Summer 1991

Communicating With Graphic User Interfaces: A Comparison of Menu Selection and Menu Bypass Techniques

Monty Lee Hammontree
Old Dominion University

COMMUNICATING WITH GRAPHIC USER INTERFACES:

A COMPARISON OF MENU SELECTION AND

MENU BYPASS TECHNIQUES

by

Monty Lee Hammontree
B. A. May 1984, David Lipscomb College
M. A. August 1986, Middle Tennessee State University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

INDUSTRIAL/ORGANIZATIONAL PSYCHOLOGY

OLD DOMINION UNIVERSITY
August, 1991

Approved by:

Glynn D. Coates (Director)

ABSTRACT

COMMUNICATING WITH GRAPHIC USER INTERFACES:
A COMPARISON OF MENU SELECTION AND
MENU BYPASS TECHNIQUES

Monty Lee Hammontree
Old Dominion University, 1991
Director: Dr. Glynn D. Coates

The present study was conducted in two phases to determine design tradeoffs relating to command bar menu and bypass code-based techniques for interacting with computers. Forty eight subjects participated. In the first phase of the experiment, mouse-, chorded key-, and function key-based menu selection techniques were compared. It was found that menus were accessed much faster with spatially mapped function keys as compared to chorded key sequences or mouse inputs, and that relative to mouse inputs compatible letter keys lead to faster command selection times. Further, the function key-based technique yielded the fastest combined access and selection times, the fastest block completion times, and the fewest errors. In the second phase of the experiment, four experimental conditions were produced by crossing two menu input devices (i.e., mouse and keyboard) with two bypass coding structures (i.e., function key-based codes and chorded key-based codes). It was found that the groups which used function key-based codes entered the menu designating portion of the bypass codes faster than those that used chorded key-based codes. The coding structure based on spatially mapped function keys also yielded faster

task completion times. Furthermore, there were fewer command substitution errors with this coding structure. Comparisons between the groups with no prior exposure to the code sequences (i.e., the groups that used the mouse to make menu selections during the first phase) revealed that the function key-based technique also led to fewer command ommissions and fewer extraneous command selections. Finally, subjective data showed menus were felt to be easier to learn, less demanding in terms of mental resources, and less anxiety provoking than bypass codes. In contrast, bypass codes were felt to be more natural, more convenient to use, and faster in terms of task times and better in terms of task performance. The findings of this study clearly indicate that both menu- and bypass code-based styles of control should be provided to promote user acceptance. Furthermore, the performance advantages observed for the function key-based technique point to it as the menu selection and bypass technique of choice.

ACKNOWLEDGEMENTS

I would like to thank Dr. Glynn Coates who always seemed to know when to help, when to encourage, and when to foster independence. His impact on the quality of this work and the professional development of its author were immeasurable. I would also like to thank Dr. Raymond Kirby and Dr. Fred Freeman for their support and guidance.

I am especially indebted to Dr. Jeffrey Hendrickson of Texas Instruments who served as my mentor and friend throughout the course of this project. Under periods of duress I thoroughly tested his technical expertise and our friendship - both survived unscathed.

In addition, I would like to thank Dr. William Muto, of the User System Engineering Lab at Texas Instruments, who provided the lab facilities, and support that made this research possible. I would also like to thank Dr. Gerhard Deffner of Texas Instruments for his support and encouragement during the early stages of this project. Without the support of the staff and management at Texas Instruments this project would not have been possible.

Additionally, I owe my sincere thanks to Mr. Billy Hensley for his dedication and hard work in writing the data collection software that was vital to the completion of this project.

Finally, I owe my deepest appreciation to my wife, Amy, who never understood my plight yet continued to stand by me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

There are two basic means by which a human can specify to a computer the actions he or she wishes to have performed: naming and pointing. Specifying actions by naming entails the entry of codes or terms that correspond (i.e., map) to the alternative actions. Specifying actions by pointing entails physically manipulating a pointing device (e.g., a mouse) in a fashion that indicates to the computer which of a series of visually represented actions are to be performed. These two modes of communication have evolved into a variety of human-computer control/dialogue styles.

A number of authors have examined these evolving control styles and have, in turn, identified the more popular or prototypical approaches (Hammond & Barnard, 1984; Margono & Shneiderman, 1985; Ramsey & Atwood, 1979; Shneiderman 1987; Shneiderman 1988; Smith & Mosier, 1986).

Margono and Shneiderman (1985) suggested there are three basic control styles which are used in human computer interfaces. The three control styles to which they refer are menu-based, command language-based, and direct manipulation-based. These three styles are distinguishable by the amount and type of guidance that is presented at the moment that a command selection is made and by the manner in which this selection is made.

If a list of available options is displayed as the user chooses a course of

action, the style of control in use is considered to be menu-based. Within menu-based systems, commands can be chosen from a menu either by referring to the commands by name or by an identifier (e.g., by typing), or by pointing at the desired option.

In cases where a user selects a course of action without the benefit of on-screen aids or guidance, the style of control in use is referred to as command language-based. When using this style of control the user selects the desired course of action by recalling and entering the required notation (e.g., by typing the name or coded referent for the command).

Finally, if the user specifies a given course of action by directly manipulating a graphic representation of the task domain, the style of control is referred to as direct manipulation-based. By definition, this style of control is achieved by manipulating the graphic representation directly with a pointing device rather than by naming (e.g., typing instructions).

Shneiderman (1988) suggested that user skill and experience have a profound influence on the trade-offs associated with the use of a given control style. He suggested that menu-based and direct manipulation-based styles of control are particularly well suited to the needs of the novice user in that they provide an environment in which the user can interact with the computer simply by locating and selecting familiar words, codes, or visual representations displayed on the screen. However, he suggested that the opposite is true for experts. Shneiderman wrote:

> Knowledgeable frequent users do not want to be distracted by having to locate an item in a list, nor do they want to have to view and move a cursor over a form. They can manipulate the possibilities in their mind and want concise notations for issuing commands with modest feedback. (p. 703)

In the face of the paradox created by the relationship between user experience and the suitability of the various control styles, Shneiderman concluded that researchers and system designers should actively seek useful and appropriate ways to blend together the characteristics of the various styles.

With Shneiderman's directive in mind, this chapter addresses: 1) current lore and conventional wisdom regarding the use of menus, direct manipulation, and command languages; 2) the benefits to be derived from blending these control styles; 3) empirical research relevant to the design and evaluation of mixed command selection techniques; 4) the manner in which today's graphical user interfaces handle command selection/execution; 5) the experimental rational underlying the present study; and finally the 6) hypotheses tested by this study.

Control Style Strengths and Weaknesses

Norman (1984) proposed that there are four stages of user activity that take place as a user interacts with a computer. These four stages are: 1) forming the intention (i.e., mentally characterizing the desired goal); 2) selecting the action (i.e., translating the intention into one of the actions possible at the moment); 3) executing the action (i.e., entering the selections into the system); and 4) evaluating the outcome.

Literature relating to the strengths and weaknesses of various control styles suggests that the different styles have different trade-off values for each of the stages of user activity and that the virtues of one style are quite often pitted against the deficits of another. This same body of literature also suggests that trade-offs for the various stages of user activity interact with the level of user experience.

Menu-based control – strengths and weaknesses. Norman (1984) pointed to the fact that menus can serve as a source of information for the intention and selection stages of an interaction and that menus can also provide information, or even the mechanism, for the execution stage. He further notes that:

> ...fans of menus usually are those who weight highly the information provided for intention and selection. Foes of menus are those who do not need assistance in these stages and who object to the loss of time and workspace during execution and evaluation. The differences come from differing needs at the different stages. (p. 372)

An examination of available literature suggests that menus have been touted as a means to: reduce memory load; structure or guide the decision making processes of novices; aid learning and cut training costs; create an easy to use interface for novice, casual, or intermittent users; reduce keying errors, syntax errors, and erroneous command entry; support error handling; reduce keystrokes; and minimize the need for keyboard skills (see Table 1).

Further examination of the literature reveals that menus have been criticized, however, for: being a slow mode of control when the user has clearly formed intentions and the knowledge necessary to carry them out; creating the perception of being slow; taking up valuable screen space; being cumbersome and inflexible; being annoying/frustrating for experienced users; causing the user to feel that they are not in control; and requiring awkward, time-consuming hand movements to and from the keyboard to auxiliary pointing devices (e.g., a mouse) (see Table 2).

Clearly, many of the leaders in the field of human-computer

TABLE 1

Advantages of Menu-Based Control by Supporting Authors

ADVANTAGES OF MENU-BASED CONTROL:	SUPPORTING AUTHORS: 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Is an easy to use mode of control for novices, casual, and/or intermittent users.											✓		✓	✓			✓	✓	✓			
Structures/guides the decision making processes of novices.		✓		✓				✓	✓			✓		✓	✓		✓	✓				
Aids learning – is easy to learn.	✓						✓						✓								✓	✓
Reduces training costs.										✓					✓	✓	✓				✓	
Reduces memory load – requires recognition rather than recall.		✓		✓	✓	✓				✓	✓			✓		✓		✓	✓	✓		
Leads to fewer keying errors, syntax errors, and/or erroneous command entries.			✓	✓										✓	✓	✓					✓	
Supports error handling.															✓							
Reduces keystrokes.															✓	✓						
Minimizes the need for keyboard skills.															✓							
Novices prefer menu-based control over command language-based control.																					✓	

1) Antin (1988); 2) Badre (1984); 3) Bertino (1985); 4) Bosser (1987); 5) Card (1984); 6) Davies, Lambert, & Findlay (1989); 7) Hall (1982); 8) Heffler (1981); 9) Larson (1982); 10) Laverson, Norman, & Shneiderman (1987); 11) Liebelt, Macdonald, Stone, & Farell (1982); 12) Norman (1984); 13) Norman (1983); 14) Paap & Roske-Hofstrand (1988); 15) Shneiderman (1988); 16) Shneiderman (1987); 17) Shneiderman (1986); 18) Streitz (1987); 19) Streitz, Leiser, & Wolters (1989); 20) Streitz, Spijkers, & van Duren (1987); 21) Taylor (1986); 22) Walker & Olson (1988)

TABLE 2

Disadvantages of Menu-Based Control by Supporting Authors

DISADVANTAGES OF MENU-BASED CONTROL:	SUPPORTING AUTHORS:															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Navigation and visual search requirements slow users who have clearly formed intentions and the knowledge necessary to carry them out.		✓	✓		✓		✓	✓	✓	✓	✓	✓		✓	✓	
Experienced users perceive it to be a slow means of control.	✓											✓				✓
Takes up valuable screen space.				✓							✓	✓	✓	✓		
Is cumbersome and inflexible.	✓	✓								✓			✓	✓		
Creates feelings of annoyance and frustration in experienced users.						✓	✓		✓			✓		✓		
Causes the user to feel they are not in control.								✓								
May require awkward time consuming hand movements to and from the keyboard to an auxiliary pointing device (e.g., a mouse).				✓												

1) Antin (1988); 2) Badre (1984); 3) Bertino (1985); 4) Davies, Lambert, & Findlay (1989); 5) Foley, Wallace, & Chan, (1984); 6) Hall (1982); 7) Heffler (1981); 8) Larson (1982); 9) Laverson, Norman, & Shneiderman (1987); 10) Liebelt, Macdonald, Stone, & Farell (1982); 11) Norman (1984); 12) Norman (1983); 13) Paap & Roske-Hofstrand (1988); 14) Shneiderman (1988); 15) Taylor (1986); 16) Walker & Olson (1988)

interaction have reached similar conclusions regarding the trade-offs involved in the use of pure menu-based control systems, which is that menu-based control is well suited to the needs of the novice or intermittent user, but that pure menu-based control is not very well suited to the needs of the expert user. However, other authors appear to offer contrasting opinions (e.g., Shneiderman, 1980; Taylor, 1986).

On the surface, Shneiderman and Taylor appear to suggest that software and technological advances have alleviated many of the shortcomings of menu-based selection, and that menu-based selection is well suited to the needs of both the novice and the expert user. A closer examination of these works, however, reveals that one of the advances to which each of these authors allude is the provision of shortcut methods (e.g., type ahead) that allow the user to bypass the selection of commands from menus. Such functionality is not indicative of menu-based selection; instead, it creates a mixed menu/command language-based system in that commands can be selected with or without the aid of on-screen guidance (i.e., a menu). For this reason, the comments made by these authors pertain to mixed systems and do not reflect characteristics of a purely menu-based approach.

Direct manipulation – strengths and weaknesses. Margono and Shneiderman (1985) suggested that the central goal of direct manipulation systems is to give the user an impression or feeling of close contact with the objects and actions of interest. They suggested this is achieved by minimizing the distance between what the user intends to do and what the system can do, and by giving the user the feeling they have control over the objects in the task domain. Margono and Shneiderman wrote:

> When these aspects are included in an interface, they may make the user's learning process easier, since little effort is needed to get from intention to action and from output to interpretation. Hence, the goal is that the user should learn the task domain instead of the computer system. Thus, a good direct manipulation interface eliminates the visibility of the computer system and its interface from the user, i.e., it should appear to the user that the task domain is manipulated directly. (p. 154)

Likewise, Hutchins, Hollan, and Norman (1985) suggested that achieving the above mentioned goal:

> ...means that the translation is simple and straightforward, that thoughts are readily translated into the physical actions required by the system and that the system output is in a form readily interpreted in terms of the goals of interest to the user. (p. 317)

Thus, as was the case with menu-based selection, direct-manipulation interfaces provide a source of information for the intention and selection stages of an interaction. By definition, direct-manipulation interfaces also provide the information and the mechanism for the execution stage. This style of control is further characterized by the provision of clear and immediate feedback that allows the user to evaluate the outcome of a transaction.

A review of available literature suggests that direct-manipulation interfaces: are easy to learn; are easy to remember; reduce error rates and consequently the need for error messages; provide clear feedback; create a sense of user understanding/outcome predictability; create a sense of user control; reduce user anxiety; promote easy translation of user thoughts into

actions; encourage exploration; lead to high user satisfaction; and allow for easy reversal of actions (see Table 3).

However, further examination of the literature reveals that the use of direct manipulation does involve trade-offs. Direct manipulation has been criticized for: being distracting to experts (i.e., forcing them to locate objects and actions on the screen when they already have clearly formed intentions that they wish to express as efficiently as possible); not being as efficient a means of control for expert users as would a command language; not aiding in the transition to the use of a command language; utilizing graphic representations that take up excessive screen space; fostering some situation/action combinations that may be cumbersome; relying on analogical representations that can be misinterpreted by users; utilizing powerful complex commands that reduce generality and flexibility; often being weak in terms of macro techniques; having difficulty with history tracing; and requiring more programming effort and more system resources (see Table 4).

It is evident that the trade-offs that have been associated with the use of direct manipulation closely mirror those associated with the use of menu-based selection. As was the case with menus, the general opinion is that direct manipulation is a good medium for providing user support but that it may not adequately meet the needs of users who desire maximally efficient methods of system control (see Smith and Mosier, 1986).

Command Language Control – Strengths and Weaknesses. Command-based control can be characterized as the minimal-frills approach to sequence control. Efficiency of input is typically emphasized to the exclusion of ease of learning and initial ease of use. Norman (1984) noted that command-based systems (c.f., operating systems) traditionally

TABLE 3

Advantages of Direct Manipulation-Based Control by Supporting Authors

ADVANTAGES OF DIRECT MANIPULATION-BASED CONTROL:	AUTHORS: 1	2	3	4	5	6	7	8
Is easy to learn.	✓	✓	✓		✓	✓	✓	✓
Is easy to retain.	✓	✓	✓	✓	✓	✓		
Reduces error rates.		✓	✓		✓	✓		✓
Provides clear feedback.	✓			✓	✓	✓		
Creates a sense of user understanding/outcome predictability.		✓		✓	✓		✓	
Creates a sense of user control.		✓		✓	✓			
Reduces user anxiety.				✓	✓			✓
Promotes easy translation of user thoughts into actions.	✓	✓	✓					
Encourages exploration.			✓		✓			
Leads to high user satisfaction.			✓		✓			
Allows for easy reversal of actions.				✓				✓

1) Hutchins, Hollan, & Norman (1985); 2) Margono & Shneiderman (1985); 3) Shneiderman (1988); 4) Shneiderman (1987); 5) Shneiderman (1983); 6) Shneiderman (1982); 7) Smith & Mosier (1986); 8) Ziegler & Fahnrich (1988)

TABLE 4

Disadvantages of Direct Manipulation-Based Control by Supporting Authors

DISADVANTAGES OF DIRECT MANIPULATION-BASED CONTROL:	AUTHORS 1	2	3	4
Expert may find it distracting to have to locate objects and actions on the screen.				✓
Is not as efficient a means of control for experts as is a command languages.		✓		
Does not encourage/facilitate the learning of the expert mode of interaction (i.e., a command language).		✓		
Utilizes graphic representations that take up a lot of screen space.			✓	
Fosters some situation-action combinations that may be cumbersome.				✓
Relies on analogical representations that can be misinterpreted by users.			✓	
Utilizes powerful complex commands that reduce generality and flexibility.	✓			
Is often weak in terms of macro techniques.				✓
Often experience history tracing difficulties.				✓
Requires more programming effort and more system resources.	✓			✓

1) Hutchins, Hollan, & Norman (1985); 2) Shneiderman (1988); 3) Shneiderman (1983); 4) Smith & Mosier (1986)

provide little or no user support for the intention, selection, or execution stages of human-computer interaction. Instead, the user is expected to memorize the appropriate commands.

