differed only in the level of control the user had over the task. In order to ensure that they differed only in terms of control, two steps were taken after the first set of dialogues were developed. In the first step four pyschologists from the Psychology Department at Central Michigan University used the two dialogues and then provided their independent suggestions for changes. After modification the dialogues were administered to a group of 55 students. After using both dialogues the students were asked to submit written evaluations of each dialogue including a discussion of the degree of control they felt they had with each dialogue. The dialogues were then modified and again reviewed with two of the psychologists.

<u>Dialogue I</u>. Dialogue I represented the system- directed environment. The student was given the opportunity to add, delete, or change records but only in that order. The students were first presented with a screen display

which asked the student to supply the data for a new record, as shown in Appendix C. When all the additions had been entered the student entered the word STOP to end the addition routine. The system then automatically sorted the file. While sorting, a message appeared on the screen informing the student that the payroll file was being sorted by employee number. This step was necessary as the student might have added records out of sequence. Later routines that searched the file by employee number required the file to be in employee number order.

Once the file was sorted the student was asked if there were any records to be deleted. The student would either enter an employee number or the word STOP. The student needed this ability to delete records that were inadvertently entered twice, or to delete employees entered with an incorrect employee number.

Next the change routine was presented to the student. The student entered the employee number of the record to be changed and was presented with the current contents of the record. The system then asked the user, one field at a time, if a change was needed. If the student responded with a Y answer the existing data were displayed on the bottom of the screen together with a place to enter the new data. When the student had completed all changes the student typed the word STOP rather than an employee number.

Once the student ended the change routine the system automatically printed a listing of the file. While listing the file the system calculated the totals and displayed on the screen messages indicating whether each total was correct or incorrect. If any total was incorrect the system again presented the add, delete, and change routines, and then produced a new listing. The sequence was always the same regardless of what the student needed to do next.

Dialogue II. The second dialogue was written in a manner that gave the user control over the data entry and modification process. This dialogue provided a command driven interaction. The student specified one of seven commands defined by the first screen display. (The HELP command was available to display this first screen again to remind the students of available commands.) With this dialogue the student specified only those routines needed and in the order desired by the student. For example, the student might have desired to change records first, then add the remaining records. Or the student may have listed the file before beginning.

The ADD routine was the same as the first dialogue except that the student was allowed to "backup." That is, if the student discovered that an item previously entered was in error, the student could go back to that field and reenter the data. (In the first dialogue it was necessary to enter all the data for an employee and then reply with an N to the question "Is the record correct (Y/N)?". Once the negative response was entered, it was necessary to enter the entire record again.) The word STOP was used in both routines to terminate the addition routine.

The DELETE, CHANGE, and LIST commands in the user-directed dialogue provided additional control over the process by allowing the user to specify which records were to be processed. With each of these commands the student was asked "Enter mode: (A)ll, (O)ne, (R)ange, (K)ey -". The student entered the appropriate letter; A, O, R, or K. The A option implied that all records would be affected by the operation, that is, all records would be deleted, listed, or presented for changes. The O option allowed the student to refer to records by their position in the file. After specifying O, the student was asked for the record number. The number entered indicated the position of the record in the file, for example a 3 referred to the third record. The R option allowed the student to specify the range of relative record numbers to process. The student was asked for the lower and upper bounds. For example, if the range was from 2 to 6, then the second through the sixth records were processed. This was particularly advantageous for the change routine as only the second through the sixth records of the assignment needed to be changed. The last option, K, allowed the student to specify the employee number, or key, of a specific record. Only the record corresponding to the key specified was processed.

The DELETE command caused the deletion routine to be invoked. The user was asked which mode of processing was required and the corresponding records were deleted. If the key mode was selected the student was informed that the file must be in key order. If records were not in sequence the SORT command was used by the student to arrange the records in employee number order.

The CHANGE command allowed the student to change any fields within a record by entering the name of the field. After each field was changed the student could enter STOP to end the changes for the record or could continue to change other fields by specifying additional field names. The change routine could be used to process specific records or groups of records as described previously. The LIST command provided the student with the ability to display records on the screen or the printer. The format of the display was either a block format that resembled the display used in the ADD and CHANGE routines, or the format could have been defined by the student. The two format options were presented by the question -

(B)lock or (F)orm Listing?

If the F option was selected the student was asked for the fields to be included on the listing, whether totals of the columns were to be prepared, and what the title of the report should be. When the listing was complete the student was prompted for the next command.

Once the student had determined that the totals were correct, the student entered the command END.

Opinion Questions

At the end of each of the two dialogues, eight questions were presented to the student one at a time to determine the students' feelings towards the dialogue just used (See Appendix D for a copy of the questions).
The first three questions asked the student to indicate their feelings concerning the ease of use of the ADD routine, the CHANGE routine, and the overall system, respectively. The responses ranged from "easy to use", to "very difficult to use". The second three questions were used to assess the anxiety of the student while using the system and were asked relative to the same three categories. The valid responses were "very comfortable", "comfortable", "neutral", "frustrated", and "very frustrated". The seventh question asked the student to assess their own typing proficiency on a scale of 1 to 5 where a "1" indicated the minimum proficiency and a "5" indicated a proficiency above fifty words per minute. The eighth question asked the student to evaluate the software in terms of "user-friendliness" ranging from "user-friendly" to "not at all user-friendly. The questions were forced choice in that the system continued to ask for a response until the student entered a valid response.

Administration of the Experiment

Two weeks before the use of either dialogue the Rotter Locus of Control instrument was administered in the classroom. As the locus of control instrument was not characteristic of the subject matter of the course, an explanation was given such that the students would not relate the control issue to the dialogue treatment. The rationale for administering the questionaire was that the Information Systems and Analysis department was in the process of developing a personality profile of the students typically enrolled in the course. The two week time period between administering the questionaire and the treatments served to further disassociate the personality aspect from the dialogue assignments.

The equipment used was the IBM PC (personal computer) with a Microline Okidata printer. Each computer had 64K memory, two disc drives and a monochrome monitor. There were eight computer systems available, all within the Grawn Hall Computer Lab at Central Michigan University.

The students were introduced to the dialogue experiment as a class assignment that would give them an opportunity to assess two forms of man-machine interactions that they, or their future employees, would likely encounter. Completion of the exercise, including the written assessment, constituted a regular assignment on a credit/no-credit basis. The dialogues were presented to the students via a set of overheads that showed each screen display that the students would encounter (See Appendix C for a copy of the overheads used). Questions posed by the students were recorded so that the same information could be conveyed to all groups.

At the time of the presentation the handout titled

"MAN-MACHINE DIALOGUE ASSESSMENT" was given to all groups together with the procedures for one of the dialogues. One section of each instructor began with the system-directed dialogue; the other section began with the user-directed dialogue. The dialogue instructions were labelled DIALOGUE I and DIALOGUE II respectively. No mention was made of the terms system-directed or user-directed.

When a student finished one dialogue the printout, with totals, was handed in, and the student was given the instructional handout for the opposite dialogue. When the second dialogue was completed the student turned in the corresponding printout, the written assessment and preference, and the diskette used by the student.

Recorded on each student's diskette was the number of steps taken, the elapsed time from beginning to end of the task, and the responses to the eight questions.

Following the treatments the aforementioned data together with the sex of the subjects (obtained from registration data) were assembled via a number of programs to create a composite file of all the data.

The final data file recorded on the student's diskette contained the following items:

Social Security Number

Preference (1 = Dialogue I, 2 = Dialogue II)

Sex (1 = Male, 2 = Female)

Locus of Control (Score of 0 to 30)

Steps for each dialogue (The summation of the number of times the ADD routine was invoked, the number of times the CHANGE routine was invoked, and the number of times the ENTER key was pressed)

Responses to Questions 1 through 8 for each dialogue (Values were 1 through 5).

Hypotheses

The hypotheses which were the basis for analysis of this study are as follows:

1. The internal group will prefer the user-directed software more often than the external group; the external group will prefer the system-directed software more often than the internal group.

2. The mean times for completion of the task will differ between internal and external locus groups when using the system-directed software.

3. The mean number of steps taken for completion of the task will differ between internal and external groups when using the system-directed software.

4. The internal group's mean score for time to completion will be less than the external group's mean scores for the user-directed software.

5. The internal group's mean number of steps will be less than the external group's mean number of steps for the user-directed software.

Statistical Analysis

The first hypothesis was tested using the Chi-square statistic and a significance level of .05. The SPSS subprogram CROSSTABS was used to produce a table of locus of control (internal, external) by preference (Dialogue I - system-directed, Dialogue 2 - user-directed).

The second hypothesis was tested by the two-tailed t-test for independent samples using a significance level of .05. The variables involved were locus of control and time to completion for each dialogue.

The third hypothesis was also tested by the t-test for independent samples using a significance level of .05. The mean number of steps was established by adding the number of times the add, delete, and change routines were invoked to the number of items entered. The test was a comparison of the mean number of steps taken when using Dialogue I, the system-directed software.

The fourth hypothesis was tested in the same manner as the second hypotheses using the t-test for independent samples. The mean time scores of Dialogue II for the internal locus subjects was compared to the scores for the external locus subjects.