A review of available literature suggests that command languages: are preferred by experts; are a faster and more powerful mode of control for experts; provide experts great versatility/flexibility; require very little screen space; and support/promote user initiative (see Table 5).

Further examination of the literature reveals, however, that command languages have been criticized for: being difficult to learn; creating a high memory load on the user (i.e., being difficult to retain); not providing sufficient user help; requiring substantial training; resulting in higher error rates, especially for novices; creating early learning problems that stifle user motivation; not being able to guide users or prevent them from trying commands in inappropriate contexts; and creating a feeling of indirectness (see Table 6).

It is clear that the trade-offs that have been associated with the use of a command language-based style of control are virtually the inverse of those associated with the use of menus and direct manipulation. In other words, command languages have generally been praised in terms of the efficiency with which expert users can execute commands. However, for all stages of user activity, such systems have been widely criticized for not providing sufficient support for those users who need it. As a result, without other forms of support (e.g., menus), such systems are not likely to be efficient, except for the most hardened user.

Combining Control Styles

Since the virtues of a command-language style of control are directly

TABLE 5

Advantage of Command Language-Based Control by Supporting Authors

ADVANTAGES OF COMMAND LANGUAGE-BASED CONTROL:	SUPPORTING AUTHORS:							
	1	2	3	4	5	6	7	8
Experts prefer command languages over other modes of control.		✓			✓	✓	✓	✓
Are a faster and more powerful mode of control for experts.	✓		✓	✓	✓			✓
Provide great versatility/flexibility.	✓		✓	✓	✓			✓
Require little screen space.				✓				
Support/promote user initiative.					✓			

1) Heffler (1981); 2) Larson (1982); 3) Norman (1983); 4) Paap & Roske-Hofstrand(1988); 5) Shneiderman (1988); 6) Streitz (1987); 7) Streitz, et al. (1987); 8) Taylor (1986)

TABLE 6

Disadvantages of Command Language-Based Control by Supporting Authors

DISADVANTAGES OF COMMAND LANGUAGE-BASED CONTROL:	SUPPORTING AUTHORS:								
	1	2	3	4	5	6	7	8	9
Are difficult to learn.	✓			✓	✓	✓		✓	✓
Create a high memory load on the user.	✓				✓	✓	✓		✓
Do not provide sufficient user help.		✓		✓		✓			
Require substantial training.						✓			
Result in higher error rates especially for novices.							✓		
Create early learning problems that stifle user motivation.							✓		
Do not guide users or prevent them from trying commands in inappropriate contexts.					✓				
Create feelings of indirectness.			✓						

1) Bosser (1987); 2) Larson (1982); 3) Margono & Shneiderman (1985); 4) Norman (1983); 5) Paap & Roske-Hofstrand (1988); 6) Shneiderman (1988); 7) Streitz, Lieser, & Wolters (1989); 7) Streitz, Spijkers, & van Duren (1987); 9) Taylor (1986)

pitted against the deficits of menus and direct manipulation, and vice versa, it is easy to visualize how combining or blending these styles would generate a number of synergistic effects. In fact, it has been projected that these styles could be combined or blended together so as to yield a control/dialogue system that would: effectively serve a broad user community (i.e., one that varied in experience and skill); support learning as well as ease of execution; reduce annoyance/frustration due to incompatibility between interface features and user needs; adapt to changes within a given user (e.g., experts slip back into casual user status, novices become experts, etc.); adapt to organizational changes (e.g., employee turnover); and encourage/facilitate the learning of an expert mode of interaction (see Table 7).

Empirical Research

In the sections that follow, empirical evidence is examined relating to: 1) the design of naming based techniques for selecting commands from menus; 2) the comparison of naming and pointing techniques; 3) the formation of rules for producing menu-bypass codes/shortcuts (i.e., developing command languages for menu-based user interfaces); 4) the differential suitability of the base control styles; and 5) the mixing or coterminous use of multiple styles.

Selecting commands from menus by naming. A command can be chosen from menus (i.e., a displayed list of options) either by entering an item identifier or by pointing at the desired selection. Perlman (Experiment 2, 1985) examined the former.

Perlman compared different types of item identifiers (i.e., codes used to select items from menus) in terms of the time it took subjects to make cued

TABLE 7

Advantage of Mixed Control by Supporting Authors

ADVANTAGES OF MIXED CONTROL:	SUPPORTING AUTHORS:													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Better able to serve a user community that varies in skill and experience.		✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	
Reduces annoyance/frustration due to mismatches between interface features and user needs			✓		✓	✓	✓	✓	✓				✓	✓
Accommodates changes within a given user (e.g., novices become experts over time).		✓	✓	✓	✓	✓	✓	✓	✓				✓	✓
Accommodates organizational changes (e.g., employee turnover).								✓						
Supports learning as well as ease of performance.	✓	✓	✓	✓	✓	✓			✓		✓	✓	✓	
Encourages/facilitates the learning of expert strategies.		✓					✓							
Is preferred over single strategy systems.	✓					✓								

1) Antin (1988); 2) Badre (1984); 3) Bertino (1985); 4) Davies, Lambert, & Findlay (1989); 5) Eason (1976); 6) Gilfoil (1982); 7) Hall (1982); 8) Heffler (1981); 9) Larson (1982); 10) Maguire (1982); 11) Norman (1981); 12) Shneiderman (1988); 13) Streitz, Spijkers, & van Duren (1987); 14) Taylor (1986)

selections. He had subjects select items from a menu of eight alphabetized computer terms displayed in a vertical list. The list was continuously displayed in the upper left hand corner of the screen. The task for the subject was to select, as quickly as possible, the listing that corresponded to a target word that appeared in the lower right hand corner of the screen. The four types of item identifiers he examined were: 1) compatible letters (i.e., the first letter of the menu item served as the item identifier); 2) incompatible letters (i.e., letters "a" through "z" were paired with menu items that had a different first letter); 3) compatible numbers (i.e., a number that matched the ordinal position of the item within the vertical list of options served as its identifier/referent); 4) incompatible numbers (i.e., a number that did not match the ordinal position of the item within the vertical list of options served as its identifier/referent). Subjects were exposed to all conditions.

Perlman found that compatible letters were the best selectors (1.14 s), followed by compatible numbers (1.47 s), but that the trend was reversed for incompatible selectors. In other words, incompatible numbers lead to faster selection times (1.93 s) than incompatible letters (2.22 s). Perlman wrote:

> The application of the results to the design of menu systems is clear. The best that can be done is to use compatible letters as selectors, but this can only be done if the designer has full control over the contents of menus. If the pairing is done automatically, such as by a program to select files, then using letter selectors can lead to the worst case, incompatible letters, and numerical selectors would be preferable. (p. 319)

Perlman and Sherwin (1988), however, appear to second guess the generalizability of Perlman's (Experiment 2, 1988) earlier findings. They

question whether the results obtained with the number selectors would have differed had the vertical format of the menu corresponded to the horizontal format of the number keys. They hypothesized that matching the physical layout of the menus and the keys used to select them would produce an interface that would have direct manipulation like characteristics. They suggested that such an interface would promote ease of use by being cognitively more intuitive and physically more response compatible.

In lieu of their questions regarding Perlman's earlier results, Perlman and Sherwin opted to reevaluate the effectiveness of number keys when a horizontal menu format was used. They also opted to look at performance with matched and unmatched menus using the traditional IBM two column function key pad as the selection device. In all, they crossed four menu formats (i.e., a two column horizontal matrix, a two column vertical matrix, a single horizontal row, and a single vertical column) with the two input device formats (i.e., horizontally formatted number keys and function keys formatted in the traditional IBM two column matrix). Crossing these factors produced six conditions in which the menus were formatted differently from the input device and two in which the selector keys and the menus were formatted in the same way.

During the course of the experiment, the numbers 1 through 10 were used as menu items. These numbers were continuously displayed on the screen in one of the four menu formats being tested. Each trial began when a menu item (i.e., a number) became highlighted in reverse video. The subjects then attempted to enter the key corresponding to the highlighted number. Each subject completed eight blocks of trials, one with each device by menu format condition.

Perlman and Sherwin found that the two compatible layouts were significantly faster than the other layouts (i.e., 875 and 968 ms versus 1016, 1036, 1038, 1096, 1158, and 1198 ms). The IBM function keys with a compatible menu format lead to the fastest performance times (875 ms).

A Scheffe post hoc test revealed that mean differences greater than 52 milliseconds were significant at the .05 level. Using this criterion for significance, it was apparent that the majority of the above cited means differed significantly from one another. These authors noted, however, that the largest differences were in the 300 millisecond range and that differences of this magnitude are barely perceivable and do not seem to be large enough to make a great difference in productivity.

The meaningfulness of these results and conclusions seem questionable, however, given the nature of the experimental conditions and the task subjects performed. The selectors/item identifiers used in this study doubled as the stimuli (i.e., the menu items). The results would have been more credible if meaningful words or icons had been used as the stimuli. The fact that the largest timing differences were barely perceptible was not surprising, given that the task followed a simple stimulus-response paradigm and that within this paradigm the item identifiers/selectors were used as the stimuli.

Bayerl, Millen, and Lewis (1988) conducted a study similar to that carried out by Perlman and Sherwin. However, when they investigated the interaction between the physical layout of function keys and the layout of on-screen labels (i.e., menus) they did so in a more face valid manner.

Subjects entered simulated fast food orders by pressing the function keys F1 through F8. Eight order/menu items were presented in three different menu formats. One format was matched to the classic IBM five

row by two column function keypad. Menus displayed in this format were presented in the lower left hand corner of the screen. A second format matched the horizontal row of function keys found on the most recent generation of keyboards. Menus displayed in this format were presented along the bottom of the screen. The third format consisted of a single vertical list. In this case, the menu was displayed in the center of the screen.

Crossing the three menu formats with the two key layouts resulted in six experimental conditions. Each subject completed forty trials with each of the six conditions. For each trial, a box in the top right corner of the screen showed three food items and instructions to "Process these orders." The three food items were replaced with a randomly generated list after three function key presses.

Analyses showed that the two combinations of key layouts and menu formats that were spatially consistent lead to significantly faster response times (combined mean = 1.996 s) than did the four conditions where the key layout and the menu format did not match (combined mean = 2.274 s). These authors also found that the overall mean response time for the horizontal function key layout was significantly faster than that of the traditional matrix layout (mean difference = 0.081 s). The fastest combination was the horizontal menu and the horizontal function key layout (mean = 1.97 s). This finding was inconsistent with Perlman and Sherwin's (1988) finding that the combination of the five by two matrix of function keys with a matched menu was faster than the horizontal row of number keys with a matched menu.

Menu selection – comparing pointing and naming. Karat, McDonald, and Anderson (1986) compared two pointing-based and one

naming-based technique for selecting items from menus. The three selection techniques examined were: 1) on-screen pointing with a finger; 2) off-screen pointing with an auxiliary pointing device; and 3) typing of alphabetized letter selectors. On-screen pointing was made possible by mounting a membrane touch panel on the front of the task display. Off-screen pointing was achieved by moving an optical mouse positioned on the side of the subjects preferred hand. Typed inputs were made on a standard IBM PC keyboard.

In the process of evaluating these techniques, Karat, McDonald, and Anderson carried out two studies in which they collected both performance and subjective data. In each study, subjects began by completing target acquisition/practice tasks with one of the three selection devices. They then used the same selection device to perform menu selection tasks with one of two mock computer applications. Subjects then completed another set of practice trials, followed by menu selection trials with the second mock application. Subjects then switched to a second selection device and completed the above mentioned scenario with that device. The cycle was then repeated for the third selection device.

The two studies were identical except that: the movement ratio on the mouse which was changed from 1:1 in the first study to 1:2 in the second; in the second study typing, speed and sex were used as grouping/category variables; and the amount of practice trials between each set of application tasks changed (i.e., 25 trials in the first study, 288 in the second).

The target acquisition/practice tasks required that subjects select a 14 x 13 mm target button which appeared in random locations on the task display. Each target button was randomly labeled with a letter “a” through “z”. Subjects were instructed to select the target as quickly as possible

while keeping selection errors to a minimum. A selection error was recorded if: the subject's finger was not inside the target area when it made contact with the screen; or the on screen pointer was not inside the target area when one of the mouse buttons was pressed; or the subject pressed a key that did not match the label inside the target button.

In the first experiment, the results from the practice trials revealed that subjects were faster at making selections with the touch panel and the keyboard (0.8 and 1.1 s per selection) than they were with the mouse (2.7 s). In the second experiment, it was found that selections with the touch panel were significantly faster than selections with either the keyboard or the mouse (.56 s verses 1.16 s and 1.30 s, respectively).

These authors also noted that for target acquisition/practice trials there were "greatly" different error rates for the different devices. During the second experiment it was observed that the error rate was relatively low with keyboard selectors (roughly 5%). In contrast, the error rates for the touch screen and the mouse were much higher (27% and 22%, respectively).

As was noted earlier, subjects also performed menu selection tasks with two mock computer applications. One of these mock applications simulated a computer telephone aid and the other a personal appointment calender. With each application, subjects had to select one or more options from hierarchically arranged menus. Half of the tasks subjects performed required that they enter typed information (such as names, comments, appointments, or telephone numbers) into fields that appeared following a menu selection.

In discussing the results from their Experiment 1, Karat, McDonald, and Anderson (1986) noted that performance measures from the menu

selection/application tasks showed an improvement of approximately 10 percent for the touch panel over the keyboard, and 10 percent for the keyboard over the mouse. However, the difference between the touch screen and the mouse was the only one that reached significance. The menu selection results for Experiment 2 were very similar to those for Experiment 1, except for the fact that in Experiment 2 the touch screen was found to be significantly faster than either the keyboard or the mouse.

Following the completion of all tasks, subjects were also asked a series of questions comparing the different combinations of input devices (touch screen, mouse, keyboard) and task types (practice, selection only, and selection with typing). In Experiments 1 and 2, subjects expressed a preference for the keyboard and the touch panel over the mouse when the type of task was selection only or selection with typing. In Experiment 1, the keyboard and the touch panel were given similar preference ratings for the selection only tasks. In Experiment 2, the touch panel was preferred over the keyboard when the task type was selection only. In both experiments, when the tasks involved both selection and typing, the keyboard was the most preferred device.

These authors pointed out that: "...both the subjective evaluations and the performance data suggest that the touch panel and the keyboard are better menu-selection devices than the mouse." However, they concluded with the following caveat:

> We do not suggest that these studies render previous findings invalid, only that more attention needs to be given to the nature of the dialogues for which the device is to be used, and the skills of the users. At the very least we must conclude that less "natural" devices such as keyboards, can in some circumstances be preferred and lead to better performance than more "natural" pointing devices such as mice. (p. 87)

Several additional factors warrant consideration when interpreting these results. First, no attempt was made to account for hand position. Subjects were allowed to position their hands anywhere they liked during and between trials. Since typing was the terminal event in the selection with typing tasks, and since all other tasks simply involved selecting on-screen targets, it was probable that the subject's pointing hand was already very close to the screen whenever they attempted to make a touch screen selection. In many, if not most, menu based applications this is not likely to be the case. Thus, the findings regarding the touch screen may have been unduly favorable, at least from a practical standpoint.

A second important factor relates to the type of menu selectors that were used in the keyboard condition. Menu items were presented in a vertical list. Alphabetized menu selectors were displayed to the left of the menu items. It was surprising that the keyboard faired as well as it did, given the type of selectors that were used. As was previously noted, Perlman (1985) examined several different types of item identifiers and found that incompatible letter selectors (e.g., alphabetized letter selectors) produced the worst performance. Perlman's findings lead Paap and Roske-Hofstrand (1988) to write:

> Incompatible letter identifiers should be avoided. The costs for incompatible letters is over one second per selection relative to the optimal case of compatible letters and two-thirds of a second relative to the digit-identifier compromise. (p. 216)

It therefore seems probable that the keyboard would have faired even better as a selection device had compatible selectors been employed. How much better remains to be seen.

Developing command languages for menu-based interfaces. Bypass codes/shortcuts (i.e., command languages for menu-based interfaces) differ from menu selector codes in that the user must recall the codes from memory in order to use them. In contrast, users only need to be able to recognize menu selectors in order to use them.

A number of authors have gleaned principles about how to make command codes (e.g., bypass codes/shortcuts) more memorable. For example, Green and Payne (1984) developed three sets of command codes for 26 commands found in an EMACs-based word processor by varying the consistency, congruence, and mnemoticity of the command codes. Using paper-and-pencil tests of free and prompted recall, they found that all three sets of codes they had constructed yielded significantly better recall scores relative to those from the EMACs-based editor.

These authors suggested that the reason the EMACs codes faired so poorly was that confiicting organizing principles were used in the construction of these codes. One of the organizing principles found in the EMACs command set is that “Ctrl” means small and “Esc” means large (e.g., Ctrl-F moves the cursor forward one character, whereas, Esc-F moves the cursor forward one word). However, in other instances “Ctrl” is used to denote forward and “Esc” is used to denote backward (e.g., Ctrl-V views the next page, whereas Esc-V views the previous page). These two rules conflict in at least two ways, since “forwards” and “backwards” are coded differently; and “Ctrl” and “Esc” have different meanings. In discussing their findings, Green and Payne wrote:

> No single experiment is adequate to support a guiding principle, but if further studies confirm our results the list of empirically-tested guiding principles for command language

> designers can be extended by a singularly important addition: the language should be structured so as to afford novice users clear and consistent organizing principles. (p. 16)

Walker and Olson (1988) formulated a set of rules for producing bypass codes/menu shortcuts. Bypass codes/menu shortcuts are command codes that function as alternatives for activating commands from a menu.

The rule structure Walker and Olson developed was hierarchical. The highest level of the structure referred to broad classes of actions and was denoted by the type of modifier key that was used (i.e., the Esc key was used for all system-related commands such as save; the Alt key was used for all deletions; and the Ctrl key was used for all other types of actions). The second level of the structure referred to specific actions and objects and was coded mnemonically (i.e., Alt FW meant to delete forward a word, while Alt BC meant to delete back a character).

Utilizing their rule structure, Walker and Olson developed a new set of command codes for 56 frequently used EMACs commands. These researchers then compared a subset of the command codes they had developed (n=25) against the original EMACs command codes in terms of learnability.

Walker and Olson chose to use a within-subjects design to make the above mentioned comparison. Subjects were first given a study sheet containing the commands and one of the two sets of command codes. The study sheet was formatted so as to highlight the groupings of similar commands. After reviewing the study sheet for two minutes, subjects were given an identical sheet minus the command codes. Subjects had two minutes to fill in the codes which they could remember. This study-test

pattern was repeated four times with the first set of codes. Subjects then read a randomly ordered list of commands at a rate of one command per five seconds and were asked to write down the appropriate command code. The same overall pattern was then repeated for the second set of command codes.

An analysis of variance, performed on the structured test data, revealed no overall difference between the two code sets. The results for the randomly ordered list were quite different, however. Overall random recall was far poorer for EMACs codes than it was for the codes developed by Walker and Olson. In fact, subjects recalled approximately twice as many of the Walker and Olson codes as they did EMACs codes (i.e., roughly 20 out of 25 versus 10 out of 25).

Other analyses conducted by these authors underscored the importance of using consistent rules (i.e., first letter mnemonics). Walker and Olson observed that, relative to the other command codes, the Cut and Copy command codes had lower scores on the random recall test. A closer inspection of the Cut and Copy command codes revealed that these codes constituted the only case where there was not a distinct one-to-one mapping between the first letter of a command and the action/object identifier portion of the command code (i.e., “C” for Cut and “K” for Copy). These authors found that 14 of the 27 times that subjects failed to correctly recall the command codes for Cut and Copy, “C” was substituted for “K” or “K” was substituted for “C”.

These authors concluded by defending the fact that the coding scheme they developed typically required three keystrokes per code, whereas the majority of the EMACs codes required only two (i.e., Ctrl EC versus Ctrl W for copy). They wrote:

> Often we hear that designers try to avoid multiple-key keybindings. We argue that, according to Card, Moran, and Newell (1983), each extra keypress costs only .28 seconds on average, whereas each cognitive process is much slower, on the order of 1.35 seconds. Simple retrieval is slower yet. Errors require a retrieval, action, recognition of the error, retrieval of how to correct it and the subsequent keystrokes. The time involved in making and correcting a keybinding that is difficult to remember is far more costly than the extra keystroke that is required in an easily remembered keybinding. (p.205)

Differential suitability of control styles. A review of the literature reveals that the suitability of a given control style, or styles, for use with various user and task characteristics is an often discussed topic (see Tables 1-6). The following section contains a detailed review of available research findings.

Whiteside, Jones, Levy and Wixon (1985) sought answers to the following research questions:

> 1) are there large user performance differences between interfaces?
> 2) what style of interface is best for what level of user?
> 3) can performance differences be attributed to interface style? (p. 185)

Their effort entailed the comparison of performance and preference data obtained from three classes of users (novice, transfer, and system user) performing simple file manipulation tasks on one of seven different interactive software systems (one menu-based, two iconic/direct manipulation-based, and four command language-based systems). Six of the seven systems tested were commercially available products.

As expected, performance differences were found between: 1) users

with little or no computer experience (novice); 2) those who used computers daily, but had never used the system assigned to their condition (transfer); and 3) those who had used computers daily and had used the system assigned to their condition for at least a month (system users). Contrary to expectations, no experience by system type interaction was found. In fact, it was found that all classes of users performed best with one of the command language interfaces and all classes of subjects did worst with the menu-based interface, with one of the icon-based interfaces coming in a close second. This finding is counter intuitive in light of the conventional wisdom outlined in the previous chapter. A closer inspection of this study, however, raises questions as to its ability to assess the effects of control style.

Taylor (1986) wrote regarding the Whiteside et al. (1985) study:

> It was found that there were two major confounding elements in this experiment that contributed greatly to their findings. The first problem was that the researchers used seven different computer operating systems running on several different types of computer hardware. Measurement of performance across different operating systems and hardware using 'time to complete task' as part of the measurement of performance ignores the tremendous differences that can occur due to differences in underlying concepts and implementation differences in file operations. Plus the fact that the interfaces used varied greatly in underlying concepts apart from those pertaining directly to mode of interaction. (p. 44)

The second point made by Taylor refers to the fact that these authors attempted to assess the effects of control style without first equating the systems along other dimensions. Thus, any results regarding the effect of control style cannot be separated from other confounding variables such as

differences in the internal consistency of the systems tested. This makes the data uninterpretable with regard to all but the first research question posed by Whiteside et al. (1985).

Whiteside et al. (1985) acknowledged this confound by writing:

> There are clearly large differences in the systems we measured. Our conclusion is that the care with which an interface is crafted is more important than the style of interface chosen, at least for menu, command, and iconic systems. (p. 190)

Several additional factors that may have masked possible experience by control style effects were the length of the experimental session (1 hour) and the fact that on-line help was the primary training mechanism for novice and transfer users. The brevity of the experimental session limits the generalizability of results for the transfer and novice users to that of individuals during their first hour of interaction with one of the systems in question.

The decision to provide a minimal amount of off-line information during training and task conditions makes any conclusions regarding the effect of control style even more suspect. The authors acknowledged that across systems there was a notable discrepancy in the quality of online help. This was particularly critical in that off-line help was limited to non-directive comments by the experimenter.

Margono and Shneiderman (1985) conducted a study somewhat similar to the Whiteside et al. (1985) investigation. They sought to determine the degree to which a direct manipulation interface (the Apple Macintosh) and a command language style of interface (the IBM PC running MS-DOS) differed in terms of "user-friendliness". This

assessment was made by comparing the speed of execution, errors, and satisfaction of 30 novice users who performed file manipulation operations on both systems.

Again, as in the Whiteside et al. (1985) study, experimental sessions were brief (i.e., 30 minutes on each of the systems tested), and the tasks required the use of a relatively small set of commands. Sessions were divided into an introductory phase, followed by a practice phase and a task phase on each of the two systems tested.

The results indicated that the Macintosh lead to faster task completion during the testing phase. In addition, fewer errors were committed with the Macintosh, and the mean satisfaction level was significantly higher with this system.

This study suffered from many of the problems associated with the Whiteside et al. (1985) study in that the results may have been influenced as much by the idiosyncrasies of the systems tested as by the style of control employed. The generalizability of the results is also limited by the fact that all subjects were either naive or transfer users and received only 30 minutes of exposure to each of the systems. Therefore, the only defensible conclusion that can be drawn is that when performing simple file manipulation tasks, extreme novices appear to prefer and perform better with the Macintosh as opposed to the IBM PC with MS-DOS.

Gilfoil (1982) conducted an experiment designed to assess the performance and satisfaction of users as they progressed from novice to expert status on a series of computer tasks. He sought to answer the following research questions:

1) What are the underlying principles of a computer user's cognitive structure and how do they develop over time?
2) How does the development of this cognitive structure affect task performance?
3) At what point in development of this cognitive structure will a user actively choose to switch from computer (menu-based) to user (command language) control for performing tasks? Why?
4) What is the relationship between performance with a computer system and user satisfaction with that system? (p.246)

These research questions were addressed by measuring: 1) cognitive development through the free-recall of system commands; 2) task performance in terms of errors, task completion time, and referrals to the help file; and 3) user satisfaction as assessed by a series of online questions. All measures were collected for each task session.

The subjects for this study were four computer-naive persons. They performed six computer tasks (creating, inputting, storing, accessing, editing, displaying, and outputting data) at each of 20 task sessions over a one month period. Participants started out working with menu-driven dialogue in which commands were selected by typing a number positioned to the left of the desired command. They were instructed at the beginning of each session, however, that they had the option of switching to a command-driven dialogue whenever they felt that they were ready. In the case of the command mode, commands were selected by typing an abbreviated form of the actual command.

An analysis of user choice regarding the active style of control revealed that the four novice users in question transitioned to a command driven dialogue after approximately sixteen to twenty hours of task

experience. With experience, users were shown to choose, perform better, and be more satisfied with a command driven dialogue.

Changes in user preference, task errors, user satisfaction, and the usage of the help facility over the course of the 20 sessions lead Gilfoil to some interesting conclusions regarding the users' development of a cognitive schema for the task environment:

> Users with relatively unstructured cognitive schemes for the task environment choose a menu mode for performing tasks, make many "semantic" errors, and refer to available help files as an aid to learning the task environment. Users at this stage are moderately satisfied with interactive sessions. As users develop a more organized mental representation of the task environment, they tend to prefer a command driven dialogue, perform tasks in significantly less time, make fewer semantic errors (but not significantly less syntactic errors) and are generally more satisfied with interactive sessions. (p. 249)

Taylor (1986) drew upon the work of Gilfoil and others when he postulated that the dialog needs of the user vary, depending on their level of experience. He sought to answer the following research questions:

> 1. Will the availability of multiple dialog modes in a user system interface that accommodates different levels of user experience make a difference in overall user performance and satisfaction?
> 2. At what level of experience will a user prefer a user-directed dialog (i.e., a command language) over a computer-directed dialog (i.e., menus)? (p. 47)

Taylor had novice and expert computer users work with one of three control styles (menu, command language, or menus and command language together) to perform a series of data base tasks (updating an address book and associated Christmas card list in dBASE III™).

Experimental sessions were broken down into a 30 to 45 minute introductory/practice phase, two 45 minute task phases, and a debriefing stage where subjects completed a satisfaction survey. The length of the session was intended to give the subjects ample time to become familiar with the interface they were to operate.

Like Gilfoil (1982), Taylor sought to assess performance as well as user satisfaction. The performance measures he examined were: 1) task completion time; 2) a quantitative score reflecting correctness of required actions; and 3) choice behavior (i.e., percentage of commands selected from menus versus percentage of commands selected with the command language under the choice condition). Satisfaction was assessed through a series of Likert-type items.

The results of this study were summarized by Taylor as follows:

(a). The results did not provide evidence to support the difference in expert and novice problem solving techniques as the basis for why experts and novices need different types of user-system interfaces.
(b). The results did not support the normative theory developed from other researcher's opinions, observations, and research that experts would find menus unsatisfactory as a user-system interface and prefer a user directed dialog mode.
(c). A user-system interface with multiple dialog modes for a population with varied experience levels was not statistically better than an interface with menu only but was better than command language only.
(d). A user-system interface with only menus was consistently better for experts in both performance and satisfaction over the other user-system interfaces used in this research.
(e). A user-system interface with only command language was consistently worse for novices in both performance and satisfaction than either of the other two interfaces used in the research.
(f). Subjects preferred multiple dialog modes but, when given multiple dialog modes, generally did not perform as well as those with menus only.

(g). When subjects were given multiple dialog modes, choice of mode to perform tasks was not correlated to either technology or dBASE experience.

(h). Satisfaction with a dialog mode seems to be based on previous experience with dialog modes, ease of using the dialog mode for the task, speed, and a bias either for or against a specific dialog mode or dialog style based on a subject's own unique set of preferences and general experiences and not just technology and application specific experience. (p. 114-115)

As is evident, the bulk of the results did not support the normative theory (i.e., that experts would prefer and perform better with a command language style of control) or the notion that providing multiple styles of control will have a synergistic effect on performance. The following are excerpts from Taylor's explanation of these results:

> An important factor in this experiment was the speed of dedicated micro system for each subject that displayed the full screen menus. It was exceptionally fast even though it painted the screen. The subject was also allowed to type ahead if they so desired. These features allowed subjects to rapidly progress through many of the options menus and type in data requirements while the menu was being generated. This speed of processing is one of the activities prized by many users especially ones that are experienced. Speed of processing is also one of the main advantages of command language, e.g. one does not have to progress through many levels of menus to perform an action. Thus, for experts, the speed of the menu seemed to have been sufficiently fast that they were satisfied. (p. 108)

> Some of the problems encountered by dBASE experienced subjects who used the command language seemed to be caused by their having fallen back to a casual user status through lack of recent use that could have kept them at the expert level for recalling and using the commands. They did not lack general experience with dBASE or technology, they just lacked adequate current experience that caused them to have a problem remembering how to form the syntax. (p. 112)

In reviewing Taylor's work, the present author noted an additional factor that may have contributed to the discrepancy between the menu-based and the command language styles of interaction. This was the fact that the hard copy help material regarding menus was directed toward the experimental task, whereas the command language handout contained no references to the experimental task.

These factors suggest that the menu-based system was efficient, effective, and accepted at least in part due to improvements that gave it command language-like characteristics. These factors also suggest that the schema for identifying dBASE III experts was ineffective and that the training/help material may have biased the results of this study in favor of the menu-based style of control. When taken in conjunction with the strong subject preference for having access to both styles of control, these factors suggest that further effort should be devoted to understanding the advantages and disadvantages of the various control styles and the possible synergistic effects of combining them. Taylor, in fact, wrote:

> In conclusion, it was found that even though multiple dialog modes did not improve performance and satisfaction over menus (although much superior to having just command language) almost 100% of the 98 subjects expressed a preference for multiple dialog modes. When subjects were given a choice of dialog modes the subjects as a whole, regardless of experience, split 60/40, menu to command language. Thus, strong consideration should be given to having multiple dialog modes in a user-system interface for application systems that are to be used by a population with a diverse experience background. (p. vi)

Using a within-subjects design, Antin (1988) compared the use of command codes, menu-based command selection, and a combination of the two in terms of objective performance and subjective preference. The

subjects for this study were six novice and six expert users of a Panel Layout Automated Interactive Design™ system.

Novice users received a one hour training session, after which they and expert users were treated alike. At the beginning of each of the three twenty five minute trial blocks subjects received a brief introduction to the upcoming control style. Each trial block required that the subject follow a scripted set of instructions, that, if successfully completed, resulted in the formation of a pseudo three dimensional image.

Antin measured performance in terms of: 1) a task time/completion score (i.e., task completion time divided by percentage of task completed); 2) response time (i.e., time from prompt to the first keystroke of a control entry); 3) input time (i.e., time from first keystroke to last keystroke in a control entry); 4) incorrect actions (i.e., making a control entry that did not contribute to task completion); and 5) inefficient actions (i.e., acting in a manner that required a greater number of steps than was necessary for task completion). Subjective reactions to the various control styles were assessed by asking users to provide an overall rating from "poor" to "excellent" for each of the control modes, and a questionnaire based upon that used by Mount, Rudisill, and Schulze (1984).

The results of this study indicated that: 1) the command code style of entry led to task time/completion scores that were significantly better than those obtained with either menu selection or the combined style; 2) there were no significant differences between the control styles in terms of the error measures taken; 3) input times for the menu-based style were significantly slower than those obtained with the command code style; 4) overall ratings were highest for the combined style; 5) eleven of the twelve

subjects perceived the combined style to be faster; and 6) all twelve subjects indicated that they preferred the combined style.

One of the most striking of these findings was that performance with the menu-based style of control was poor relative to the command code style. Given that subjects were novices with only one hour of training, this finding contradicts the conventional wisdom espoused by most human factors experts.

A closer examination of Antin's data reveals that performance with the menu-based style was especially poor in terms of the amount of time it took subjects to input a selection. However, as was the case with the Taylor (1986) study, experimental conditions appear to have unduly biased the results against one of the primary control styles, and as a result negatively biased performance with the combined mode. In this case, the negative bias was toward the menu-based style of control. In particular, an extremely awkward method of menu-based command entry was employed. To select a command from a menu, subjects had to move the cursor to the desired command and press return. Cursor movement was achieved by simultaneously pressing the control key and either the U, D, R, or L key corresponding to up, down, right, and left. In the command mode of interaction, commands were selected and entered by typing a one or two letter command code. In the combined mode, the menus were displayed and commands could be entered by either cursor movement or the use of command codes.

Despite the poor performance of the menu-based style of control and the lack of performance effects in favor of the combined mode, Antin wrote:

> Based on the unanimous preference of both experienced and novice users for the combined mode, it is recommended that menus in some form be strongly considered for a wide variety of systems. The menus should allow direct typing of coded control entries as well as selection of displayed options by pointing. (p. 181)

He continued by writing:

> However, it must be remembered that performance was better using command entry. Therefore, any implementation of menus should also allow the experienced user the option to temporarily deactivate the menu. (p. 181)

However, as alluded to earlier, great caution should be taken in interpreting the performance results of this study. It was not clear if the command style of control would have resulted in better performance had a less awkward method of menu-based command selection been employed.