The fifth hypotheses was also tested using the t-test for independent samples. A .05 significance level was used to reject the hypothesis. The variables were the mean number of steps for internal locus subjects compared with the mean number of steps for the external locus subjects in completing the assignment for Dialogue II, the user-directed software.

CHAPTER IV

ANALYSIS OF DATA

This chapter provides an analysis of the data collected during the study. As outlined in Chapter III, the sample consisted of four sections of the course "Introduction to Data Processing". The sections were further divided on the locus of control variable. The median score on the Rotter locus of control instrument was used to classify the subjects as having an internal locus of control (0-10) or an external locus of control (11-30), hereafter referred to as internal and external, respectively. Table 2 reflects the composition of each section in terms of the locus of control attribute.

Table 2

Number of Study Participants

	Instructor A		Instructor B		Total
Locus of Control	S/U	U/S	S/U	U/S	
Internal	22	22	26	21	91
External	26	24	30	34	114
Total	48	46	56	55	205

Note. S/U = System-directed dialogue was used before the user-directed dialogue.
U/S = User-directed dialogue was used before the system-directed dialogue.

The locus of control instrument was completed by 205 students; of those 183 also completed the dialogue assignments. One student completed the dialogue assignments who did not complete the locus of control instrument. That student was included in all analyses except those involving the locus of control variable.

The performance data were collected via diskettes supplied by the students. Apparent inconsistencies in the number of students included in each analysis resulted from students not typing "END" at the end of the second dialogue. Without "END", the computer software did not write the question file or the performance file to the student's diskette for the second dialogue. Hence, there are differences in the number of students reported for each dialogue.

While it was possible to have the students repeat the second dialogue, it was the judgement of the researcher that misleading performance data might result because of the additional practice obtained by the student. Therefore, the students with incomplete data for the second dialogue were not asked to repeat the assignment.

Hypothesis Tests

The first hypothesis stated:

> The internal group will prefer the user-directed software more often than the external group; the external group will prefer the system-directed software more often than the internal group.

This hypothesis was tested using a Chi-square test of independence. The SPSS procedure CROSSTABS was used to produce a 2 X 2 table with one dimension being locus of control (internal, external) and the other dimension being preference (system-directed, user-directed) as shown in Table 3. The test resulted in a Chi-square value of .66 which was not significant (p = .41). Clearly there was no support for the hypothesis.

Table 3

Locus of Control by Preference

	Preference		
Locus of Control	System Directed	User Directed	Total
Internal	42	42	84
External	56	42	98
Total	98	84	182

Note. Chi-square = .66 Significance = .41

The second hypothesis was stated as

> The mean times for completion of the task will differ between internal and external groups when using the system-directed software.

The second hypothesis was tested with a two-tailed, independent t-test. The independent variable was locus of control and the dependent variable was time to completion for the system-directed dialogue. The completion times reflected the number of minutes that elapsed from the time the student started the dialogue until the correct answers were obtained. The times recorded did not include the time necessary to answer the questions at the end of the dialogue exercise. Table 4 presents the means and standard deviations for the internal and external groups.

Table 4

Completion Times for the System-Directed Dialogue Task

Locus of Control	Number of Cases	Mean Times	SD
Internal	88	29.92	14.19
External	101	27.22	9.02

A test for homogeneity of variance yielded an F value of 2.48 with a probability of .000. Therefore a t-test was performed using a separate variance estimate for determining the standard error term. The *t* value was 1.53 (*p* = .128). Again the hypothesis was not supported.

The third hypothesis was:

> The mean number of steps taken for completion of the task will differ between internal and external groups when using the system-directed software.

The third hypothesis was also tested using a t-test for independent samples. The independent variable was again locus of control. The dependent variable was the number of steps used to complete the task using the system-directed software. The number of steps was calculated by adding the number of times the ENTER key was used to the number of times the ADD, DELETE, CHANGE, and LIST routines were invoked. The means and standard deviations are shown in Table 5. The independent t-test yielded a *t* value of -1.12 (p = .266). The hypothesis was not supported.

Table 5

Steps Taken to Complete the System-Directed Task

Locus of Control	Number of Cases	Mean Number of Steps	SD
Internal	88	184.69	19.08
External	100	188.17	23.10

The fourth hypothesis stated:

> The internal group's mean score for time to completion will be less than the external group's mean scores for the user directed software.

The fourth hypothesis was tested with a one-tailed, independent t-test. The completion times recorded for the user-directed dialogue for the internal students were compared to the completion times for the external students. Relevant means and standard deviations are given in Table 6.

The *t* value for the difference between means was 1.03 (p = .153). The fourth hypothesis was not supported.

Table 6

Completion Times for the User-Directed Task

Locus of Control	Number of Cases	Mean Time	SD
Internal	77	34.75	15.11
External	88	32.36	14.70

The fifth hypothesis was:

> The internal group's mean number of steps will be less than the external group's mean number of steps for the user-directed software.

The fifth hypothesis was also tested using a one-tailed t-test for independent samples. The dependent variable was the number of steps needed to complete the task with the user-directed software. The number of steps was again calculated by adding the number of times the ENTER key was used to the number of times the ADD, DELETE, CHANGE and LIST routines were invoked. Means and standard deviations are given in Table 7. The analysis yielded a t of .68 (p = .249). Hence the fifth hypothesis could not be supported.

Having failed to support any of the hypotheses a possible explanation was sought. One possible explanation was that the locus of control groups were not significantly different from each other because they were split at the median.

Table 7

Number of Steps for the User-Directed Task

Locus of Control	Number of Cases	Mean Steps	SD
Internal	73	126.58	34.45
External	84	122.75	35.35

The groups were redefined to include only the upper and lower 27 percent, as has been suggested by Kelly (1939). The test for each of the hypotheses was repeated using the newly defined, smaller groups. The results are shown in Tables 8 and 9. With a Chi-square of .115 (p = .734) and obtained t's ranging from -1.38 to +1.08 (p ranged from .172 to .364), all of the tests again proved non-significant. Even with a greater differentiation between internals and externals, none of the hypotheses was supported.

Table 8

Locus of Control by Preference Using Upper and Lower 27%

Locus of Control	Preference System-Directed	User-Directed	Total
Internal	19	23	42
External	25	24	49
Total	44	47	91

Note. Chi-square = .115 Significance = .734

Table 9

Performance for Upper-Lower (27%) Locus of Control Groups

Performance Measure	Number of cases	Mean	SD	t	df	p
Time for System-Directed Task						
Internal	45	29.22	11.79	1.09	95.00	.280[a]
External	52	26.90	9.20			
Time for User-Directed Task						
Internal	40	36.20	17.96	.35	80.00	.364[b]
External	42	34.83	17.51			
Steps for System-Directed Task						
Internal	45	182.46	15.43	-1.38	90.94	.172
External	51	187.62	21.14			
Steps for User-Directed Task						
Internal	37	135.40	46.01	.75	75.00	.227
External	40	127.77	42.91			

Note: [a] = one-tailed test
[b] = two-tailed test

Supplementary Analyses

Order Effect. T-tests were done to determine if the order of the treatments had any effects on the performance by the subjects. Performance was measured in terms of time to completion and in terms of the number of steps necessary to achieve the correct totals on the data entry task.

The subjects were classified by the dialogue with which they began the experiment. Table 10 shows the results of the comparison of performance measures. With t values ranging from -1.07 to +1.52 (p ranging from .131 to .747), it was obvious that there were no significant differences. The order of the treatment did not have an effect on performance.

The preference for either dialogue could also have been affected by the order in which the treatments were administered. A 2 X 2 table with one dimension being the order of the treatment (system-directed/user-directed, user-directed/system-directed) and the other dimension being preference (system-directed dialogue, user-directed dialogue) was constructed (Table 11). A Chi-square test revealed no support for preference being dependent upon the order of the treatments.

Instructor Effect. Next the data were analyzed to determine if the classroom instructor had an effect on the performance of the students with either treatment.

Table 10

Effect of Treatment Order on Performance

Performance Measure	Number of cases	Mean	SD	t	df	p
Time for System-Directed Task						
S/U	102	29.61	12.18	1.52	189.00	.131
U/S	89	27.04	11.07			
Time for User-Directed Task						
S/U	90	32.32	13.20	-1.07	139.70	.286
U/S	75	34.86	16.69			
Steps for System-Directed Task						
S/U	101	188.26	22.49	1.30	188.00	.196
U/S	89	184.25	19.72			
Steps for User-Directed Task						
S/U	87	125.35	34.30	.32	155.00	.747
U/S	70	123.51	36.81			

Note. S/U = System-directed task was performed before the user-directed task.
U/S = User-directed task was performed before the system-directed task.

Table 11

Effect of Treatment Order on Preference

Preference	Order of Treatment S/U	U/S	Total
System-Directed	52	47	99
User-Directed	48	36	84
Total	100	83	183

Note. S/U = System-directed dialogue was used first.
U/S = User-directed dialogue was used first.
Chi-square value of .226
Significance = .633

The students were grouped by the instructor for the section in which the students were enrolled. Time to completion of each task and the number of steps taken to complete each task by instructor group, constituted the performance data shown in Table 12. With t values from .65 TO 1.60 (p from .113 to .519) it was apparent that there was no support for the hypothesis that the instructor of the course had influenced the performance of the students.

The effect of the course instructor on the students' preference was also analyzed. The analysis was a 2 X 2 Chi-square with one dimension being the course instructor (Instructor A, Instructor B) and the other dimension the student's preference (system-directed, user-directed). The data are given in Table 13. The test yielded a Chi-square of .000 (p = 1.0), thus there was no support for the hypothesis that preference was

Table 12

Effect of Course Instructor on Performance

Performance Measure	Number of cases	Mean	SD	t	df	p
Time for System-Directed Task						
Instructor A	94	28.97	11.81	.65	189.00	.517
Instructor B	97	27.87	11.65			
Time For User-Directed Task						
Instructor A	72	34.33	16.49	.65	163.00	.519
Instructor B	93	32.81	13.59			
Steps for System-Directed Task						
Instructor A	94	187.43	20.38	.67	188.00	.504
Instructor B	96	185.36	22.17			
Steps for User-Directed Task						
Instructor A	68	130.01	44.43	1.60	101.06	.113
Instructor B	89	120.34	25.88			

dependent upon the instructor of the course.

Table 13

Effect of Course Instructor on Preference

Preference	Course Instructor: Instructor A	Instructor B	Total
System-Directed	45	54	99
User-Directed	39	45	84
Total	84	99	183

Note. Chi-square = .0 Significance = 1.0

Sex of the Subjects. The effect of the subjects' sex (male, female) upon their preference for either dialogue was analyzed using a Chi-square statistic. The 2 X 2 table was constructed with sex as one dimension and preference as the other dimension. As can be seen the results from Table 14 indicated that the sex of the individual did not relate to preference.

Table 14

Effect of Sex on Preference

Preference	Sex: Male	Female	Total
System-Directed	49	50	99
User-Directed	40	44	84
Total	89	94	183

Note. Chi-square = .010 Significance = .916

The performance of the subjects was also analyzed by sex (male, female) to determine if either sex varied significantly in terms of time to completion of each task or in terms of the number of steps used to complete each task. Table 15 gives the means, standard deviations, and the results of the analysis. It was found that the females differed significantly from the males in terms of time for completion of the system-directed task (t = 2.96, p = .004) and they were significantly faster in terms of time to completion of the user-oriented task (t = 2.77, p = .006). The females did not differ significantly from the males in terms of the number of steps to complete either dialogue task.

As there was a significant difference in time between the male and female groups but no significant difference in the number of steps used to complete each task, it was hypothesized that one group possessed better typing skills than the other. The hypothesis was tested by comparing males and females on the typing level reported via question seven (Appendix D) at the end of each dialogue. The means and standard deviations are shown in Table 16. A one-tail t-test for independent samples produced a significant statistic (p = .000). The t-value indicated females reported having better typing ability than did males. An examination of reported typing ability therefore verified that there was a significant difference between males and females.

At the end of each dialogue task seven other questions

Table 15

Effect of Sex on Performance

Performance Measure	Number of cases	Mean	SD	t	df	p
Time for System-Directed Task						
Male	94	30.91	12.47	2.96	188.00	.004
Female	97	26.00	10.44			
Time for User-Directed Task						
Male	81	36.69	15.91	2.77	163.00	.006
Female	84	30.38	13.22			
Steps for System-Directed Task						
Male	93	186.33	20.46	-.04	188.00	.972
Female	97	186.44	22.13			
Steps for User-Directed Task						
Male	75	125.74	36.24	.41	155.00	.683
Female	82	123.42	34.67			

Table 16

Relationship of Sex to Reported Typing Skill

Performance Measure	Number of cases	Mean Response	SD	t	df	p
Reported Typing Level at the end of the System-Directed Task (1-5)						
Males	94	2.62	1.05	-.6.57	188.00	.000
Females	96	3.68	1.16			
Reported Typing Level at the end of the User-Directed Task (1-5)						
Males	85	2.65	1.08	-5.28	172.00	.000
Females	89	3.56	1.16			

were asked in addition to the question on typing ability (See Appendix D). The questions were included in the experiment to explore student perceptions of the terms "ease of use", "anxiety", and "user-friendliness" as they relate to man-machine dialogues. As the dialogues of this study only differed in the control dimension, responses to the questions could be useful in defining the aforementioned terms as they relate to software dialogues.

Ease of Use Questions. The first three questions were asked to determine the difficulty of using each dialogue. It was predicted that internal individuals would find the user-directed software easier to use than the system-directed software. It was also thought that the external group would find the system-directed software easier to use than the user-directed software.

The ease of use questions were analyzed by a one-tailed t-test for independent samples. As is shown in Table 17, there were no significant findings for the ease of use questions asked after the system-directed task (t = -.84 to .07, p from .201 to .471). However the internal group rated the ADD command of the user-directed software as easier to use more often than did the external group (t = -1.67, p = .049). There was not a significant difference in ease of use responses for the CHANGE command or for the overall user-directed system between the internal and external groups (t = -1.51 to -1.22, p = .066 to .111).

The responses to the ease of use questions were also

Table 17

Relationship of Locus of Control to Ease of Use

Performance Measure	Number of cases	Mean Response	SD	t	df	p
Ease of Use - Question 1 (Add Routine)						
System-Directed Dialogue						
Internal	88	1.45	.843	-.84	187.00	.201
External	101	1.55	.793			
User-Directed Dialogue						
Internal	80	1.61	.893	-1.67	171.00	.049
External	93	1.89	.966			
Ease of Use - Question 2 (Change Routine)						
System-Directed Dialogue						
Internal	88	1.61	.952	.07	187.00	.471
External	101	1.60	.895			
User-Directed Dialogue						
Internal	80	1.80	.947	-1.51	171.00	.066
External	93	2.03	1.058			
Ease of Use - Question 3 (Over-all System)						
System-Directed Dialogue						
Internal	88	1.60	.810	-.10	187.00	.460
External	101	1.61	.787			
User-Directed Dialogue						
Internal	80	1.92	1.065	-1.22	171.00	.111
External	93	2.11	1.009			

analyzed in terms of the stated preference for either the system-directed or user-directed software. A one-tailed t-test for independent samples was used to compare the preference groups on their responses to each question. Table 18 shows the means and standard deviations. In each test the results were significant (_t_ ranged from -2.38 to 5.67, _p_ ranged from .000 to .008).

Anxiety Questions. Questions 4, 5, and 6 were asked of the students to record their level of anxiety in relation to the ADD routine, the CHANGE routine, and the overall system. The responses to each question were analyzed by preference group with a one-tailed t-test for independent samples as shown in Table 19. There was not a significant relationship between preference and the responses to the anxiety question for the ADD routine of the system-directed dialogue (_t_ = -1.55, _p_ = .061). However, there was a significant difference in the responses to each of the other anxiety questions (_t_ = -2.17 to 5.19, _p_ = .000 to .017). The hypothesis that preference was related to the level of perceived frustration was supported in all but one instance -- the effect, upon preference, of the ADD routine of the system-directed dialogue.

The same anxiety questions were then evaluated in relation to locus of control. It was expected that the internal group would rate the system-directed software as more frustrating than would the external group; similarly it was hypothesized that the external group would rate the

Table 18

Relationship of Preference to Ease of Use

Performance Measure	Number of cases	Mean Response	SD	t	df	p
Ease of Use - Question 1 (Add Routine)						
System-Directed Dialogue						
Internal	99	1.33	.728	-2.83	178.00	.002
External	81	1.66	.851			
User-Directed Dialogue						
Internal	92	1.90	.961	3.17	164.86	.001
External	75	1.48	.760			
Ease of Use - Question 2 (Change Routine)						
System-Directed Dialogue						
Internal	99	1.44	.798	-2.41	150.33	.008
External	81	1.77	1.012			
User-Directed Dialogue						
Internal	92	2.18	1.068	4.15	163.54	.000
External	75	1.58	.790			
Ease of Use - Question 3 (Over-all System)						
System-Directed Dialogue						
Internal	99	1.43	.657	-2.75	147.35	.003
External	81	1.75	.859			
User-Directed Dialogue						
Internal	92	2.35	1.044	5.67	163.66	.000
External	75	1.56	.775			

Table 19

Relationship of Preference to Anxiety

Diaglogue[1] Preference Group	Number of cases	Mean Anxiety Response	SD	t	df	p
Anxiety - Question 4						
(Add Routine)						
System-Directed Dialogue						
Preferred System-Directed	99	1.68	.765	-1.55	178.00	.061
Preferred User-Directed	81	1.87	.872			
User-Directed Dialogue						
Preferred System-Directed	92	1.96	.931	3.22	164.06	.002
Preferred User-Directed	75	1.56	.702			
Anxiety - Question 5						
(Change Routine)						
System-Directed Dialogue						
Preferred System-Directed	99	1.80	.911	-2.12	178.00	.017
Preferred User-Directed	81	2.09	.917			
User-Directed Dialogue						
Preferred System-Directed	92	2.32	1.039	4.69	159.45	.000
Preferred User-Directed	75	1.69	.697			
Anxiety - Question 6						
(Overall System)						
System-Directed Dialogue						
Preferred System-Directed	99	1.72	.726	-2.17	178.00	.015
Preferred User-Directed	81	1.97	.806			
User-Directed Dialogue						
Preferred System-Directed	92	2.36	.991	5.19	164.40	.000
Preferred User-Directed	75	1.66	.759			

Note. [1] Dialogue refers to the dialogue that was being used when the anxiety question was being asked.

user-directed software as more frustrating than would the internal group. The data to test the hypotheses are shown in Table 20. Neither hypothesis was supported (*t* = -1.27 to -.08, *p* = .102 to .468).