Davies, Lambert, and Findlay (1989) conducted a study somewhat similar to Antin's (1988) in that they also compared control styles which involved the use of menus, command codes, or a combination of the two. They assessed the performance of relatively naive subjects working under one of four different control styles. Subjects were trained and tested on the use of six different word processing commands available in Microsoft Word™ running on the RM Nimbus. Each experimental session consisted of a learning/tutorial phase (40 min), a review phase (3 min), and a task phase (approximately 10 min). Performance was assessed by means of the following measures taken during the task phase: 1) total task time; 2) number of errors requiring experimenter intervention; and 3) the number

of times subjects consulted help information. The four control style conditions investigated were:

> 1) Mouse selection, menu always present. Subjects in this group used a mouse to choose and execute commands from a menu that was visible on the screen throughout both the tutorial and testing phases of the experiment.
> 2) Keyboard entry, menu always present. Subjects executed commands by entering an abbreviated form of the command via the keyboard. The menu was available as an external memory prop throughout both the tutorial and test phases of the experiment.
> 3) Keyboard entry, menu always absent. Subjects entered commands via the keyboard. In this condition the memory prop provided by the menu was not available during the tutorial or test phases of the experiment.
> 4) Keyboard entry, menu absent at test. Subjects entered commands via the keyboard. The menu was available as a memory prop during the tutorial phase, but was removed for the test phase. (p. 137)

The objective of the study was to determine how users make the transition from relying on an external prop provided by menus, to relying on their own memory for system commands. The following research question were examined: 1) are there any differences in efficiency when commands are executed via selection with the mouse, compared with direct keyboard entry; 2) when commands are selected by keyboard entry do different kinds of learning experiences (menus always present versus always absent versus present during training but absent during testing) have differential effects on task performance?

A series of planned comparisons was conducted. The only comparison to reach significance was between Groups three and four (i.e., keyboard-menu always absent and keyboard-menu absent at test). These groups differed in terms of total task time, consultations, and interventions

with Group four (i.e., menu absent at test) performing significantly worse on each of these indices. Davies et al. (1989) wrote regarding these results:

> A number of points can be drawn from the results of the experiment. First, after a 40 min tutorial session in which the basic commands needed to operate a word processing package were learned, performance was not enhanced by the presence of an external memory prop in the form of a permanently visible menu. Indeed, the group who never had the benefit of a memory prop showed the fastest mean performance time, achieving this without compromising performance accuracy. Even at this early stage in learning, users were able to rely on their own internal memory for basic word processing commands. Second, when provided with a permanently visible menu users tend to rely on it as a memory prop, leading to disruption of performance if the prop is removed. It appears that the keyboard, menu always absent condition encouraged active learning of system commands, leading to efficient performance in the test task. In contrast when the menu was visible during the tutorial, relatively little passive learning of commands appeared to take place, since performance in the keyboard, menu absent at test condition was significantly slower and required significantly more consultations/ experimenter interventions than the other three conditions. (p. 141-142)

The authors concluded that:

> Since no advantage was gained from the memory prop, this appears to question the value of providing a menu at all for frequently used commands. The only value of providing menus may turn out to be for consistency across different applications or for completeness within an application. (p. 142)

However, they noted:

> Not surprisingly, these results and our interpretation raise a number of further issues. The first of these concerns the relevance of the above interpretation for command learning over a much more extensive period than the single session of learning studied here. Second, in more realistic settings

> menu availability is controlled by the user, rather than imposed on the user, as in the present study. These considerations naturally raise questions concerning the general relevance of our conclusions for interface design. (p. 143)

Several additional aspects of this study raise questions as to the relevance of the conclusions drawn by Davies et al. (1989). One such aspect was the fact that performance assessments were based solely on data obtained during an extremely brief task phase (approximately 10 min).

The results of this study clearly indicate that the group which was trained with menus but had them taken away for the task phase performed poorly relative to the other groups. It is clear, however, that this group was given very little time to adjust to the change in control style. As a result, it was impossible to determine how long it would have taken this group to reach performance levels equivalent to the other groups. If under more realistic conditions (i.e., where users have control over the availability of menus) disruptions in performance were observed as users transitioned from menu dependence to shortcut usage, the relevance of these disruptions would be largely determined by the length of time that they persisted and how frequently they reoccurred.

An additional issue of importance was the fact that this study dealt with a very small set of commands (i.e., six). It would be very risky to generalize results obtained with a set of six commands to a situation in which a much larger set of commands, varying in frequency of use, was employed. Since the latter case is in fact much more indicative of the real world, the results obtained by Davies et al. (1989) should be replicated within

such a context prior to being assessed in terms of their implications for interface design.

Finally, many of today's software applications are designed to encourage learning through on-line exploration rather than formal training. If this is indeed a desirable approach, there are initial advantages to be gained from displaying commands within menus, even those for which users will quickly switch to a command code style of entry. In such cases, command codes can be displayed in the menus so that, when ready, the user can make the transition from a recognition-based style of interaction (i.e., menus) to a recall-based style (i.e., command codes).

Like Davies et al. (1989), Streitz, Spijkers, and van Duren (1987) investigated whether prior experience with menu-based selection affected the subsequent learning of a command language in comparison with learning the command language from the beginning. The basic question they wished to address was: "Does menu selection offer novices an opportunity to acquire the relevant knowledge enabling them to switch to a command language?" A secondary objective was to determine the extent to which a menu versus a command language style of control leads to the development of an overview of available commands.

In order to investigate these questions, Streitz et al. (1987) developed a series of texts of 120 words in which subjects had to repeatedly use the same seven commands. These 7 commands were nested within a larger set of 25 commands displayed in five pull-down menus on a modified Apple Macintosh™ computer. Commands were selected either by picking them with a mouse or by pressing a special command key along with a letter key.

The experimental design utilized by Streitz et al. was a 2 x 2 between-subjects design. The factors that were manipulated were the presence versus absence of prior experience, and type of help device that was available (i.e., menus versus a help window displaying the commands and command codes in a columnar format).

Utilizing the above mentioned design, Streitz et al. assessed the following: 1) transfer of learning from the initial interaction phase and the learning phase; 2) the degree to which subjects in the various conditions developed an overview of the larger set of 25 commands; 3) the degree to which they had to rely on the help devices; and 4) the speed with which they were able to edit texts.

Streitz et al. found that the group that had used a help window (i.e., a large menu containing the set of 25 commands and their respective codes in a vertical list) and command-key activation (i.e., selection by naming) in the initial interaction phase had clearly learned many of the command codes before being instructed to do so. Additional findings of interest include: 1) the group working with menus as a help device and mouse activation scored better on the overview test; 2) novice users performed faster with the menu-based style of control; and 3) the users of the menu-based style had a clear advantage in the time they spent searching for commands.

Streitz et al. wrote:

> It can be concluded that it is most favorable for novices to start with menu selection. Main reasons are that they gain a better overview over the total range of available commands and that - in the beginning - they can edit faster than novices using a control command language. Moreover, using menu selection does not interfere with subsequent learning of the codes. Apparently, menu selection and the command language are so much different from each other that any transfer does not occur at all, neither positive nor negative. ...

> In summary, the results strongly suggest that menu selection should be available in combination with a control command language, so that the user to who menu selection becomes to slow, can switch. The optimal interface design should be offering both dialogue modes allowing for degrees of freedom. For a limited set of frequent commands the experienced user can use the control command language whereas menu selection is still available for the less often used actions. In addition, this results in minimizing memory load to the user. (p. 845-846)

Unfortunately, these authors did not include an experimental condition in which menus were used throughout the experiment. Therefore, whereas it was possible for these authors to determine that novices were initially faster with menus, it was not possible to confirm the implicit hypothesis that practiced users edit faster with command codes. These authors did report that after editing 10 texts, user task times with command codes were on the average only 49 percent of the task times when menu selection was used. However, the menu selection times were calculated after users had edited only five texts.

As was the case with most of the aforementioned studies, subjects were trained on the use of, and performed tasks with, only a small set of commands. In this study, however, steps were taken to try to create a more prototypical interface and to avoid the effects of extraneous variables such as the unfamiliarity of the mouse. One step was to train subjects extensively (i.e., 4 hours) on application functionality and mouse usage (Streitz personal communication, 1989). A second step was to nest the seven commands necessary for task performance within a larger set of commands (i.e., 25 commands displayed within five pull down menus of five items each). A third step was to allow the users to refer back to the menus while they attempted to make the transition from menu dependence

to shortcut usage. Subjects were, however, forced to use the command codes to make command selections and were given instructions to try to remember them (i.e., using the menus as a help or selection device was discouraged).

Today's Graphic User Interfaces Environments

A review of the commercial software market reveals that graphic user interfaces are quickly becoming the standard approach to human-computer dialogue. The current generation of systems (i.e., Apple Macintosh™, Microsoft Windows™, Motif™, Next™, Open Look™, Presentation Manager™) do, in fact, blend direct manipulation, menu selection, and command-language styles of dialogue/control. Many aspects of these electronic work environments are designed to be manipulated directly via a pointing device (SAA CUA Advance Interface Design Guide, 1988). These platforms also provide command bar-based menus that serve as external memory aids as well as a vehicle for command selection. A command-language style of control is also provided in the form of abbreviated command codes (e.g., Ctrl C for copy).

In these systems, if the user is able to cope with the memory requirements, the command language style of control is consistently portrayed as being more efficient than its menu-based counterpart (Davies, Lambert, & Findley, 1989; Streitz, Spijkers, & van Duren, 1987). Davies et al. (1989) noted that it is thus desirable to design systems which, "...ease and facilitate the transition from the relatively circuitous menu selection method of command entry to relying on internal memory for command names and entering them via keystrokes" (p. 136).

A closer review of existing platforms (i.e., Apple Macintosh™, Microsoft Windows™, Motif™, Next™, Open Look™, Presentation Manager™) suggests they have characteristics that make the transition extremely difficult. One such characteristic is that there is no continuity of action between that required to select items from menus and that required to enter commands via the command language (i.e., bypass codes). For each of the above mentioned platforms, the focal means for selecting commands from menus is to point at the selection with an auxiliary pointing device (i.e., a mouse). In contrast, the command language that is provided requires that the user memorize highly abbreviated codes that correspond to the various commands and enter them via the keyboard. Using the command language versus using the menus thus differs with regard to the input/motor requirements (i.e., pressing a key verse pointing with the mouse) as well as memory requirements (i.e., recognizing a command from a list versus recalling a command code from memory). Users thus face several stark changes as they attempt to make the transition from menu-based selection to command language-based selection. In effect, the user must learn an entirely new dialogue.

A second negative characteristic of current systems is that when coding commands these systems maintain brevity of input at the cost of the meaningfulness of the codes. Typically, a single modifier key (e.g., Ctrl or Alt) is paired with a single letter to designate a given command. With the large number of commands found in today's applications, the use of such a strategy means that a limited set of the command codes can be coded meaningfully. Many times the command codes appear to have been arbitrarily generated based on the availability of unpaired keys. Such conditions create a paired associate learning task in which many pairs do

not have an obvious mnemonic. Such tasks have been shown to be extremely difficult (Kling & Riggs, 1971).

Thirdly, it has been shown that the use of consistent organizing principles facilitates the learning of command codes. In contrast, it has been shown that it is extremely difficult to learn flat lists or lists that contain conflicting organizing principles (e.g., Green & Payne, 1984). The command languages used in current graphic user interface (GUI) environments are, for the most part, flat in structure. The majority of the command codes employed in these systems reflect no organizing principle other than the fact that a meaningful letter (i.e., mnemonic) found in the command verb is paired with a modifier key. For a small subset of commands, however, these systems employ a different strategy; namely, the assignment of shortcut keys based on their spatial position on the keyboard. The classic example of this strategy is the use of "X" as the shortcut for "cut," "C" for "copy," and "V" for "paste." It is therefore accurate to say that these systems employ conflicting or inconsistent coding strategies.

As a result of a lack of consistency, continuity, organization, and a shortage of mnemonics, current designs create situations in which it is extremely difficult to learn more than just a few command codes. Nielsen (1987) wrote regarding Apple's graphic user interface environment:

> As more and more full-featured applications take advantage of larger memory and faster CPUs, problems will also arise with the non-hierarchical menu structure and with the use of command-key equivalents for menus. The command-key equivalents will not really be enough for some of tomorrows programs, and the problem can already be seen in Microsoft Word version 3. Overloading functionality on command-keys cause problems as more and more applications get mutually

> inconsistent in their command-key usage. Some other approach to accelerating expert performance is needed. (p. 248)

The obvious question raised by the above mentioned issues is: "What steps can be taken to facilitate maximally the selection of commands from menus, as well as, the bypassing of menus via command codes?" The previously reviewed literature suggests a number of possibilities.

Experimental Rationale

Menu-based command selection. As was mentioned in the previous section, command bar menus are a feature of most graphic user interface environments. To access these menus users either point at the title of the desired menu and press a mouse button or they simultaneously press a modifier key and an underlined letter in the menu title. To complete the command selection sequence with the mouse, users typically drag the cursor over the intended item and release the mouse button. In the second case, users typically select a command by pressing a letter key corresponding to an underlined letter in the desired command. The first scenario will be referred to as "Mouse Menus", the second as "Chorded Menus".

A review of the literature failed to uncover any direct comparisons of the above mentioned techniques. However, as previously mentioned, Karat, McDonald, and Anderson (1986) did find that when their tasks involved typing subjects preferred to use the keyboard rather than the mouse to make menu selections.

A review of the literature also uncovered a second approach to

keyboard-based menu selection that has been shown to have desirable characteristics. Perlman and Sherwin (1988) suggest that matching the physical layout of selector keys to the layout of the menus would produce an interface that would have direct manipulation-like characteristics. They also suggested that such an interface would promote ease of use by being cognitively more intuitive and physically more response compatible. Their research supported their suppositions.

The horizontal row of function keys that runs along the top of most keyboards matches the horizontal layout of the command bar, thus providing a ready made set of selectors for accessing command bar menus. When the menu is displayed, however, the items within it are presented in a vertical list. A glance at the keyboard reveals that there are no vertical rows of keys that can be spatially mapped to vertical menus containing more than three or four items. The question thus becomes, "What type of keyboard selectors would work best for the second half of the command selection sequence?" Given the lack of a spatially mapped alternative, a review of the literature suggests that it would be best to use compatible letter selectors (e.g., the first letter in the desired command). The resulting combination of function key menu access and underlined letter key selectors will be referred to as "Fkey Menus".

Bypass code-based command selection. As previously noted, the bypass code sets used within existing graphic user interface applications are typically ill-organized, inconsistent, and often poor in terms of mnemoticity. As a result, it is extremely difficult to learn more than just a few command codes.

One factor that contributes to the above mentioned problems is that current approaches maintain brevity of input at the cost of consistency and

mnemoticity. A second problem lies in the fact that there is no continuity between the menu-based dialogue that is used to select commands and the bypass code dialogue that is also used to select commands. Finally, these problems are accentuated by the fact that the command languages (i.e., bypass codes) used in current graphic user interface environments are, for the most part, flat in structure. The previously reviewed literature suggests a partial remedy for these problems.

Walker and Olson (1988) developed a hierarchically organized set of command codes that required three keystrokes (i.e., a modifier key and two letter key presses). The first keystroke denoted the broad class of actions from which the user wished to select. The second and third keystrokes denoted the action and object of interest. These authors suggested that benefits that the third keystroke offered in terms of consistency, organizational capabilities, and mnemoticity far outweighed the time costs.

The two keyboard-based approaches to menu selection discussed in the previous section (i.e., Chorded Menus and Fkey Menus) are hierarchically organized. The key sequences used by these two approaches appear to be well organized, consistent, and highly mnemonic. These two coding structures will be referred to as Fkey Codes and Chorded Codes.

The prospect of using these key sequences as bypass codes raises an interesting possibility. Novices could use the above mentioned menu selection techniques to make menu-based command selections. As they became more experienced they could use the same sequences as bypass codes. This strategy would alleviate the discontinuity that normally exists between these two styles of dialogue.

Hypotheses

Menu-based command selection. It was expected that performance times would decrease as experience increased. It was also expected that the three menu selection techniques (i.e., Fkey Menus, Chorded Menus, and Mouse Menus) would differ in terms of the average time it took to access target menus. In contrast, the Fkey Menus and Chorded Menus were not expected to differ in terms of the average time required to select a target command from a displayed menu. However, in terms of the average time required to select a target command from a displayed menu, Fkey Menus and Chorded Menus were expected to differ from Mouse Menus. It was also expected that the three techniques would differ in terms of the average time required to complete an entire menu-based command selection sequence (i.e., the time required to access the target menu plus the time required to select the target command from that menu). It was further expected that the number of selection errors would differ across the three techniques. Finally, it was expected that menu interaction technique would interact with experience and that this interaction would be reflected in the various performance times.

Bypass code-based command selection. It was expected that performance times would decrease as experience increased. It was further expected that the time it took to enter the first portion of the hierarchical bypass code sequence would vary across the two coding structures (i.e., Fkey Codes versus Chorded Codes). In contrast, it was expected that the two bypass coding structures would not vary in terms of the time it took to enter the second portion of the hierarchical bypass code sequence. It was further expected that the two bypass coding structures would differ in terms of the average time required to enter the entire selection sequence.

The average number of selection errors was expected to vary across structure. It was also expected that structure would interact with experience and that this interaction would be reflected in the various performance times.