User Friendly Questions. Question number 8 asked the student to rate each dialogue in terms of user-friendliness. It was expected that the system-directed dialogue would receive a higher user-friendly rating (lower score) by those preferring that dialogue than it would from the group preferring the user-directed dialogue. Table 21 illustrates that the difference in response scores between the preference groups was not significant for the system-directed dialogue (*t* = -1.52, *p* = .065).

However when the same question was asked at the end of the user-directed dialogue the results were significant (*t* = 3.68, *p* = .000). That is, those who preferred the user-directed dialogue gave the dialogue a better user-friendly rating than those who preferred the system-directed dialogue.

Finally the internal/external groups were compared on their responses to the question of user-friendliness (Table 22). There was no significant difference between the responses to the question asked at the end of the system-directed dialogue (*t* = -.73, *p* = .232) but there was a significant difference (*t* = 1.69, *p* = .047) between the groups on the user-friendly question when asked at the end of the user-directed dialogue.

Table 20

Relationship of Locus of Control to Anxiety

Dialogue[1] Locus of Control	Number of cases	Mean Anxiety Response	SD	t	df	p
Anxiety - Question 4 (Add Routine)						
System-Directed Dialogue						
Internal	88	1.72	.854	-.87	187.00	.193
External	101	1.83	.801			
User-Directed Dialogue						
Internal	80	1.72	.856	-1.25	171.00	.107
External	93	1.89	.902			
Anxiety - Question 5 (Change Routine)						
System-Directed Dialogue						
Internal	88	1.93	.980	-.28	187.00	.388
External	101	1.97	.877			
User-Directed Dialogue						
Internal	80	1.96	.906	-1.27	171.00	.102
External	93	2.15	1.021			
Anxiety - Question 6 (Overall System)						
System-Directed Dialogue						
Internal	88	1.85	.851	-.08	187.00	.468
External	101	1.86	.679			
User-Directed Dialogue						
Internal	80	2.00	.994	-1.21	171.00	.114
External	93	2.18	.988			

Note. [1] Dialogue refers to the dialogue used at the time the anxiety question was asked.

Table 21

Relationship of Preference to User-Friendliness

Dialogue Preference Group	Number of cases	Mean Response	SD	t	df	p
User-Friendly - Question 8						
System-Directed Dialogue						
Preferred System-Directed	99	1.48	.747	-1.52	178.00	.065
Preferred User-Directed	81	1.65	.744			
User-Directed Dialogue						
Preferred System-Directed	92	2.07	.997	3.68	162.67	.000
Preferred User-Directed	75	1.58	.718			

Table 22

Relationship of Locus of Control to User Friendliness

Dialogue Locus of Control	Number of cases	Mean Response	SD	t	df	p
User-Friendly - Question 8						
System-Directed Dialogue						
Internal	88	1.53	.742	-.73	187.00	.232
External	101	1.61	.748			
User-Directed Dialogue						
Internal	80	1.75	.907	-1.69	171.00	.047
External	93	1.98	.950			

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

The theory on which this study was based was that performance in a man-machine interaction is dependent upon 1) an individual's cognitive abilities, 2) an individual's short term memory capacity, and 3) the degree of anxiety produced by the interaction. This study focused on anxiety in the man-machine interaction.

It has been suggested by the literature that individuals with different personality attributes will respond differently to the same computer environment. That is, if the software is not congruent with the personality of the individual, anxiety will result. It has also been suggested that one characteristic of computer software, control, might well be a factor in causing anxiety in man-machine dialogues. In order to test the aforementioned performance theory, that is, that anxiety in a man-computer interaction has an effect on performance, a situation was constructed to induce anxiety using the locus of control characteristic.

Two computer software dialogues were developed such that one dialogue offered limited control to the user and one dialogue gave the user considerable control over the interaction. Hence two computer environments were created to satisfy two different groups of users. The users were then classified as internal (those who desire control) and external (those who do not desire control), based on scores from the Rotter Locus of Control Scale.

Performance was measured in terms of the time to completion of each task and in terms of the number of steps taken to complete each task. It was found that there were no significant differences in performance between internal and external groups when the groups were defined by the median locus of control score. The locus of control groups were then redefined to include only the extreme upper and lower 27 percent and the tests conducted again. Even with only the most internal and most external individuals included there was no evidence to support the theory that performance varies because of anxiety as produced by personality differences.

As the results did not support the theory several questions need to be answered at this juncture. Were the study instruments sound? Were the performance measurements valid? Were the tasks appropriate? And finally, was the study a valid test of the theory or is the theory wrong?

Locus of Control

The locus of control scale by Rotter was designed to measure generalized expectancies of individuals. If the instrument is an accurate reflection of locus of control, as the literature suggests, then it must be concluded that locus of control is not a factor in situations where individuals interact with computers. However, it is possible that those who are generally internal in other situations are in fact not as internal when working with computers or that those who are external in the general sense are not as external when working with computers. That is, the locus of control measure may not be situation specific. What is needed is an instrument that can accurately assess locus of control in specific situations.

It is also possible that the desire for control in a man-machine interaction is a function of the user's familiarity with the task and the software employed. Individuals, whether they be internal or external in a general sense, may well react differently when working with computers. In order to pursue this thought it would be necessary to provide users with software that allows them to select the amount of control they desire and to observe their choices over a period of time. What is important to note is that the generalized measure of locus of control did not materialize as relevant to performance in the use of either the limited control (system-directed) or the control (user-directed) dialogue.

Performance Measures

Performance was measured in two dimensions, time and steps to completion of the task. The time recorded was the number of minutes that elapsed from the time the exercise was started until correct totals for the task were achieved.

The time was recorded by the computer software and could not be altered by the user. As can be seen from the analysis, females completed the assignments in less time than males. As the completion time can be affected in this type of study by typing ability subjects should be tested for typing proficiency prior to the experimental treatment to control for this variable.

While the difference in mean completion times for males and females has been attributed to typing ability, this is not the only conclusion that is possible. Other differences between males and females should be explored to determine if there are other important differences that could affect time to completion.

Typing ability could also have an impact on performance other than the obvious ability to enter data at a faster rate. It is possible that limited typing ability might cause frustration in a man-machine interaction. If such frustration exists comprehension of the syntactic and semantic requirements of the dialogue could be affected and hence performance could be influenced as well.

Performance was also measured in terms of the number of steps needed to complete the task. The number of steps required was the sum of the number of times each routine was used and the number of times the enter key was pressed. The enter key was used after any data was entered at the keyboard. If a field was changed or added the enter key was pressed, hence the amount of activity within a routine was measured. The step measurement was therefore an accurate reflection of the total amount of activity but was not useful for measuring activity within each routine. It is suggested that in future studies the activity in each routine should be measured independently so that more detailed analysis would be possible.

With the system-directed dialogue the user did not have alternative ways to complete the task, hence the number of steps was strictly a function of the number of errors made by the student. With the user-directed dialogue the student had options within the change and list routines. For example, if the student elected to change a range of records rather than specific records, several fewer steps were needed. The step count for the user-directed dialogue did not differentiate between steps taken because of errors and steps taken because of the approach to solving the problem. Future studies should measure performance in a manner that would allow analysis of both variables. Such analysis would have been mandatory if the purpose of this study had been to analyze the method used to complete the task. Although the performance measures could be improved, they did adequately measure performance for the purposes of this study. They did not contribute to the negative findings of this study.

Dialogue Routines

The system-directed software was developed such that the user had very limited control over the task to be completed, whereas the user-directed software was designed to give the user considerable control. The degree to which each dialogue actually conveyed a sense of control to the student could have had a profound effect on the results of the study. Although the pilot study and evaluations by the psychologists indicated that the dialogues were effective in isolating control, verbal responses from participants raise some questions relative to the change routine. As stated previously, in the system-directed dialogue the student could only change a specific employee record and after specifying that record the student had to answer yes or no to each field displayed on the screen. The presentation of the material on the screen in this dialogue was more pleasing to many students than the method of presentation in the user-directed dialogue. While the format of the screen did not affect the amount of control the student actually had, it could have had an affect on their attitude toward the software and hence their performance as well as their preference. Also, had the preference question been subdivided into several questions, more could have been learned about the specific routines employed and hence the overall preference for either dialogue.

In the change routine of the user-directed dialogue the student had several options including the ability to change a range of records (the most efficient method) or of changing a specific record. Regardless of the choice made the student next specified fields to be changed by typing the name of the field to be changed. Even though the user-directed dialogue enabled the student to change data more quickly many students expressed a dislike for typing in field names. Software that would allow the same degree of control but would perhaps accept shorter abbreviations or codes representing the field names would alleviate this criticism. For example, VAC might be used in place of VACHRS, or REG might be used in place of REGHRS. It is not likely that the specification of the whole field name adversely affected the results.