It was expected that it would be easier to make the transition from menu-based command selection to bypass code-based selection when the same sequences were used for each type of selection. In other words, it was expected that individuals who had experience with Fkey Menus or Chorded Menus would enter the first portion of the bypass codes faster than those who had experience with Mouse Menus. It was further expected that individuals who had experience with Fkey Menus or Chorded Menus would enter the second portion of the bypass codes faster than those who had experience with Mouse Menus. It was also expected that individuals who had experience with Fkey Menus or Chorded Menus would enter the entire bypass code sequence faster than those who had experience with Mouse Menus. It was further expected that the impact of the type of previously used menu selection technique would diminish over time and that this interaction would be reflected in the various performance times. In addition, it was expected that individuals who had experience with Fkey Menus or Chorded Menus would commit fewer errors than those who had experience with Mouse Menus. Finally, it was expected that menu selection technique would interact with bypass coding structure and with experience and that this interaction would be reflected in the various performance times.

METHOD

This study was divided into two phases. In the first phase, the menu selection phase, two naming-based techniques and one pointing-based technique for selecting commands from command bar menus were evaluated. In the second phase, the menu bypass phase, two rule structures used to generate bypass codes for command bar-based menu items were evaluated.

Experimental Design

Menu Selection Phase. The purpose of the menu selection phase of the experiment was to generate design principles that could be used to improve the speed, accuracy, and acceptability of menu-based control techniques. This phase was conducted as a 3 X 6 X 72 mixed-factor design. Menu-based command selection technique was manipulated as a between-subjects variable with the following three levels:

1) Fkey Menus. Subjects in this condition (n=12) popped down menus by depressing function keys that were spatially mapped to the command bar. Command selections were then made by depressing an underlined letter in the target command (see Appendix A).

2) Chorded Menus. Subjects in this condition (n=12) popped down menus by simultaneously depressing a modifier key and the first letter in the menu title. Command selections were then made by depressing an underlined letter in the target command (see Appendix B).

3) Mouse Menus. Subjects in this condition (n=24) accessed pull-down menus by depressing the mouse button while the mouse pointer was over a menu title in the command bar. Command selections were then made by holding down the mouse button, dragging until the target command became highlighted, and releasing (see Appendix C).

Trial blocks and trials were manipulated as within-subjects variables. Each of six task blocks (i.e., proof marked sections of text) contained 72 trials (i.e., command selection tasks) (see Experimental Tasks). The number of trial blocks for the menu selection phase was set to six since pilot data indicated six blocks would be sufficient to demonstrate that combined menu access and command selection times for each of the three selection techniques had reached asymptotic levels.

Both performance and subjective data were collected. For this phase, the principle measures of interest were: 1) Menu access time – time from the last event to precede the selection of a task-specified command to the time the target menu was accessed; 2) Command selection time – time from the moment the target menu was accessed to the time the target command was selected; 3) Combined selection time – time from the last event to precede the selection of a task-specified command to the time the target command was selected; 4) Block completion time – time from the event preceding the first menu selection within a block to the selection of the last specified command within that block; 5) Selection errors – frequency of unspecified command selections and command omissions; 6) Subjective ratings – responses to semantically anchored Likert-style items.

Menu Bypass Phase. The purpose for the menu bypass phase of the experiment was to derive design principles that could be used to ease the transition from menu to code usage and to improve the usability and

memorability of bypass codes. This phase was conducted as a 2 X 2 X 4 X 72 mixed-factor design. Menu input device was manipulated as a between-subjects factor with two levels, namely, keyboard versus mouse. Bypass rule structure was also manipulated as a between-subjects factor with the following two levels:

1) Fkey Codes. Function keys, designating the menu for the target command, served as the first stratum in this hierarchical coding structure. The second stratum consisted of mnemonic letter codes that designated the specific command to be selected from the menu.

2) Chorded Codes. A modifier key plus a mnemonic letter code, designating the menu for the target command, served as the first stratum in this coding structure. The second stratum consisted of mnemonic letter codes that designated the specific command to be selected from the menu.

Crossing menu input device with bypass rule structure created four experimental conditions (12 Ss per condition). These conditions varied in terms of the continuity between the menu selection technique and the bypass selection technique used by the subject, as well as the type of mnemonic used to generate the first portion of the bypass codes (i.e., spatially mapped function keys versus lexically meaningful letter keys).

Trial block and trials were again manipulated as within-subjects variables. The four menu bypass blocks also contained 72 command selection trials each.

For the menu bypass phase of the experiment, the principle measures of interest were: 1) Menu designation time – time from the last event to precede the entry of a task-specified command code to the entry of the menu designating portion of the code; 2) Command designation time – time from the input of the menu designating portion of the code to the entry of the

command designating portion of the code; 3) Combined selection time – time from the last event to precede the entry of a task-specified command code to the entry of the command designating portion of the code; 4) Block completion time – time from the event preceding the first menu selection within a block to the selection of the last specified command within that block; 5) Selection errors – frequency of unspecified command selections and command omissions; 6) Rank-order comparisons – subjective responses to forced-choice items.

Subjects

Forty eight subjects were solicited from a temporary employment agency. Subjects were screened based on their computer-related experience and typing ability. The criteria for eligibility were that the individual: had no experience with a graphical user interface; had little or no word processing experience; had never used a computer mouse; and could type a minimum of 20 words per minute. Subjects were paid $6.50 an hour to participate in the study.

There were 38 females and 10 males ranging in age from 18 to 40. Overall, the subjects had an average of 6.9 months of computer-related experience. The most common form of experience among the subjects was data entry (27 percent of the subjects), followed by word processing (8 percent), programming (4 percent), and other miscellaneous computer experience (2 percent). Twelve subjects were randomly assigned to each of four groups based on the menu bypass conditions.

The subjects in the Mouse Menus/Fkey Code condition included eight females and four males ranging in age from 18 to 31. This group had an average of 13.9 months of computer-related experience. The subjects in the

Mouse Menus/Chorded Code condition included ten females and two males ranging in age from 19 to 39. This group had an average of 6.4 months of computer-related experience. These two groups were combined to create the Mouse Menus group for the menu selection phase of the experiment.

The subjects in the Keyboard Menus/Fkey Code condition included ten females and two males ranging in age from 18 to 40. This group had an average of 2.3 months of computer-related experience. The subjects in the Keyboard Menus/Chorded Code condition included eleven females and one male ranging in age from 18 to 33. This group had an average of 5.0 months of computer-related experience.

Materials

Background questionnaire. The background questionnaire used in this study contained questions regarding the subjects' age, sex, typing speed, occupation, and computer related-experience (see Appendix D).

Tutorial booklet. Permission was granted from Apple™ computer to use materials found in the Macintosh™ guide. These materials were combined with materials developed by the present author to construct a step-by-step set of instructions for each of the tasks that subjects had to perform during the course of the experiment. Captured screen images were used to supplement the written instructions (see Appendix E). Irrespective of the condition to which subjects were assigned, the instructions in the training booklets were identical, except for two pages which described the steps necessary to select a command from a menu.

Trial blocks. Trial blocks were constructed by combining chapters from a publicly available book. These chapters were combined into sections (blocks) ranging in length from 36 to 38 pages (see Appendix F). Proof

reading marks, consistent with the objectives outlined in the Experimental Tasks section, were added to each block of text.

Bipolar rating scales. Subjective evaluations of the menu-based interface were obtained with a sentence completion/Likert-style questionnaire (see Appendix G).

Comparison questionnaire. Rank-order comparisons of menu-based and bypass code-based command selection techniques were obtained with a forced-choice questionnaire (see Appendix H).

Software

Word processing package. A commercially available Macintosh-based word processing package was customized for use in this study. This approach allowed the issues of interest to be investigated within the context of a full and face valid system. The package in question defaults to an eight menu command bar-style interface housing 76 commands, excluding font and point size options. For experimental purposes, those commands not necessary for successful task completion were deactivated. Deactivated items still appeared in the menus, however, they were de-emphasized (i.e., grayed) and were not selectable.

Interface customizing utility. ResEdit™ was used to make the cosmetic changes needed across the various experimental conditions. The capability to underline letters in a menu title or command was made possible by creating an edited version of the Chicago font and substituting it as the system font. Function key and chorded key access to menus, and keyboard access to commands, were made possible by altering mouse and keyboard generated events as they pass through the journaling mechanism found in the Macintosh™ system software.

Keystroke capture utility. A keystroke capture tool, developed in the User Systems Engineering Laboratory at Texas Instruments, was used to collect time stamped records of all user inputs (i.e., mouse and keyboard events). This C program gathered information from a series of Macintosh™-based system level resources and compiled this information into a detailed event log. In addition to raw mouse and keyboard inputs, this log also denoted the active application, all instances where menus were displayed, and all instances where commands were selected.

Hardware

Subjects interacted with an Apple Macintosh™ IIx computer configured with: 4 megabytes of memory, a 150 megabyte hard drive; an extended QWERTY keyboard; a one button mouse, and a 15-inch (diagonal) color monitor.

A standard VHS video camera was used to record all trials. The video camera was focused on the monitor. The resulting tapes were used to clarify the nature of subject errors.

Experimental Tasks

A review of the literature failed to uncover any naturalistic studies of command usage in today's graphic user interface environments. This lack of direct evidence, along with vast differences between these environments and those environments for which direct evidence is available (e.g., line-based text editors), necessitated that such evidence be gathered or that theoretical/indirect evidence be used as a basis for constructing the necessary command selection tasks.

For the present study, the principle emphasis with regard to task construction was to design tasks in which the distribution of command frequency of use typified that found in the “real world”. The decision was made to construct a frequency distribution of command usage based on Zipf's law. Based on an analysis of word usage in spoken and written communications, Zipf (1954) theorized that many classes of human behavior fit a distribution where the most often performed behavior, within a class, is exercised twice as often as the second most frequent behavior, the third most frequent behavior occurs only once for every three occurrences of the most frequent behavior, and so on. In other words, the ratio between the most frequently performed behavior and all other behaviors is such that the most frequent behavior occurs an average of n times for every occurrence of a behavior of the n^{th} rank in the frequency distribution (i.e., the ratios are 2:1, 3:1, 4:1, 5:1, 6:1, 7:1, etc.).

In the present case, the objective was to construct a face valid distribution for a set of 20 core commands found in most graphic user interface-based word processors. To use Zipf's law as the basis for this distribution, it was necessary to first obtain a reliable rank ordering of the commands based on their frequency of use. This ordering was obtained by having experts (N = 9) order a shuffled stack of cards where each of the cards (N = 20) had one of the 20 commands printed on it. The experts were instructed to sort the cards into three stacks representing high, medium, and low usage, to sort the cards in each stack from most to least frequently used, and then to combine the three stacks to obtain a rank ordering for the commands.

The degree of agreement between the nine sets of rankings was tested with Kendall's Coefficient of Concordance, and the resulting coefficient

was found to be statistically significant ($W = 0.64$, $p < .05$). However, as Taylor (1968) noted "...it is not enough to show that some order exists beyond the chance level. Stability between pooled ranks must also be demonstrated." In this case, the estimate of the reliability of the pooled ranks was 0.93.

Once it was determined that the pooled ranks were sufficiently reliable, the overall mean ranks were used as the basis for constructing a Zipf's distribution for command usage frequency. The overall rank order and number of times that a command was repeated within a block are presented in Table 8. These command frequencies were applied to 10 sections of text ranging in length from 36 to 38 pages. In other words, proof reading marks were added to each of the 10 blocks of text indicating where 20 paste operations should be performed, 10 copy operations, 7 cut operations, etc.

Procedure

The temporary services agency, which supplied subjects for this study, gave each subject a typing test just prior to their reporting for the experiment. Subjects brought the results of this test to their first experimental session.

The experiment was carried out across two days in an effort to minimize the effects of fatigue. On the first day, subjects reported at approximately 2:00 o'clock P.M. Subjects were first informed of the nature of the experiment and the tasks that they would be performing. They were then asked to complete a consent form and the background questionnaire.

Subjects were then given the tutorial booklet containing step-by-step instructions on how to perform the experimental tasks using the menu

TABLE 8

Zipf's Distribution of Command Usage Frequencies Per Block

Command	Frequency Per Block
Paste	20
Copy	10
Cut	7
Save	5
Bold	4
Print	3
Close	3
Plain Text	3
Open	2
Quit	2
New	2
Align Center	2
Underline	2
Check Spelling	1
Italic	1
Select All	1
Find	1
Align Left	1
Align Right	1
Justify	1

based style of interaction to which they were assigned (e.g., how to use the mouse, how to select commands from menus, how the commands work, etc.). The training booklet also mentioned the availability of bypass codes and the fact that subjects would latter be instructed on how to use them. It did not, however, provide any details as to how the bypass codes worked. Results from pilot testing suggested that subjects would take between an hour to an hour and a half to complete the training exercises.

Upon completion of the training exercises, subjects were given an overview of the editing marks that appeared in the task materials. Subjects then completed blocks one and two. Following the completion of the second block, subjects were dismissed for the day.

Subjects reported at 8:00 o'clock A.M. on the second day. They then completed two additional trial blocks (i.e., blocks three and four), at which time they were given a five minute break. Subjects then completed blocks five and six, after which they were given the bipolar rating scale designed to assess their opinions regarding the particular menu selection technique they had used in the menu selection phase. The menu selection phase concluded with a fifteen minute break.

The menu bypass phase began with a brief instructional period. During the instructional period the experimenter demonstrated the use of bypass codes. The experimenter then displayed each of the menus for the subject and pointed out the selector key for each of the target commands. Subjects were then instructed to use bypass codes to select commands whenever possible (i.e., whenever they could reliably recall the necessary codes). Subjects were told, however, that whenever they were unsure as to which menu a command was in or what the selector letter was for that command, they should use the menus to select the command. The

experimenter concluded the instructional period by emphasizing to the subjects that their objective was to become familiar with and to use as many bypass codes as possible.

Following the instructional period, subjects completed blocks 7 and 8, took a 5 minute break, and then completed blocks 9 and 10. Following the completion of block 10, subjects completed the comparison questionnaire designed to assess their perceptions regarding menu-based command selection and bypass code-based command selection. Subjects were then debriefed.

RESULTS

Menu Selection Phase

The independent variables investigated in the menu selection phase of the experiment were menu type and trial block. Subjects completed six trial blocks with one of the following menu types: function key menu access with letter key command selection (Fkey Menus); chorded key menu access with letter key command selection (Chorded Menus); or mouse-based menu access with mouse-based command selection (Mouse Menus).

The effects of menu type and block on user performance were assessed in terms of block completion time, menu access time, command selection time, combined menu access and command selection time, and error frequencies. The subjective reactions of users to the respective menu types were assessed through a series of semantically anchored Likert-style items.

Block completion time. Time-stamped keystroke data were used to calculate the time from the task event that preceded the first menu selection in a block to the selection of the last specified command in that block. These times were subsequently analyzed with a 3 X 6 mixed-factor analysis of variance (ANOVA), with the between-groups factor representing menu type and the within-groups factor representing trial block.

Observations also constituted a within-groups factor having 72 levels. However, the degrees of freedom associated with this factor were so large that the main effect and interaction effects involving it were trivial tests. Therefore, F-tests were not reported for these portions of the model.

The main effect of menu type was found to be significant ($F(2,45) = 11.63$, $p < .01$; Table 9, Figure 1). Newman-Keuls tests showed that, on the average, the Fkey Menus group completed the task blocks faster (31.10 min) than did the Mouse Menus or the Chorded Menus groups (39.50 min and 40.33 min, respectively; Table 10).

The main effect of block was also significant ($F(5,225) = 239.571$, $p < 0.01$; Table 9, Figure 2). Newman-Keuls tests showed that, on the average, subjects took more time to complete Block one (64.08 min) than they did any other block. On the average, Blocks two and three took longer to complete (36.98 and 38.52 min, respectively) than did Blocks four, five, and six (30.47, 27.20, and 28.39 min, respectively). The completion times for Blocks four and six did not differ, whereas, Blocks four and five did (Table 11).

Menu access time. Keystroke data were used to calculate the time between the last user input to precede the search for and selection of a target menu (e.g., a mouse click or keypresses in the content region of the window) to the time the target menu was requested (i.e., the user initiated the input that triggered the display of the menu). These times were subsequently analyzed with a 3 X 6 mixed-factor ANOVA, with the between-groups factor representing menu type and the within-groups factor representing block.

The main effect of menu type was significant ($F(2,45) = 12.20$, $p < 0.01$; Table 12, Figure 3). Newman-Keuls tests showed that, on the average, the Fkey Menus group accessed the target menus faster (3.13 s) than did the Mouse Menus or the Chorded Menus groups (4.05 s and 4.65 s, respectively; Table 13).

The main effect of block was also significant ($F(5,225) = 269.29$, $p < 0.01$; Table 12, Figure 4). Subjects took longer, on the average, to access menus

TABLE 9

ANOVA Summary Table for Block Completion Times

SOURCE	MS	df	F
MENU TYPE	2048.511	2	11.634 **
SUBJECT(MENU TYPE)	176.079	45	.
BLOCK	9084.879	5	239.571 **
BLOCK*MENU TYPE	17.712	10	0.467
BLOCK*SUBJECT(MENU TYPE)	37.921	225	.

* $p < 0.05$ ** $p < 0.01$

TABLE 10

Newman-Keuls Tests on Block Completion Times by Menu Type

Menu Type	Mean
Chorded Menus	40.33 (A)
Mouse Menus	39.50 (A)
Fkey Menus	31.10 (B)

NOTE: Means sharing a common letter in parentheses were not significantly different ($p < 0.05$). Times given in minutes.