Treatment Task

The task performed using each dialogue involved the adding, changing, deleting, sorting, and listing of records. Such activities are common to a data entry operation, and are usually performed by clerical personnel. The task was well suited to this study because it was easily learned and did not introduce extraneous variables that a more complex task might involve. The task itself however raises several concerns. As the task was rather elementary in terms of the skills required there might have been inadequate motivation to perform at one's capacity. This is a particular concern with the user-directed task where alternative methods of completing the task were possible. There was no reward provided for superior performance except the satisfaction of completing the task. Had the assignment grade taken into account the number of steps to completion, performance might have improved. With the system-directed dialogue the only way to reduce the number of steps would be to make fewer mistakes. With the user-directed dialogue either fewer errors or a better choice of options would have yielded fewer steps to completion. Increased incentive for improving performance would have in turn increased the possible anxiety to a level that might have affected the overall performance and hence the outcome of the study.

Self-Report Questions

After each dialogue seven questions were asked relating to ease of use, anxiety, and the user-friendliness of the software. The questions were intended to capture the attitude of the subjects towards each dialogue at the time the subjects were using the computer software.

The questions were analyzed in terms of dialogue preference and locus of control. While there were significant findings relative to locus of control the reader is cautioned that the validity of the internal, external classifications in the man-computer interaction might well be questioned.

Ease of Use. There were three ease of use questions for each dialogue relating to the ADD routine, the CHANGE routine, and the overall system. There was no significant relationship between locus of control and ease of use for the systems-directed dialogue. There also was no significant relationship between locus of control and ease of use of the CHANGE routine or the overall system for the user-oriented dialogue.

However, the external locus of control group rated the ADD routine of the user-directed dialogue easier to use than did the internal group. The ADD routines in the two dialogues were identical except for two features. First, the ADD routine could be invoked at any time in the user-directed dialogue. Second, the user-directed dialogue allowed the user to back up one field at a time to correct mistakes made during data entry. It is suggested that these features are more important to internal individuals than to external individuals.

The responses to each ease of use question were also compared by preference group. In all six cases there were significant differences between the group responses. It may be concluded that in the selection of dialogues that differ only in the control dimension, ease of use will be a factor in individual preference.

Anxiety and User-Friendliness. Five of the six anxiety questions posed at the end of each dialogue proved to be related to preference. Based on this relationship it can be said that the level of anxiety was sufficient in the study to cause the students to consider this dimension of the software in their preference. However, there was no relationship between locus of control and the responses to any of the anxiety questions.

The user-friendly question was also asked at the end of each dialogue. The responses for the system-directed dialogue were not found to differ significantly between preference groups or between locus of control groups. However, the responses to the user-friendly question asked at the end of the user-directed dialogue did differ significantly for both the preference groups and the locus of control groups. The internal group rated the user-directed software as user-friendly more often than did the external group. Likewise, the group that preferred the user-directed dialogue rated that dialogue as user-friendly more often than did the group that preferred the system-directed dialogue.

It can be stated that software which gives individuals control over the man-machine interaction (dialogue) will be considered as user-friendly by internal locus of control individuals. Similarly external individuals will tend to rate the same software as not being as user-friendly.

Effect of Order and Instructor

The subjects were classified according to which dialogue they used first to determine if the order of the treatments had an effect upon their performance or preference. There was no relationship between the order in which the dialogues were used and the performance or preference of the group.

The subjects were also grouped according to course instructor. Performance did not vary significantly between the two groups, nor was there a relationship between preference and instructor. It can therefore be concluded that the instructors did not have a significant impact on the students' preference or performance.

Performance Theory

It is concluded that the performance theory was not supported because of the personality trait chosen to test the theory. Either the locus of control instrument used does not properly classify students for their locus of control when working with computers, or if the classification is correct, the lack of congruence between the individual and the software is not sufficient to test the theory. It is not possible, based on this study, to ascertain which is true. It is suggested that possibly both are true and that to properly test the theory other approaches should be examined. Such approaches should utilize stronger measures of individual differences, preferably in the cognitive domain.

Recommendations for Further Study

The theory that performance is affected by anxiety was clearly not supported by this study. As the personality trait, locus of control, did not produce sufficient anxiety to support the theory, other approaches are suggested.

First, anxiety in man-machine interactions requires definition. In this study anxiety was treated as an incongruence between personality traits and computer dialogues.

While this definition may be adequate for an exploratory study, further work should begin by refining the definition. It is also possible that anxiety manifests itself in forms that are measurable. The development of a working definition and the development of an instrument to measure anxiety in a man-machine interaction is needed.

Second, this study did not attempt to measure anxiety but only to induce anxiety in one situation and remove anxiety in the other. It is suggested that anxiety exists on a continuum and that performance at different levels of anxiety will be a curvilinear relationship. That is, at certain points anxiety might well enhance performance, at other points it may diminish performance.

Third, personality differences were not sufficient to produce anxiety at a level that affects performance, hence other facets of the man-machine interaction need to be manipulated to test the theory. The other facets to be studied might be in the form of motor skills, such as typing, or might be cognitive differences such as reading comprehension. The advantage of using such individual characteristics outside the affective domain is that there are readily available instruments for measuring these traits; instruments which have greater validity than is possible with those available for measuring affective traits.

Fourth, typing proficiency affected performance in this study but did not affect preference. What is not known is the impact of motor skills, such as typing, on comprehension in a man-computer interaction. Studies need to be conducted to determine if typing ability might influence the rate at which computer users learn the syntactic and semantic structures of man-computer dialogues.

Fifth, motivational factors should be established in studies on performance to insure that students are working to their capacity when completing the exercise.

Sixth, a longitudinal study would perhaps reveal more about the desire of individuals for control over man-machine dialogues. While it is known that novice users do not seek control, it is not known at what level of exposure to man-machine dialogues users begin to desire control.

Another experiment might offer several levels of control within a single software product. (A similar method is used in video games where the user can select their level of proficiency from novice to expert.) With such software the users' selection of control level could be recorded over a series of treatments to ascertain if locus of control was relevant to desire for control in computer software when a choice was available.

APPENDIX A

APPENDIX A

ROTTER LOCUS OF CONTROL SCALE

Student Number _______________

This is a questionnaire to find out the way in which certain important events in our society affect different people. Each item consists of a pair of alternatives lettered a or b. Please select the one statement of each pair (and only one) which you more strongly believe to be the case as far as you're concerned. Be sure to select the one you actually believe to be more true rather than the one you think you should choose or the one you would like to be true. This is a measure of personal belief: obviously there are no right or wrong answers.

Please answer these items carefully but do not spend too much time on any one item. Be sure to find an answer for every choice.

In some instances you may discover that you believe both statements or neither one. In such cases, be sure to select the one you most strongly believe to be the case as far as you're concerned. Also try to respond to each item independently when making your choice; do not be influenced by your previous choices.

Circle the appropriate answer for you.

1 a. Children get into trouble because their parents punish them too much.
  b. The trouble with most children nowadays is that their parents are too easy with them.

2 a. Many of the unhappy things in people's lives are partly due to bad luck.
  b. People's misfortunes result from the mistakes they make.

3 a. One of the major reasons why we have wars is because people don't take enough interest in politics.
  b. There will always be wars, no matter how hard people try to prevent them.

4 a. In the long run people get the respect they deserve in this world.
  b. Unfortunately, an individual's worth often passes unrecognized no matter how hard he tries.

5 a. The idea that teachers are unfair to students is non-sense.
  b. Most students don't realize the extent to which their grades are influenced by accidental happenings.

6 a. Without the right breaks one cannot be an effective leader.
  b. Capable people who fail to become leaders have not taken advantage of their opportunities.

7 a. No matter how hard you try some people just don't like you.
b. People who can't get others to like them don't understand how to get along with others.

8 a. Heredity plays the major role in determining one's personality.
b. It is one's experiences in life which determine what they're like.

9 a. I have often found that what is going to happen will happen.
b. Trusting to fate has never turned out as well for me as making a decision to take a definite course of action.

10 a. In the case of the well prepared student there is rarely if ever such a thing as an unfair test.
b. Many times exam questions tend to be so unrelated to course work that studying is really useless.

11 a. Becoming a success is a matter of hard work, luck has little or nothing to do with it.
b. Getting a good job depends mainly on being in the right place at the right time.

12 a. The average citizen can have an influence in government decisions.
b. This world is run by the few people in power, and there is not much the little guy can do about it.

13 a. When I make plans, I am almost certain that I can make them work.
b. It is not always wise to plan too far ahead because many things turn out to be a matter of good or bad fortune anyhow.

14 a. There are certain people who are just no good.
b. There is some good in everybody.

15 a. In my case getting what I want has little or nothing to do with luck.
b. Many times we might just as well decide what to do by flipping a coin.

16 a. Who gets to be the boss often depends on who was lucky enough to be in the right place first.
b. Getting people to do the right thing depends upon ability, luck has little or nothing to do with it.

17 a. As far as world affairs are concerned, most of us are the victims of forces we can neither understand, nor control.
b. By taking an active part in political and social affairs the people can control world events.

18 a. Most people don't realize the extent to which their lives are controlled by accidental happenings.
b. There realy is no such thing as "luck".

19 a. One should always be willing to admit mistakes.
b. It is usually best to cover up one's mistakes.

20 a. It is hard to know whether or not a person really likes you.
b. How many friends you have depends upon how nice a person you are.

21 a. In the long run the bad things that happen to us are balanced by the good ones.
b. Most misfortunes are the result of lack of ability, ignorance, laziness, or all three.

22 a. With enough effort we can wipe out political corruption.
b. It is difficult for people to have much control over the things politicians do in office.

23 a. Sometimes I can't understand how teachers arrive at the grades they give.
b. There is a direct connection between how hard I study and the grades I get.

24 a. A good leader expects people to decide for themselves what they should do.
b. A good leader makes it clear to everybody what their jobs are.

25 a. Many times I feel that I have little influence over the things that happen to me.
b. It is impossible for me to believe that chance or luck plays an important role in my life.

26 a. People are lonely because they don't try to be friendly.
b. There's not much use in trying too hard to please people, if they like you, they like you.

27 a. There is too much emphasis on athletics in high school.
b. Team sports are an excellent way to build character.

28 a. What happens to me is my own doing.
b. Sometimes I feel that I don't have enough control over the direction my life is taking.

29 a. Most of the time I can't understand why politicians behave the way they do.
b. In the long run the people are responsible for bad government on a national as well as on a local level.

APPENDIX B

MAN-MACHINE DIALOGUE ASSESSMENT

The purpose of this assignment is to give you exposure to two different man-machine dialogues that are frequently seen in software today.

As students of business, you will become the future managers of various business functions. In that capacity, you and your staff will interact with computers more than any previous generation has. It is therefore important that you are aware of alternatives available in man-computer dialogues.

This assignment consists of two exercises which will be handed out separately. Both exercises involve the same task, the data entry function in a payroll application on the IBM PC microcomputer. You will be provided with a page of data representing a typical Departmental Time Report and with Operating Procedures for each exercise. (The individual exercises are labelled DIALOGUE I and DIALOGUE II.) You will start with whichever dialogue is distributed to you first. Only after the first exercise is handed in should you do the other exercise.

On each Departmental Time Report a double line marks where a previous operator stopped. That is, the first half of the data has been entered already. The circled items above the double line indicate that the particular item was entered incorrectly and must be changed. The data below the double line must be entered by you.

The exercises differ only in the data used and the manner in which the data is manipulated. Each exercise permits you to add, delete, change, or list the payroll data. The specific procedures are outlined on separate pages labled Operating Procedures. Each exercise is complete when you have totals that match the totals on the bottom of the Departmental Time Reports.

As you complete each exercise evaluate the experience in your own words. The main difference in the two exercises is the dialogue used. Focus your comments on the dialogue. There are no right or wrong answers, only valid opinions.

When you have finished the first exercise hand in your printout and pick up the second assignment.

When you have finished both exercises complete the final part the evaluation indicating your personal preference. At this time hand in your diskette, the evaluation form, and your second printout.

DIALOGUE EVALUATION FORM

Name ______________________

Student # _________________

Section # _________________

Please provide your own observations on using each dialogue. List any comments or criticisms as well as suggestions for improvements.

DIALOGUE I ____

____

____

____

____

____

____

DIALOGUE II ____

____

____

____

____

____

____

****

Complete this portion <u>after</u> you have had an opportunity to use both dialogues.

Which dialogue would <u>you</u> <u>personally</u> prefer? Please circle either a or b.

a) DIALOGUE I
b) DIALOGUE II

Why? ____

____

____

DISKETTE NUMBERS 1 THROUGH 8

SSN ____________

Name ____________________

Section ________

Operating Procedures - DIALOGUE I

The programs you need for this task are available from the Grawn Lab personnel. Present this sheet when asking for the diskette.

If the IBM PC is not on insert the program diskette in drive A and your diskette in drive B. Turn on the computer. (If the computer is already on, insert the diskettes as above, and reboot the system by pressing CTRL, ALT, and DEL.)

The program will ask you for your social security number, name, and <u>section number</u>. The following steps outline what you will be doing with this program.

1. You will be asked to add any new records not already entered. After each record entered you will be asked 'IS IT CORRECT (Y/N)'. If you reply with a Y the record will be written to the diskette and you will be prompted for the next record. If you reply with an N the record will not be written to the diskette, and you will be prompted to enter the data again. When you have added all the records you may end the ADD routine by typing the word STOP when prompted for employee number. (Complete all entries by pressing the ENTER key.)

2. The system will next sort the file by employee number. This step is necessary when records have not been entered in employee number order.

3. The program will next process any changes you need to make. Enter the employee number of the record you wish to change. The record will be displayed on the screen with a prompt message 'CHANGE (Y/N)' beside each field. Enter a Y if you wish to change the corresponding field.

4. The program will next list the file and provide you with totals of each numeric field. If there are any errors the program will tell you which totals are erroneous and will begin again asking for any additions to the file (in case you forgot to enter a record). If you have no additions, respond with 'STOP' when asked for employee number.

5. The process will continue as outlined in steps 1 through 4 until the totals are correct.

6. Once you have achieved the correct totals you will be presented with questions to answer. Please be as accurate as possible when answering the questions.

7. Return the program diskette to the lab personnel. If you this is your first exercise, hand in your printout and pick up the last exercise. If you have completed both exercises, hand in your printout, your evaluation of the two dialogues, and your diskette.

Acme Manufacturing Co.
Departmental Time Report

DATA FOR DIALOGUE I

****

EMP#	NAME	DEP	SFT	REG	OT	RATE	SICK	VAC
1010	ABLE, JOHN	3	1	20	6	9.65	12	8
1015	BACKUS, DAVID	5	3	40	2	8.95	0	0
1020	BINTZ, JAMES	2	2	36	0	9.45	0	4
1025	CHILDS, DANIEL	8	3	40	8	9.65	0	0
1030	CZARKA, STANLEY	1	1	32	0	8.95	0	8
1035	DESIDERIO, BRET	4	3	16	5	7.65	8	16
1040	DYMOWSKI, JAN	3	1	40	7	9.35	0	0
1045	ELZINGA, ABBA	2	3	38	0	8.75	2	0
1050	FAHNDRICH, IVAN	6	2	40	9	9.80	0	0
1055	FRARY, ROBERT	4	2	36	8	8.78	0	4
1060	GATTUSO, JANICE	5	1	40	0	7.95	0	0
1065	GAJSIEWICZ, JOE	2	2	38	2	7.85	2	0
1070	GRODESKI, ORVAL	3	3	40	0	5.45	0	0
1075	HAVIARAS, MARK	9	3	25	0	8.65	15	0
1080	HILLMAN, GLENN	3	2	0	0	7.95	0	40
1085	HUNT, WILLIAM	4	1	40	6	8.45	0	0
1090	IAQUINTA, BETH	3	1	34	4	4.95	4	2
1095	JARCZYNSKI, AL	1	1	11	1	9.85	1	28
1100	KENNEDY, DONALD	4	3	32	0	7.95	8	0
1105	KLINE, ROBERT	4	2	40	8	6.45	0	0
1110	LYTTLE, JAMES	6	1	40	0	9.65	0	0
1115	MAUCH, SUZANNE	3	1	32	0	7.95	0	8
1120	MUSYNSKI, MARK	7	2	40	12	8.45	0	0
1125	SEELEY, SHARI	5	2	32	4	6.95	8	0
1130	WEAVER, TIMOTHY	4	3	40	0	3.35	0	0
TOTALS -				822	82	202.83	60	118

DISKETTE NUMBERS 9 THROUGH 16

SSN ________________

Name ________________________________

Section ________

Operating Procedures - DIALOGUE II

The programs you will need for this task are available from the Grawn Lab personnel. Present this sheet when asking for the diskette.

Insert the program diskette in drive A and your diskette in drive B. If the IBM is not on, turn it on now. If it is already on, reboot the system by pressing CTRL, ALT, and DEL.

The program will prompt you for your social security number, name, and your section number.

With this series of programs you will be asked for 'commands'. The valid commands are ADD, CHANGE, DELETE, SORT, LIST, HELP, and END. Each command has options as outlined below. The commands may be used in any order.

ADD Adds a new record to the file.

options: Shift ^ - Pressing the shift and the ^ symbol found over the '6' key will allow you to 'back-up' when entering new data. You may use this key sequence repeatedly to back up more than one field. The ADD routine is terminated by typing the word STOP instead of an employee number.

DELETE Delete a record from the file.

options:
- A - Delete all records.
- O - Delete one record. For example, the first, or the fifth.
- R - Delete a range of records, such as the third through the sixth.
- K - Delete a specific record by providing the record key, that is, the employee number.

CHANGE   Change fields within record(s).

options: A - The program will present all records
O - The program will present a specific record within the file. A 3 indicates the third record, etc.
R - A range of records will be presented for changing. For example, you could specify changing the second through the eight records in the file.
K - A specific record is presented for changing. The record is identified by its key, i.e., employee number.

As each record is presented for your changes the program will ask for the field to be changed. After each change is made you may either change another field or end the changes for the record by typing the word STOP. The valid field names are presented on the screen with the current data.