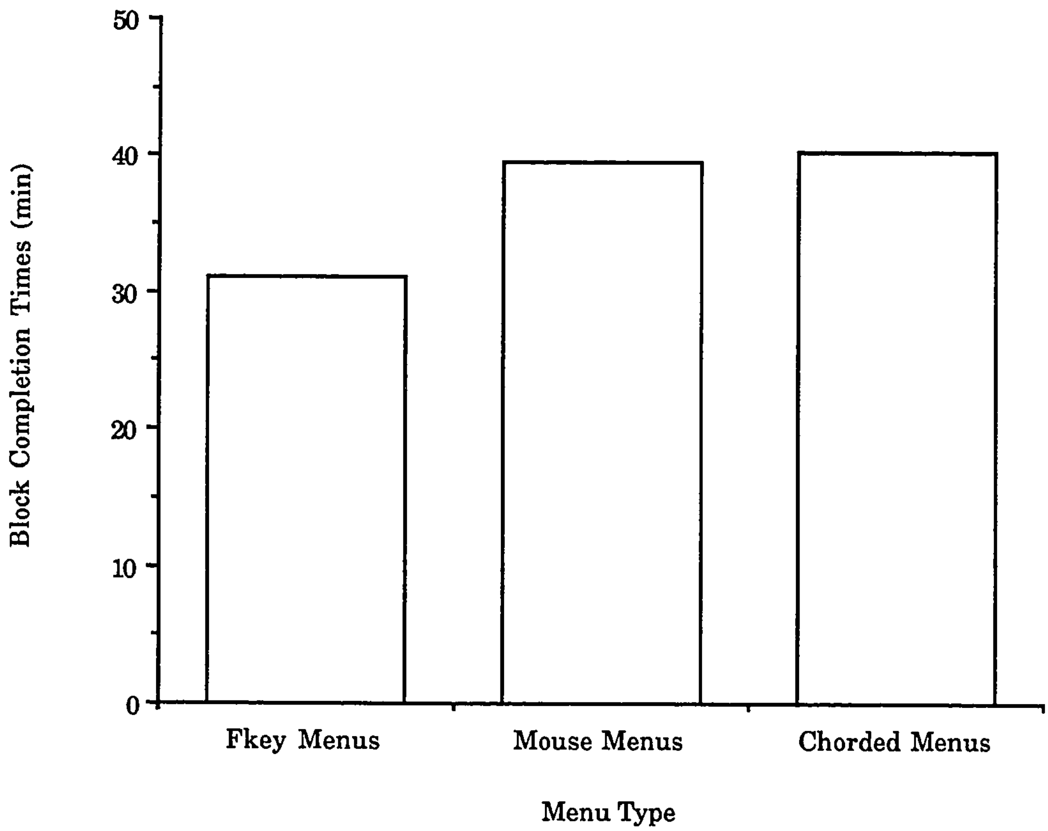

Figure 1. Block completion time by menu type.

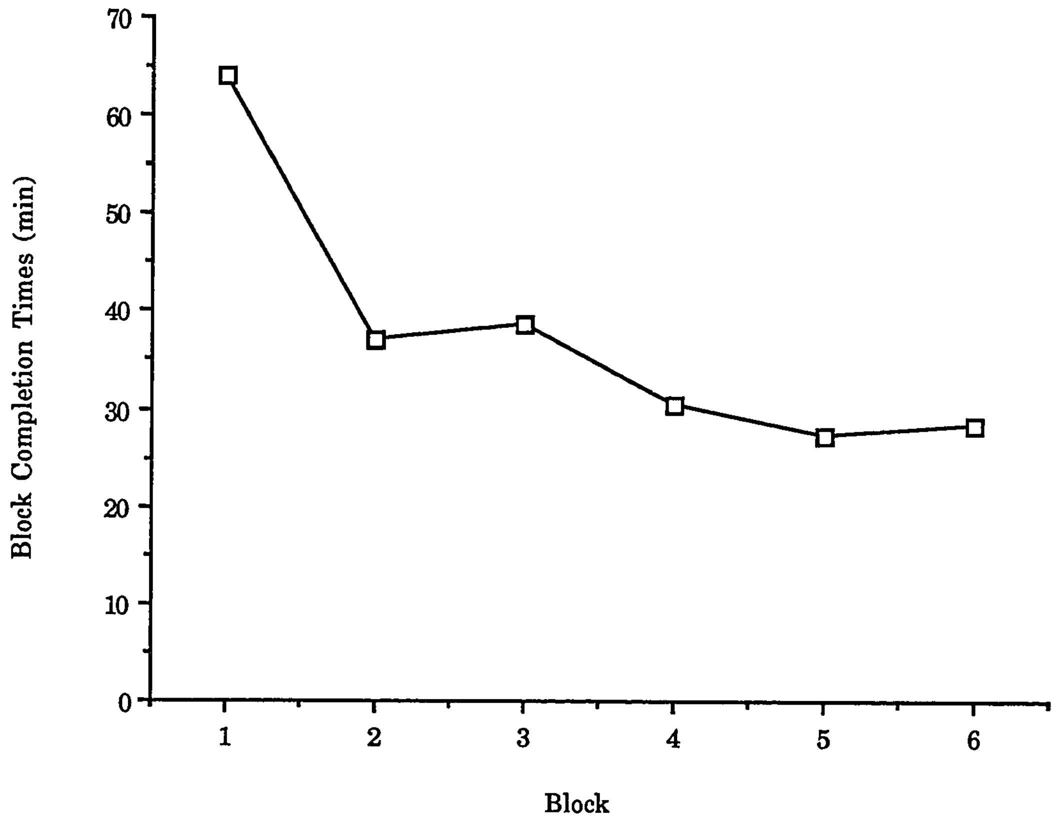

Figure 2. Block completion times by block.

TABLE 11

Newman-Keuls Tests on Block Completion Times by Block

Block	Mean
1	64.08 (A)
2	36.98 (B)
3	38.52 (B)
4	30.47 (C)
6	28.39 (C) (D)
5	27.20 (D)

NOTE: Means sharing a common letter in parentheses were not significantly different ($p < 0.05$). Times given in minutes.

TABLE 12

ANOVA Summary Table for Menu Access Times

SOURCE	MS	df	F
MENU TYPE	3039.246	2	12.20 **
SUBJECT(MENU TYPE)	249.059	45	.
BLOCK	8633.015	5	269.29 **
BLOCK*MENU TYPE	280.024	10	8.73 **
BLOCK*SUBJ(MENU TYPE)	32.056	225	.
OBSERVATION	300.285	71	.
OBSERV*MENU TYPE	39.982	142	.
OBSERV*SUBJ(MENU TYPE)	18.661	3195	.
OBSERV*BLOCK	341.995	355	.
OBSERV*BLOCK*MENU TYPE	37.345	710	.
OBSERV*BLOCK*SUBJ(TYPE)	17.183	15566	.

* $p < 0.05$ ** $p < 0.01$

TABLE 13

Newman-Keuls Tests on Menu Access Time by Menu Type

Menu type	Mean
Chorded Menus	4.65 (A)
Mouse Menus	4.05 (A)
Fkey Menus	3.13 (B)

NOTE: Means sharing a common letter in parentheses were not significantly different ($p < 0.05$). Times given in seconds.

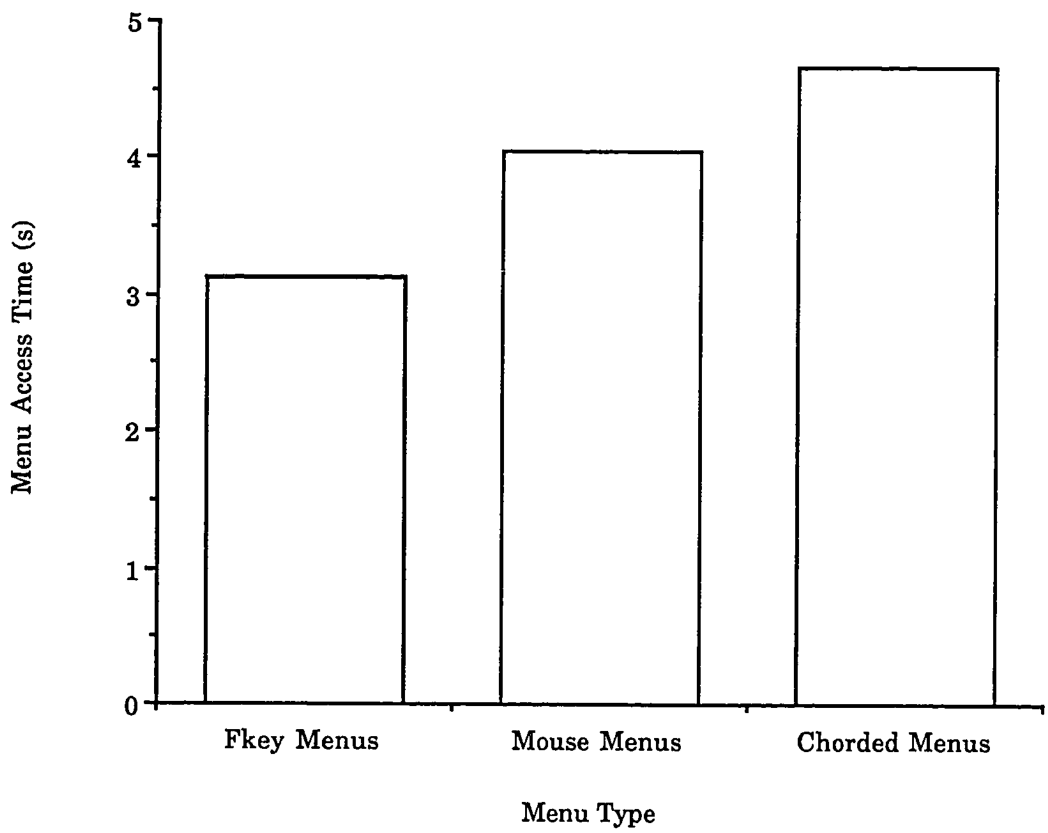

Figure 3. Menu access time by menu type.

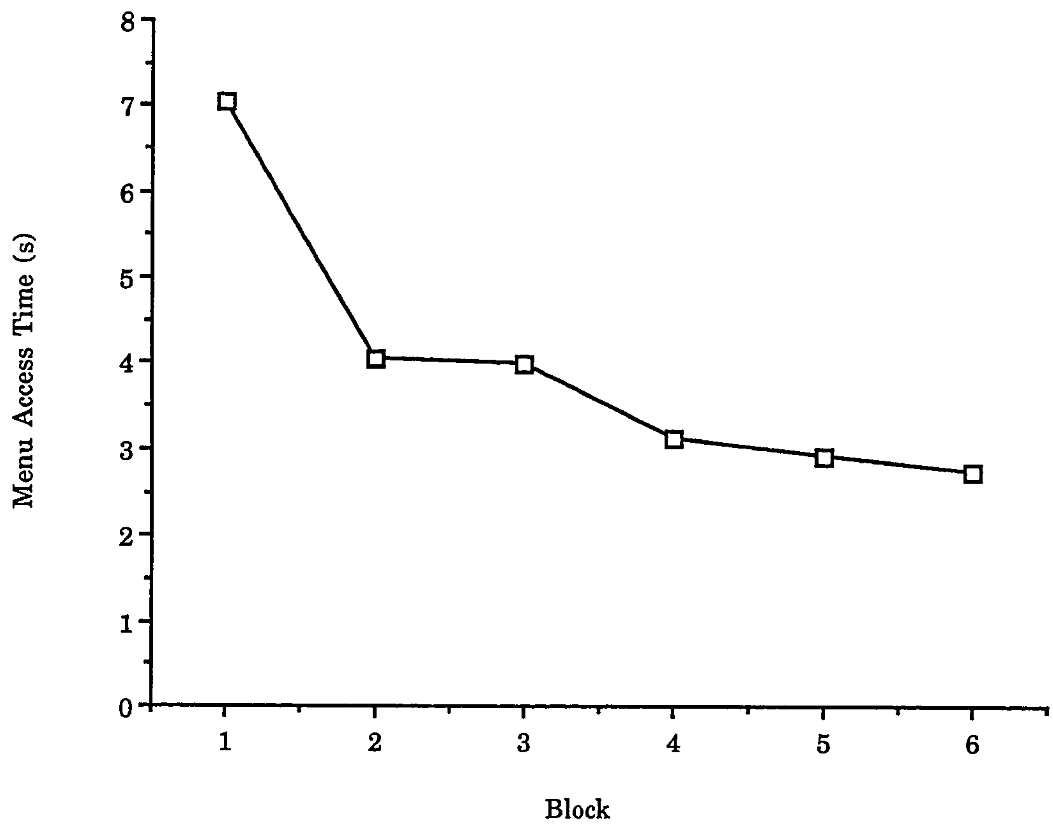

Figure 4. Menu access time by block.

during Block one (7.04 s) than they did during any other block. They also took longer, on the average, to access menus during Blocks two (4.05 s) and three (3.99 s) than they did during Blocks four, five, and six (3.12 s, 2.91 s, and 2.74 s, respectively). The means for Blocks four and six differed but the means for Blocks four and five, and Blocks five and six did not (Table 14).

Finally, there was a significant interaction between menu type and block ($F(10, 225) = 8.73$, $p < 0.01$; Table 12, Figure 5). Simple-effects tests revealed that the simple main effect of menu type was significant at all six blocks (Table 15). Newman-Keuls tests showed that for Block one the mean menu access time for the Chorded Menus group was slower (9.08 s) than that of the other two groups, while the mean menu access time for the Mouse Menus group and the Fkey Menus group did not differ (6.55 s and 5.97 s respectively; Table 16). In contrast, over the remaining blocks it was found that the mean menu access times for the FKey Menus group were faster (3.21, 3.09, 2.36, 2.17, 1.98 s, for the respective blocks) than those of the Mouse Menus group (4.19, 4.23, 3.37, 3.13, and 2.95 s, for the respective blocks) or the Chorded Menu group (4.80, 4.42, 3.38, 3.19, and 3.07, for the respective blocks; Table 16).

Command selection time. Keystroke data were used to calculate the time from the moment the target menu was accessed to the time the target command was selected (i.e., the user initiated the input that triggered the selection of the command). These times were subsequently analyzed with a 3 X 6 mixed-factor ANOVA, with the between-groups factor representing menu type and the within-groups factor representing block.

The main effect of menu type was significant ($F(2,45) = 9.17$, $p < 0.01$; Table 17, Figure 6). Newman-Keuls tests showed that, on the average, the command selection times for the Fkey Menus group and the Chorded

TABLE 14

Newman-Keuls Tests on Menu Access Times by Block

Block	Mean
1	7.04 (A)
2	4.05 (B)
3	3.99 (B)
4	3.12 (C)
5	2.91 (C) (D)
6	2.74 (D)

NOTE: Means sharing a common letter in parentheses were not significantly different ($p < 0.05$). Times given in seconds.

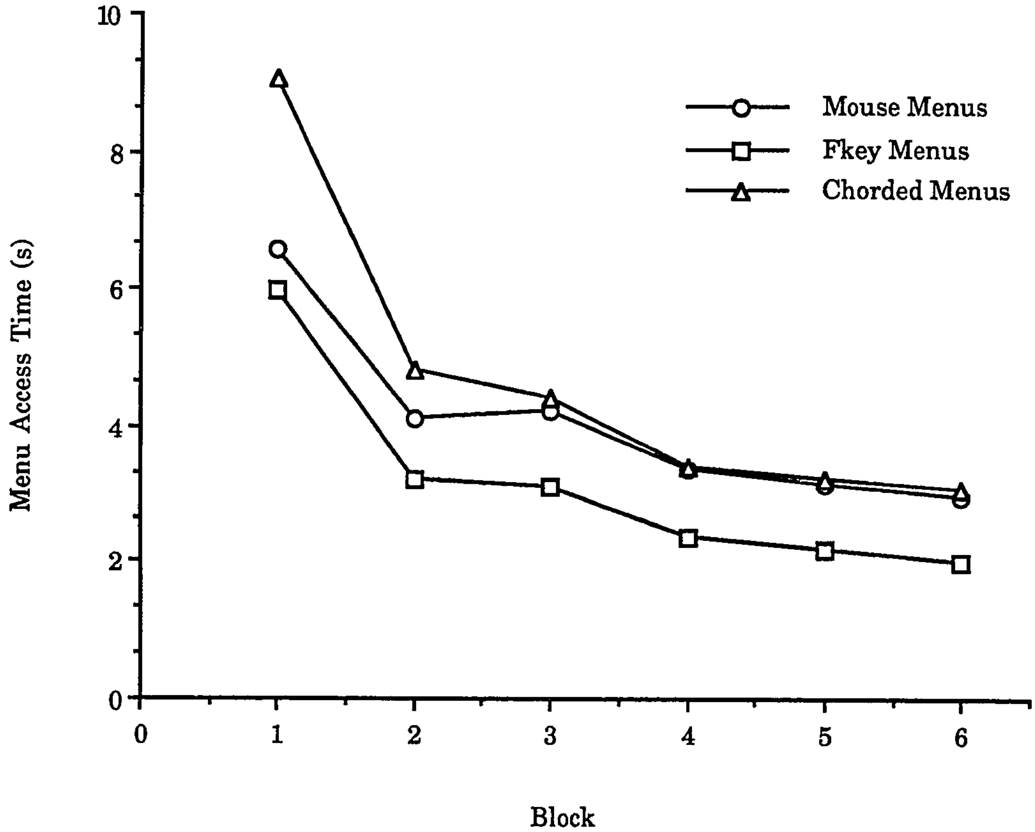

Figure 5. Menu access times for menu type by block interaction.

TABLE 15

Summary Table for Menu Access Time Simple-Effects F-Tests of Menu Type by Block

SOURCE	MS	df	F
MENU TYPE at BLOCK 1	2448.89	2	11.48 **
SUBJ(MENU TYPE) at BLOCK 1	213.36	45	.
MENU TYPE at BLOCK 2	544.53	2	8.60 **
SUBJ(MENU TYPE) at BLOCK 2	63.35	45	.
MENU TYPE at BLOCK 3	474.55	2	8.43 **
SUBJ(MENU TYPE) at BLOCK 3	56.32	45	.
MENU TYPE at BLOCK 4	331.52	2	12.14 **
SUBJ(MENU TYPE) at BLOCK 4	27.30	45	.
MENU TYPE at BLOCK 5	309.49	2	15.06 **
SUBJ(MENU TYPE) at BLOCK 5	20.54	45	.
MENU TYPE at BLOCK 6	330.39	2	11.06 **
SUBJ(MENU TYPE) at BLOCK 6	28.48	45	.

* $p < 0.05$ ** $p < 0.01$

TABLE 16

Newman-Keuls Tests on Menu Access Times for Menu Type at Blocks One Through Six

Menu Type	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6
Chrd. Menus	9.08 (A)	4.80 (A)	4.42 (A)	3.38 (A)	3.19 (A)	3.07 (A)
Mouse Menus	6.65 (B)	4.19 (A)	4.23 (A)	3.37 (A)	3.13 (A)	2.95 (A)
Fkey Menus	5.97 (B)	3.21 (B)	3.09 (B)	2.36 (B)	2.17 (B)	1.98 (B)

NOTE: Means sharing a common letter in parentheses were not significantly different ($p < 0.05$). Times given in seconds.

TABLE 17

ANOVA Summary Table for Command Selection Times

SOURCE	MS	df	F
MENU TYPE	558.65	2	9.17 **
SUBJECT(MENU TYPE)	60.95	45	.
BLOCK	792.69	5	91.60 **
BLOCK*MENU TYPE	25.04	10	2.89 **
BLOCK*SUBJ(MENU TYPE)	8.65	225	.
OBSERVATION	17.36	71	.
OBSERV*MENU TYPE	8.18	142	.
OBSERV*SUBJ(MENU TYPE)	4.83	3195	.
OBSERV*BLOCK	17.29	355	.
OBSERV*BLOCK*MENU TYPE	10.43	710	.
OBSERV*BLOCK*SUBJ(TYPE)	4.761	15566	.

* $p < 0.05$ ** $p < 0.01$

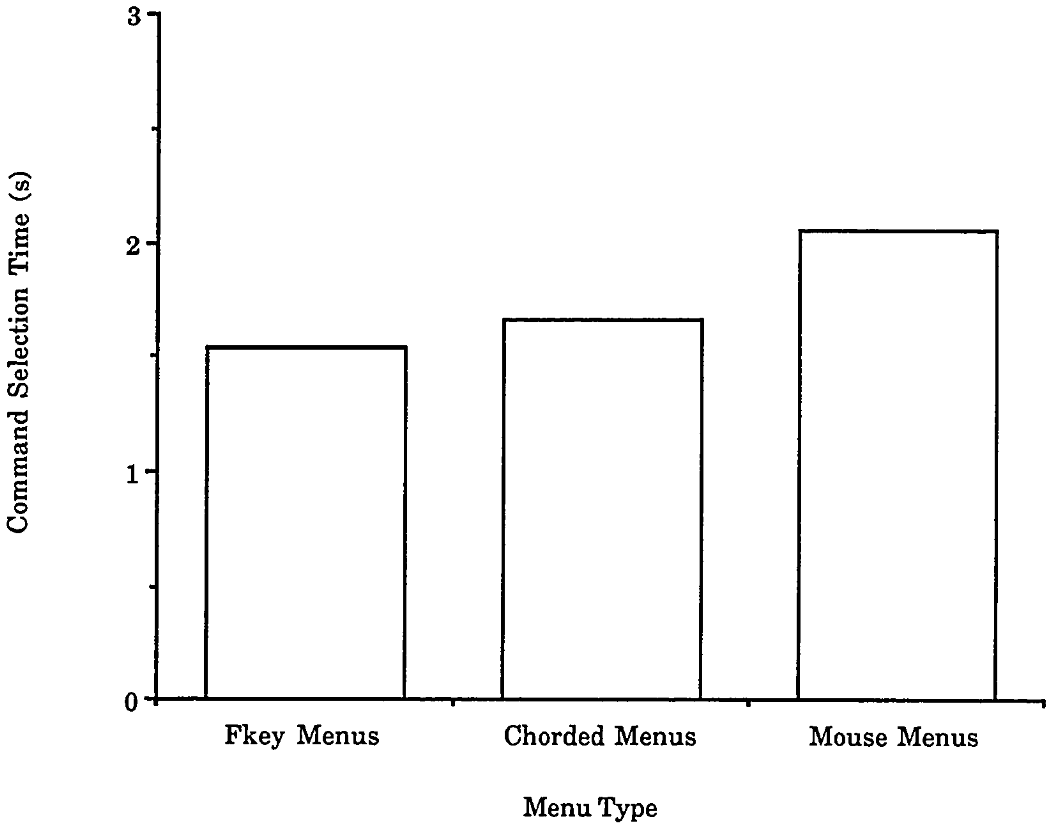

Figure 6. Command selection time by menu type.

Menus group were faster (1.54 s and 1.66 s, respectively) than those of the Mouse Menus group (2.06 s; Table 18).

The main effect of block was also significant ($F(5,225) = 91.60$, $p < 0.01$; Table 17, Figure 7). Newman-Keuls test showed that the mean command selection time for Block one (2.69 s) was slower than that of the blocks that followed. For Block two the mean command selection time was also slower (2.00 s) than that of the remaining blocks. Similarly, the average command selection time for Block three (1.86 s) was slower than that of Blocks four, five, and six (1.56 s, 1.45 s, and 1.39 s, respectively). Tests of the differences between these latter blocks revealed that the mean command selection time for Blocks four and six were different, but that the times for Blocks four and five, and five and six were not (Table 19).

The interaction between menu type and block was also significant ($F(10, 225) = 8.73$, $p < 0.01$; Table 17, Figure 8). Simple effects tests revealed a simple main effect of menu type at Blocks two through six (Table 20). Newman-Keuls tests conducted at Block two showed no differences between the three menu types (Table 21). However, Newman-Keuls tests conducted at Blocks three through six indicated that the mean command selection times for the Fkey Menus group (1.62, 1.22, 1.12, and 0.98 s, for the respective blocks) and Chorded Menus group (1.64, 1.31, 1.23, and 1.11 s, for the respective blocks) were faster than those of the Mouse Menus group (2.09, 1.86, 1.77, and 1.73 s, for the respective blocks; Table 21).

<u>Combined menu access and command selection time</u>. Keystroke data were used to calculate the time between the last user input to precede the search for and selection of a target menu to the time the target command was selected. These times were subsequently analyzed with a 3 X 6 mixed-factor ANOVA, with the between-groups factor representing menu type

TABLE 18

Newman-Keuls Tests on Command Selection Times by Menu Type

Menu Type	Mean
Chorded Menus	2.06 (A)
Mouse Menus	1.66 (B)
Fkey Menus	1.54 (B)

NOTE: Means sharing a common letter in parentheses were not significantly different ($p < 0.05$). Times given in seconds.

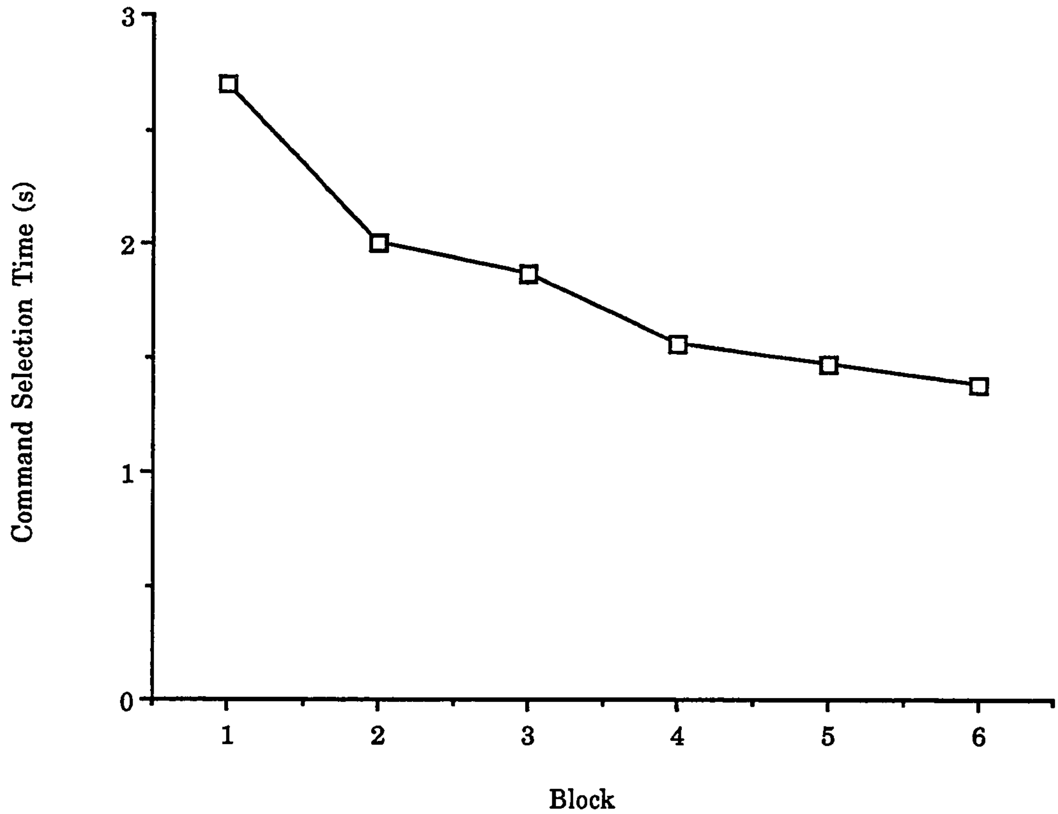

Figure 7. Command selection times by block.

TABLE 19

Newman-Keuls Tests on Command Selection Times by Block

Block	Mean
1	2.69 (A)
2	2.00 (B)
3	1.86 (C)
4	1.56 (D)
5	1.45 (D) (E)
6	1.39 (E)

NOTE: Means sharing a common letter in parentheses were not significantly different ($p < 0.05$). Times given in seconds.

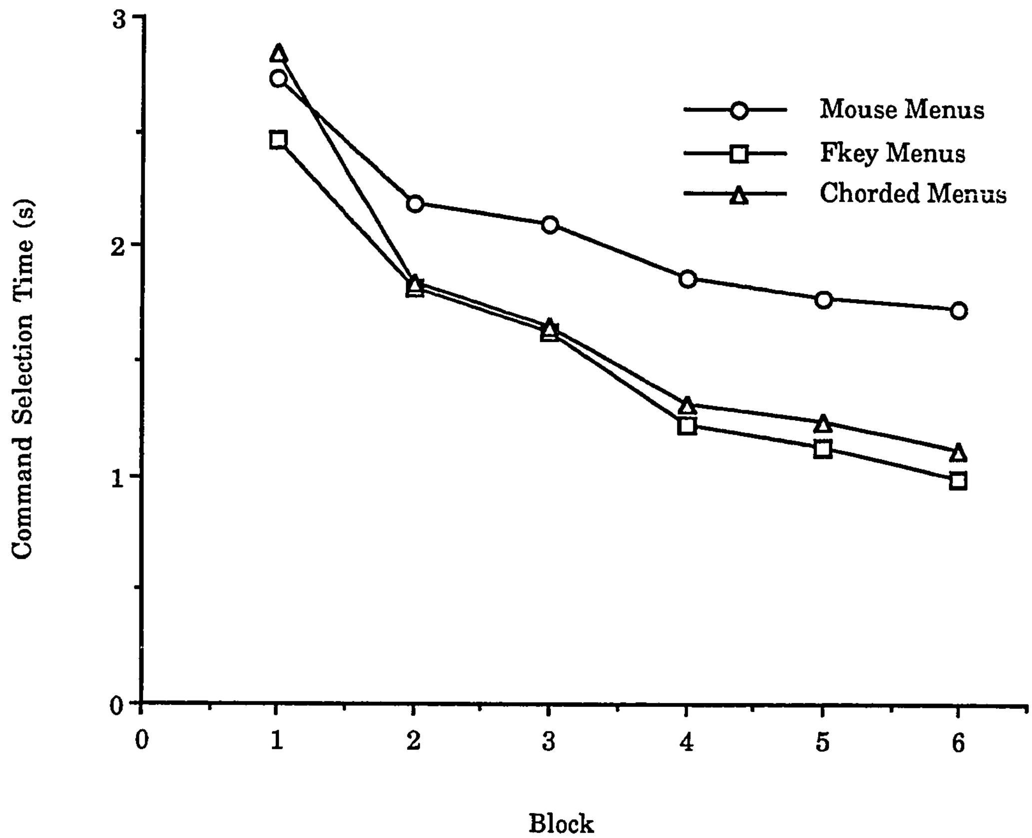

Figure 8. Command selection times for menu type by block interaction.

TABLE 20

Summary Table for Command Selection Time Simple-Effects F-Tests of Menu Type by Block

SOURCE	MS	df	F
MENU TYPE at BLOCK 1	32.76	2	0.57
SUBJ(MENU TYPE) at BLOCK 1	57.86	45	.
MENU TYPE at BLOCK 2	56.71	2	4.42 *
SUBJ(MENU TYPE) at BLOCK 2	12.82	45	.
MENU TYPE at BLOCK 3	89.77	2	8.40 **
SUBJ(MENU TYPE) at BLOCK 3	10.69	45	.
MENU TYPE at BLOCK 4	150.86	2	16.29 **
SUBJ(MENU TYPE) at BLOCK 4	9.26	45	.
MENU TYPE at BLOCK 5	152.76	2	23.32 **
SUBJ(MENU TYPE) at BLOCK 5	6.55	45	.
MENU TYPE at BLOCK 6	201.01	2	28.59 **
SUBJ(MENU TYPE) at BLOCK 6	7.03	45	.

* $p < 0.05$ ** $p < 0.01$

TABLE 21

Newman-Keuls Tests on Command Selection Times for Menu Type at Blocks One Through Six

Menu Type	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6
Chrd. Menus	2.85 (A)	2.19 (A)	2.09 (A)	1.86 (A)	1.77 (A)	1.73 (A)
Mouse Menus	2.73 (A)	1.83 (B)	1.64 (B)	1.31 (B)	1.23 (B)	1.11 (B)
Fkey Menus	2.47 (A)	1.81 (B)	1.62 (B)	1.22 (B)	1.12 (B)	0.98 (B)

NOTE: Within a given column means sharing a common letter in parentheses were not significantly different ($p < 0.05$). Times given in seconds.

and the within-groups factor representing block.

The main effect of menu type was significant ($F(2,45) = 9.83$, $p < 0.01$; Table 22, Figure 9). Newman-Keuls tests showed that, on the average, the combined access and selection times were shorter for the Fkey Menus group (4.66 s) than they were for the Mouse Menus or the Chorded Menus groups (6.11 s and 6.32 s, respectively; Table 23).

The main effect of block was also significant ($F(5,225) = 295.22$, $p < 0.01$; Table 22, Figure 10). Subjects took longer, on the average, to access and select commands during Block one (9.73 s) than they did during the other blocks. They also took longer, on the average, to access and select commands during Blocks two (6.06 s) and three (5.85 s) than they did during Blocks four, five, and six (4.68 s, 4.38 s, and 4.13 s, respectively). Blocks four and six differed, but Blocks four and five, and Blocks five and six did not (Table 24).

Finally, there was a significant interaction between menu type and block ($F(10, 225) = 9.00$, $p < 0.01$; Table 22, Figure 11). Simple effects tests revealed that the simple main effect of menu type was significant at all six blocks (Table 25). Newman-Keuls tests showed that, on the average, it took the Chorded Menus group longer to access and select commands in Block one (11.93 s) than it did for the other two groups. These tests also showed that the Fkey Menus and the Mouse Menus groups did not differ in terms of the average time required to access and select a command in Block one (9.28 s and 8.44 s respectively; Table 26). In contrast, over the remaining blocks the Fkey Menus group took less time, on the average, to access and select target commands (5.02, 4.71, 3.58, 3.28, and 2.96 s, for the respective blocks) than did the Mouse Menus (6.29, 6.33, 5.23, 4.90, and 4.68 s, for the respective blocks) or the Chorded Menus groups (6.63, 6.06, 4.69, 4.42, and

TABLE 22

ANOVA Summary Table for Combined Access and Selection Times

SOURCE	MS	df	F
MENU TYPE	4466.59	2	9.83 **
SUBJECT(MENU TYPE)	453.60	45	.
BLOCK	14583.52	5	295.22 **
BLOCK*MENU TYPE	444.47	10	9.00 **
BLOCK*SUBJ(MENU TYPE)	49.40	225	.
OBSERVATION	403.30	71	.
OBSERV*MENU TYPE	57.66	142	.
OBSERV*SUBJ(MENU TYPE)	24.16	3195	.
OBSERV*BLOCK	462.68	355	.
OBSERV*BLOCK*MENU TYPE	57.27	710	.
OBSERV*BLOCK*SUBJ(TYPE)	22.58	15566	.