SORT   Sort the file to employee number (EMP#).

options: A - Sort to ascending order (lowest to highest)
D - Sort to descending order (highest to lowest)

LIST   List record(s) on the screen or the printer.

options: Screen or Printer
B - Block mode; this mode is the same format as you see when adding or changing.
F - Form listing; in this mode you specify fields you would like on the output. You need not include the fields DEP and SHIFT if you want the data to fit on one line. When you have entered all the fields you need, press the ENTER key.
TOTAL NUMERIC FIELDS (Y/N) - Answer with a 'Y' to have totals printed under each numeric field.
TITLE - Type an appropriate title for the for the listing.

HELP   Provides a list of the valid commands.

END   You must type END when you arrive at the correct totals, otherwise the remainder of this exercise will not be completed and it will be necessary to redo the entire exercise.

Acme Manufacturing Co.
Departmental Time Report

DATA FOR DIALOGUE II

****

EMP#	NAME	DEP	SFT	REG	OT	RATE	SICK	VAC
1010	ABLE, JOHN	3	1	35	9	8.67	15	8
1015	BACKUS, DAVID	5	3	30	2	9.85	0	0
1020	BINTZ, JAMES	2	2	35	1	9.54	0	6
1025	CHILDS, DANIEL	8	3	40	7	9.56	0	8
1030	CZARKA, STANLEY	1	1	33	8	9.58	0	0
1035	DESIDERIO, BRET	4	3	26	4	7.56	16	8
1040	DYMOWSKI, JAN	3	1	37	4	5.39	0	0
1045	ELZINGA, ABBA	2	3	39	0	8.86	1	0
1050	FAHNDRICH, IVAN	6	2	4	8	9.89	0	0
1055	FRARY, ROBERT	4	2	37	6	9.89	4	0
1060	GATTUSO, JANICE	5	1	0	4	7.67	7	6
1065	GAJSIEWICZ, JOE	2	2	34	3	7.34	3	4
1070	GRODESKI, ORVAL	3	3	30	0	4.54	0	0
1075	HAVIARAS, MARK	9	3	52	15	8.56	0	0
1080	HILLMAN, GLENN	3	2	25	0	5.68	15	0
1085	HUNT, WILLIAM	4	1	0	0	7.95	0	40
1090	IAQUINTA, BETH	3	1	33	4	3.45	4	5
1095	JARCZYNSKI, AL	1	1	22	2	5.89	1	21
1100	KENNEDY, DONALD	4	3	23	0	8.05	8	0
1105	KLINE, ROBERT	4	2	39	7	5.35	0	0
1110	LYTTLE, JAMES	6	1	40	0	9.65	0	0
1115	MAUCH, SUZANNE	3	1	23	0	5.97	8	0
1120	MUSYNSKI, MARK	7	2	4	2	7.34	0	0
1125	SEELEY, SHARI	5	2	23	3	5.84	7	0
1130	WEAVER, TIMOTHY	4	3	20	10	4.46	0	0
TOTALS -				684	99	186.53	89	106

APPENDIX C

APPENDIX C

OVERHEADS USED TO ILLUSTRATE ALL SCREEN DISPLAYS

```
A>DATE
Current Date is Tue 1-01-1980
Enter new date: 2-6-83
```

Screen 1 - Dialogue I and II

```
PRESS 'CAPS LOCK' KEY - THEN ENTER
PRESS 'NUM LOCK' KEY -THEN ENTER
```

Screen 2 - Dialogue I and II

```
SOCIAL SECURITY NUMBER (NO HYPHENS): 364500717
LAST FIRST NAME (NO COMMAS):? DOE JOHN
SECTION NUMBER :? 5558

IS THE ABOVE CORRECT (Y/N)?  Y
```

Screen 3 - Dialogue I and II

ACME MANUFACTURING CO.
WEEKLY PAYROLL - ADDITIONS

If you make an error while entering data, note the error on paper and continue entering the record.

```
EMPLOYEE NUMBER      :  :----:

NAME (DOE, JOHN)     :

DEPENDENTS (0-9)     :

SHIFT (1,2,OR 3)     :

REGULAR HOURS        :

OVERTIME HOURS       :

PAY RATE             :

SICK HOURS           :

VACATION HOURS       :
```

Screen 4 - Dialogue I

THE SYSTEM IS NOW SORTING RECORDS TO EMPLOYEE NUMBER

Screen 5 - Dialogue I

ACME MANUFACTURING CO.
WEEKLY PAYROLL - DELECTIONS
Check your listing for nay invalid employee numbers.

Enter the invalid number to delete the record, or enter 'STOP'.

ENTER EMPLOYEE NUMBER: :----:

Screen 6 - Dialogue I

```
                    ACME MANUFACTURING CO.
                WEEKLY PAYROLL - CORRECTIONS

Enter employee number of record to change, or the word STOP

          Employee #   :1015:   BACKUS, DAVID

************************************************************

    Name            : ABLE JOHN          Change (Y/N) :N:
    Dependents      :  3                 Change (Y/N) :N:
    Shift           :  1                 Change (Y/N) :N:
    Regular hours   :  2.00              Change (Y/N) :N:
    Overtime hours:    6.00              Change (Y/N) :N:
    Pay rate        :  5.69              Change (Y/N) :Y:
    Sick hours      : 12.00
    Vacation hours:    7.00

                    Current Data:    12.00
                    Change to    : ? ______
```

Screen 7 - Dialogue I

ACME MANUFACTURING CO.
WEEKLY PAYROLL - VERIFICATION

THE TOTAL REGULAR PAY IS NOT CORRECT.
THE TOTAL OVERTIME PAY IS NOT CORRECT.
THE PAY RATE HASH TOTAL IS INCORRECT.
THE TOTAL SICK HOURS IS NOT CORRECT.
THE TOTAL VACATION HOUR IS INVALID.

When the printer has finished you may add, change, or delete as necessary to correct the errors.

Screen 8 Dialogue I

```
ACME MANUFACTURING
Payroll Data Entry
Dialogue II
-----------------------------------------------------------
The valid commands are :  ADD
                          CHANGE
                          DELETE
                          LIST
                          SORT
                          END

Note that the CHANGE, DELETE, and LIST require
that you specify whether all records, a specific
record, or a range of records are to be used.

FILE - EXPER2        KEY FIELD  - EMP#    # OF RECORDS - 10
COMMAND OR  END - :----------:
```

Command Screen - Dialogue II

```
FILE - EXPER2          KEY FIELD - EMP#     # OF REOCRDS - 10
COMMAND   -:ADD-------:
( 11 )
EMP#         < 4 >:1060:
NAME         < 15 >GATTUSO, JANICE:
DEP          < 1 >:5:
SHIFT        < 1 >:2:
REGHRS      :? 0
OTHRS       :? 4
RATE        :? 7.67
SICKHRS     :? 7
VACHRS      :? 6
Are data entries correct  (Y or N)?  Y
```

Add Screen - Dialogue II

```
FILE - EXPER2         KEY FIELD - EMP#     # OF RECORDS - 11
COMMAND   -:DELETE----:
Enter mode: A)ll,  O)ne,  R)ange,  K)ey - K

***  KEY FILE MUST BE SORTED  ***
Enter key value    1060:
```

Delete Screen - Dialogue II

```
FILE - EXPER2        KEY FIELD  - EMP#    # OF RECORDS -  11
COMMAND  -:CHANGE----:
Enter mode: A)ll,  O)ne,  R)ange,  K)ey - K

***   KEY FILE MUST BE SORTED   ***
Enter key value   1060:

( 11 )
EMP#          :1060
NAME          :GATTUSO, JANICE
DEP           :5
SHIFT         :2
REGHRS        : 40
OTHRS         : 4
RATE          : 7.67
SICKHRS       : 7
VACHRS        : 6
Enter name of field - REGHRS
REGHRS      :?
```

Change Screen - Dialogue II

```
FILE - EXPER2           KEY FIELD   - EMP#    # OF RECORDS - 11
COMMAND  -:LIST------:
Listing on printer  (Y or N)? Y
Enter mode: A)ll,  O)ne,  R)ange,  K)ey - A
B)lock or F)orm listing? F


EMP#          NAME       DEP          SHIFT      REGHRS
OTHRS         RATE       SICKHRS      VACHRS

Select fields you wish for form
Enter name of field - EMP#
Enter name of field - NAME
Enter name of field - REGHRS
Enter name of field - OTHRS
Enter name of field - RATE
Enter name of field - SICKHRS
Enter name of field - VACHRS
Enter name of field - STOP
Total numeric fields  (Y or N)? Y
Forms title - LIST OF DATA USED IN DIALOGUE II
```

List Screen - Dialogue II

APPENDIX D

APPENDIX D

QUESTIONS PRESENTED AT THE END OF EACH DIALOGUE SESSION

1. How would you rate the operations allowing additions to the file?
    - a) easy to use
    - b) moderately easy to use
    - c) neutral
    - d) difficult to use
    - e) very difficult to use

2. How would you rate the operation allowing changes to the file?
    - a) easy to use
    - b) moderately easy to use
    - c) neutral
    - d) difficult to use
    - e) very difficult to use

3. How would you rate this program overall in terms of ease of use?
    - a) easy to use
    - b) moderately easy to use
    - c) neutral
    - d) difficult to use
    - e) very difficult to use

4. How did you feel when using the add routine?
    - a) very comfortable
    - b) comfortable
    - c) neutral
    - d) frustrated
    - e) very frustrated

5. How did you feel when using the change routine?
    - a) very comfortable
    - b) comfortable
    - c) neutral
    - d) frustrated
    - e) very frustrated

6. How did you feel about the overall system?
    - a) very comfortable
    - b) comfortable
    - c) neutral
    - d) frustrated
    - e) very frustrated

7. How fast do you type?
   a) hunt and peck
   b) know where to place hands but have to look at keyboard frequently
   c) 10 to 30 words per minute
   d) 30 to 50 words per minute
   e) above 50 words per minute

8. How would you rate this system?
   a) user-friendly
   b) almost user-friendly
   c) neutral
   d) not very user-friendly
   e) not at all user-friendly

SELECTED BIBLIOGRAPHY

SELECTED BIBLIOGRAPHY

Boillot, M. Computer communication modes and their effect on student attitudes towards programming. Nova University thesis, April 1974. (ERIC Reproduction Service No. ED 098 957.

Broadbent, D. E. & Broadbent, M. H. P. "The allocation of descriptor terms by individuals in a simulated retrieval system." Ergonomics, 1978, 21, 343-354.

Brosey, M. & Schneiderman, B. "Two experimental comparisons of relational and hierarchical database models." International Journal of Man-Machine Studies, 1978, 10, 625-637.

Card, S. & Moran, T. "The keystroke level model of user performance time with interactive systems." Communications of the ACM, July 1980, 23, 7, 396-410.

Codd, E. F. "A data base sublanguage founded on the relational calculus." Proceedings, ACM SIGFIDET, Workshop on Data Description, Access, and Control, 1971, pp. 35-68.

Conrades, G. H. "Information Processing - Dollars to Cents". Information Processing January 1982, 1, 1, 16a-16d.

Crowne, D. P. & Marlowe, D. The Approval Motive. Wiley, New York, 1964.

Control Data Corporation XEDIT VERSION 3 REFERENCE MANUAL, 60455730, Publications and Graphics Division, ARH219, St. Paul, 1981.

Date, C. J. An Introduction to Database Systems (2nd ed.). Addison-Wesley, 2nd edition, Reading, Massachusetts, 1977.

Dehning, W., Essig, H., & Maass, S., "The Adaptation of Virtual Man-Computer Interfaces to User Requirements in Dialogs." Lecture Notes in Computer Science, 1981, 110, Springer-Verlag, Berlin.

Dijkstra, E. W. "GO TO statement considered harmful." Communications of the ACM, 1968, 11, 147-148.

Dock, V. & Essick, E. Principles of Business Data Processing with MIS (4th ed.). Science Research Associates, Chicago, Ill., 1982, pp. 205, 350, 473.

Gold, M. M. "Time-Sharing and batch processing: An experimental comparison of their values in a problem-solving situation." Communications of the ACM, May 1969, 12, 5, 249-259.

Goodman, T. & Spence, R. "The effect of system response time on interactive computer aided problem solving." ACM SIGGRAPH '78, Conference Proceedings, 1978, pp. 100-104.

Gore, M. & Stubbe, J. Elements of Systems Analysis (2nd ed.). Wm. C. Brown, Dubuque, Iowa, 1979, p. 255.

Green, T. R. G. "Conditional program statements and their comprehensibility to professional programmers." Journal of Occupational Psychology, 1977, 50, 93-109.

Greenblatt, D. & Waxman, J. "A study of three database query languages." In B. Shneiderman (Ed.), Databases: Improving Usability and Responsiveness, Academic Press, New York, 1978, pp. 77-97.

Grossberg, M., Wiesen, R. A., & Yntema, D. B. "An experiment on problem solving with delayed computer responses." IEEE Transactions on Systems, Man, and Cybernetics, SMC-6, March 1976, 3, 219-222.

Halstead, M. "Elements of Software Science." Operating and Programming Systems Series, Elsevier Computer Science Library, New York, 1977.

Hansen, J. V. "Man-machine communications: An experimental analysis of heuristic problem-solving under on-line and batch-processing conditions." IEEE Transactions on Systems, Man and Cybernetics, November 1976, 6, 11, 746-752.

Kelly, T. L., "The Selection of Upper and Lower Groups for the Validation of Test Items." J. Educ Pschol., January 1939, pp 17-24.

Lee, J. & Schneiderman, B. "Personality and programming: Time-sharing vs. batch processing." Proceedings of the ACM National Conference, 1978, pp. 561-569.

Lochovsky, F. H. & Tsichritzis, D. C. "User performance considerations in DBMS selection." Proceedings of the ACM SIGMOD, 1977, pp. 128-134.

Lochovsky, F. H. Data base management system user performance. Ph.D. dissertation, Univ. Toronto, Canada, 1978.

Manuel, T. "Japan Maps Computer Domination." BYTE, May 1982, pp. 140-144.

Martin, J. Applications without programmers. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1982.

Miller, L.A., & Thomas, J.C., "Behavioral Issues in the Use of Interactive Systems." Proceedings of the 6th Informatik Symposium of IBM Germany on Interactive Systems, Lecture Notes in Computer Science, 1976, 49, Springer-Verlag, Berlin, pp 193-215.

Miller, R. B. "Response time in man-computer conversational transactions." Proceedings Spring Joint Computer Conference 1968, 33, AFIPS Press, Montvale, New Jersey, 267-277.

Rathus, S. "A 30-item Schedule for Assessing Assertive Behavior." Behavior Therapy, 1973, 4, 398-406.

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

Reisner, P. "Human Factors Studies of Database Query Languages: A Survey and Assessment." Computing Surveys, March 1981, 13, 1, 13-31.

Reisner, P. Boyce, R. F., & Chamberlin, D. D. "Human factors evaluation of two data base query languages: SQUARE and SEQUEL." Proceedings of the National Computer Conference, AFIPS Press, Montvale, New Jersey, 1975.

Ritchie, M. "Human Factors in the Long Run." Human Factors April 1970, 12, 153-157.

Rotter, J. B. "Generalized Expectancies for Internal vs. External Control of Reinforcement." Psychological Monographs, 1966, 80, 1, 1-28.

Sackman, H. "Experimental analysis of man-computer problem-solving." Human Factors, 1970, 12, 187-201. (a)

Sackman, H. Man-Computer Problem Solving. Auerbach Publishers Inc., Princeton, New Jersey, 1970. (b)

Schatzoff, M., Tsao, R., & Wiig, R. "An experimental comparison of time-sharing and batch processing." Communications of the ACM, May 1967, 10, 5, 261-265.

Sheil, B. A. "The Psychological Study of Programming." Computer Surveys, March 1981, 13, 1, 101-120.

Sheppard, S. B., Curtis, B., Milliman, P., Borst, M. A., & Love, T. "First year results from a research program on human factors in software engineering." Proceedings of the National Computer Conference, 1979, 48, AFIPS Press, Montvale, New Jersey, 73-79.

Shneiderman, B. "Exploratory Experiments in Programmer Behavior." International Journal of Computer and Information Sciences, June 1976, 5, 2, 123-143.

Shneiderman, B. "Human Factors Experiments for Developing Quality Software." State of the Art Report on Software Reliability, 1977, (Berkshire, England: Infotech).

Shneiderman, B. "Human Factors Experiments in Designing Interactive Systems." COMPUTER, December 1979, pp. 21-38.

Shneiderman, B. Software Psychology - Human Factors in Computer and Information Systems. Winthrop Publishers, Inc., Cambridge, Massachusetts, 1980.

Sime, M. E., Green, T. R. G., & Guest, D. J. "Psychological evaluation of two conditional constructions used in computer languages." International Journal of Man-Machine Studies, 1973, 5, 1, 105-113, 123-143.

Sime, M. E., Green, T. R. G., & Guest, D. J. "Scope marking in computer conditionals - a psychological evaluation." International Journal of Man-Machine Studies, 1977, 9, 107-118.

Sondheimer, N. "On the fate of software enhancements." Proceedings of the National Computer Conference, 1979, 49, AFIPS Press, Montvale, New Jersey.

Stern, R. A. & Stern N. An Introduction to Computers and Information Processing, 1982, John Wiley & Sons, New York.

Walther, C. E. & O'Neil Jr., H. F. "On-line user-computer interface - the effects of interface flexibility, terminal type, and experience on performance." Proceedings of the National Computer Conference, 1974, 43, AFIPS Press, Montvale, New Jersey.

Weinberg, G. M. The Psychology of Computer Programming, Van Nostrand Reinhold, New York, 1971.

Weissman, L. "Psychological complexity of computer programs: An experimental methodology." ACM SIGPLAN Notices, 1974, 9.

Welty, C. "A Comparison of a Procedural and a Nonprocedural Query Language: Syntactic Metrics." Ph.D. Dissertation, University of Massachusetts, 1979, University Microfilms International, Ann Arbor, Mich.

Winer, B. J. Statistical Principles in Experimental Design, McGraw-Hill, New York, 1971.

Winship, B. J. & Kelly, J. D. "A verbal response model of assertiveness." Journal of Counseling Psychology, 1976, 23, 3, 215-220.