* $p < 0.05$ ** $p < 0.01$

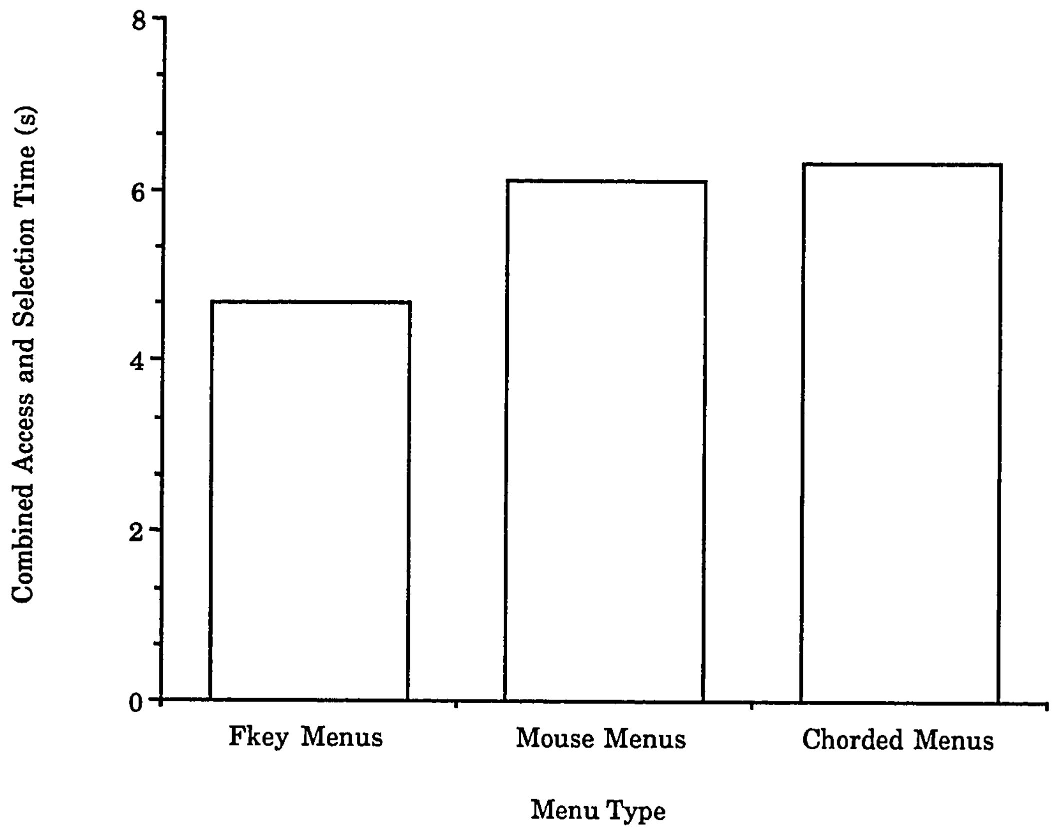

Figure 9. Combined access and selection time by menu type.

TABLE 23

Newman-Keuls Tests on Combined Access and Selection Times for Menu Type

Menu Type	Mean
Chorded Menus	6.32 (A)
Mouse Menus	6.11 (A)
Fkey Menus	4.66 (B)

NOTE: Means sharing a common letter in parentheses were not significantly different ($p < 0.05$). Times given in seconds.

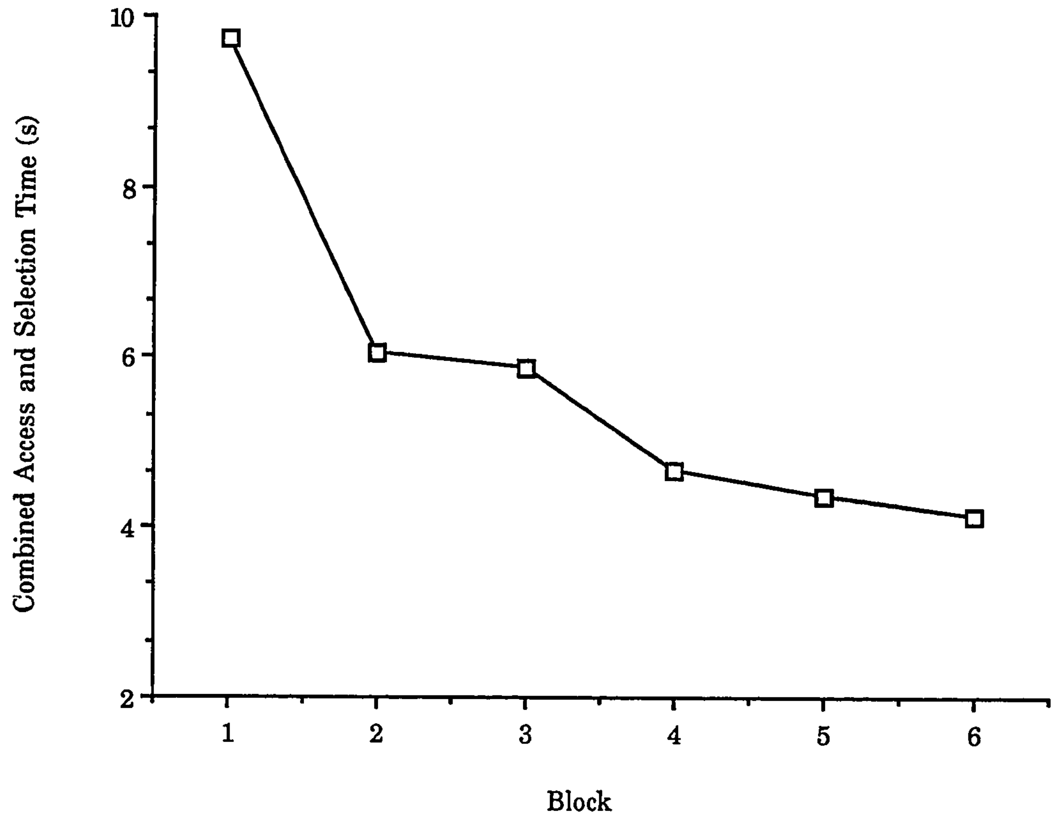

Figure 10. Combined access and selection time by block.

TABLE 24

Newman-Keuls Tests on Command Selection Times by Block

Block	Mean
1	9.73 (A)
2	6.06 (B)
3	5.85 (B)
4	4.68 (C)
5	4.38 (C) (D)
6	4.13 (D)

NOTE: Means sharing a common letter in parentheses were not significantly different ($p < 0.05$). Times given in seconds.

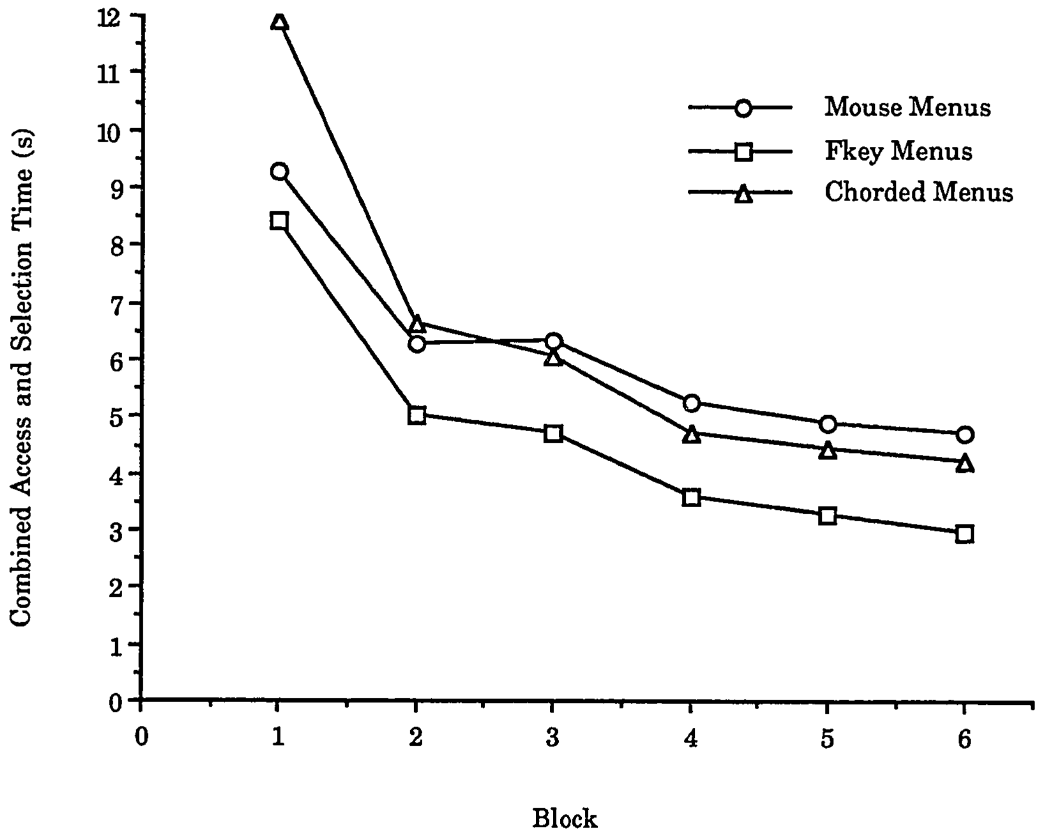

Figure 11. Combined access and selection times for menu type by block interaction.

TABLE 25

Summary Table for Command Selection Time Simple-Effects F-Tests of Menu Type by Block

SOURCE	MS	df	F
MENU TYPE at BLOCK 1	2919.53	2	8.00 **
SUBJ(MENU TYPE) at BLOCK 1	364.97	45	.
MENU TYPE at BLOCK 2	649.73	2	6.36 **
SUBJ(MENU TYPE) at BLOCK 2	102.10	45	.
MENU TYPE at BLOCK 3	763.96	2	8.04 **
SUBJ(MENU TYPE) at BLOCK 3	95.03	45	.
MENU TYPE at BLOCK 4	776.75	2	14.02 **
SUBJ(MENU TYPE) at BLOCK 4	55.40	45	.
MENU TYPE at BLOCK 5	741.01	2	21.43 **
SUBJ(MENU TYPE) at BLOCK 5	34.58	45	.
MENU TYPE at BLOCK 6	837.93	2	16.42 **
SUBJ(MENU TYPE) at BLOCK 6	49.52	45	.

* $p < 0.05$ ** $p < 0.01$

TABLE 26

Newman-Keuls Tests on Combined Access and Selection Times for Menu Type at Block One Through Six

Menu Type	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6
Chrd. Menus	11.93 (A)	6.63 (A)	6.06 (A)	4.69 (A)	4.42 (A)	4.19 (A)
Mouse Menus	9.28 (B)	6.29 (A)	6.33 (A)	5.23 (A)	4.90 (A)	4.68 (A)
Fkey Menus	8.44 (B)	5.02 (B)	4.71 (B)	3.58 (B)	3.28 (B)	2.96 (B)

NOTE: Within a given column means sharing a common letter in parentheses were not significantly different ($p < 0.05$). Times given in seconds.

4.19 s, for the respective blocks; Table 26).

Errors. Video tape and keystroke recordings were used to identify and tally three types of errors. Errors were recorded if subjects: selected commands not specified in the task materials (Extra Actions); substituted an unspecified command for a task-specified command (Substitutions); or omitted a task-specified command (Omissions). Errors were cross-tabulated by type of error and by menu type. The resulting frequency counts were analyzed through a series of Chi-Square tests.

A Chi-Square test of independence was performed on the menu type by error type contingency table. This test indicated that menu type and error type were associated (Chi-Square = 20.598, df = 4, $p < 0.05$).

A series of Chi-Square goodness of fit tests were conducted to determine if the three types of errors were evenly distributed across the three menu types. The Extra Actions errors were found to be uniformly distributed (Chi-Square = 3.203, df = 4, $p > 0.05$). In contrast, the Substitution and Omission errors were not evenly distributed (Chi-Square = 11.771, df = 4, $p < 0.05$ and Chi-Square = 25.096, df = 4, $p < 0.05$, respectively). A comparison of the observed and expected frequencies revealed that the number of Substitution errors committed by the Fkey Menus group was far less than expected (cell contribution to Chi-Square = 8.53) while the number of Omission errors committed by the Mouse Menus group was far more than expected (cell contribution to Chi-Square = 12.33; Tables 27).

A goodness of fit test was also performed on the total number of errors committed with each menu type. This test indicated that the errors were not uniformly distributed (Chi-Square = 21.130, df = 4, $p < 0.05$). More errors than expected were committed by the Mouse Menus group (cell

TABLE 27

Cross-tabulation of Errors by Error Type and by Menu Type

	MENU TYPE			
ERROR TYPE	MOUSE	CHORDED	FKEYS	TOTAL
EXTRA ACTIONS	327	143	140	610
SUBSTITUTIONS	62	35	12	109
OMISSIONS	189	48	56	293
TOTAL	578 n = 24	226 n = 12	208 n =12	1012 N = 48

contribution to Chi-Square = 10.24), and fewer errors than expected were committed by the Fkey Menus group (cell contribution to Chi-Square = 8.00).

Subjective ratings. The rating scale data were analyzed with Kruskal-Wallis One-Way Analysis of Variance by Ranks tests with menu type as the independent variable. A rank-order difference was found for two of the twelve items (Table 28). Mean rank comparison tests showed that, relative to the Chorded Menus group, the Fkey Menus group felt that it was easier to remember which menu a command was in ($|\text{Mean Rank}_{fkey} - \text{Mean Rank}_{chrd}| = 12.750$, $p < 0.05$). Mean rank comparison tests also showed that, relative to the Fkey Menus group, the Mouse Menus group found the command selection process to be more invigorating ($|\text{Mean Rank}_{mse} - \text{Mean Rank}_{fkey}| = 11.708$, $p < 0.05$).

Menu Bypass Phase

Menu input device, bypass coding structure, and trial blocks served as the independent variables for the menu bypass phase of the experiment. Subjects were categorized according to the menu input device they had used during the menu selection phase. The Fkey Menus and Chorded Menus groups were categorized as Keyboard Menus groups. The Mouse Menus group retained its earlier classification (Mouse Menus).

Subjects used the same device to interact with menus that they had used in the menu selection phase of the experiment. However, subjects were instructed to avoid selecting commands from menus whenever possible. Instead, subjects were encouraged to use bypass codes to select commands. Subjects used either: function key/letter key codes (Fkey

TABLE 28

Kruskal-Wallis Analyses of Variance by Menu Interaction Technique For Rating Scale Dimensions

	Mean Ranks			
Activity/Dimen.	Mouse	Chorded	Function	Chi-Square
learning/ease	27.00	18.08	25.92	4.121
remembering/ease	26.25	16.37	29.12	6.299 *
high use select/speed	22.44	23.17	29.96	2.610
low use select/speed	25.62	17.25	29.50	5.178
select/ease	25.96	17.92	28.29	4.622
select/convenience	25.48	22.25	24.79	0.512
use/vigor	29.04	25.58	17.33	6.212 *
use/comfort	24.92	23.17	25.00	0.187
use/good	24.90	23.67	24.54	0.078
use/speed	25.27	23.83	23.62	0.174
use/friendliness	24.06	20.50	29.37	3.000

* $p < 0.05$

Codes); or chorded key/letter key codes (Chorded Codes). Subjects completed four blocks of menu bypass trials.

User performance was assessed in terms of: block completion time; menu designation time; command designation time; combined menu and command designation time; and error frequencies. A series of forced-choice items were also administered, after completion of the final trial block, to assess user perceptions regarding menu-based versus bypass code-based command selection.

Block completion time. Time-stamped keystroke data were used to calculate the time from the task event that preceded the first menu selection in a block to the selection of the last specified command in that block. These times were subsequently analyzed with a 2 X 2 X 4 mixed-factor analysis of variance (ANOVA), with the between-groups factors representing menu input device and bypass coding structure, and the within-groups factor representing block.

The main effect of menu input device was significant ($F(1,44) = 6.76$, $p < 0.01$; Table 29, Figure 12), as was bypass structure ($F(1,44) = 7.82$, $p < 0.01$; Table 29, Figure 13), and block ($F(3,132) = 113.58$, $p < 0.01$; Table 29, Figure 14). The block means indicated that the Keyboard Menus groups finished the trial blocks faster (26.64 min) than the Mouse Menus groups (30.38 min). The means also showed that the groups that used Fkey Codes finished the trial blocks faster (26.50 min) than those that used Chorded Codes (30.52 min). Finally, Newman-Keuls tests showed that, on the average, subjects took longer to complete Block seven (36.85 min) than they did Blocks eight, nine, and ten (26.56, 25.30, and 25.32 min, respectively; Table 30).

TABLE 29

ANOVA Summary Table for Block Completion Times

SOURCE	MS	df	F
MENU INPUT DEVICE	668.764	1	6.76 *
BYPASS CODING STRUCTURE	772.941	1	7.82 **
DEVICE*BLOCK	308.222	1	3.12
SUBJ(DEVICE*BYPASS)	98.873	44	.
BLOCK	1501.431	3	113.58 **
BLOCK*DEVICE	106.933	3	8.09 **
BLOCK*BYPASS	33.722	3	2.55
BLOCK*DEVICE*BYPASS	5.050	3	0.38
BLOCK*SUBJ(DEVICE*BYPASS)	13.219	132	.

* $p < 0.05$ ** $p < 0.01$

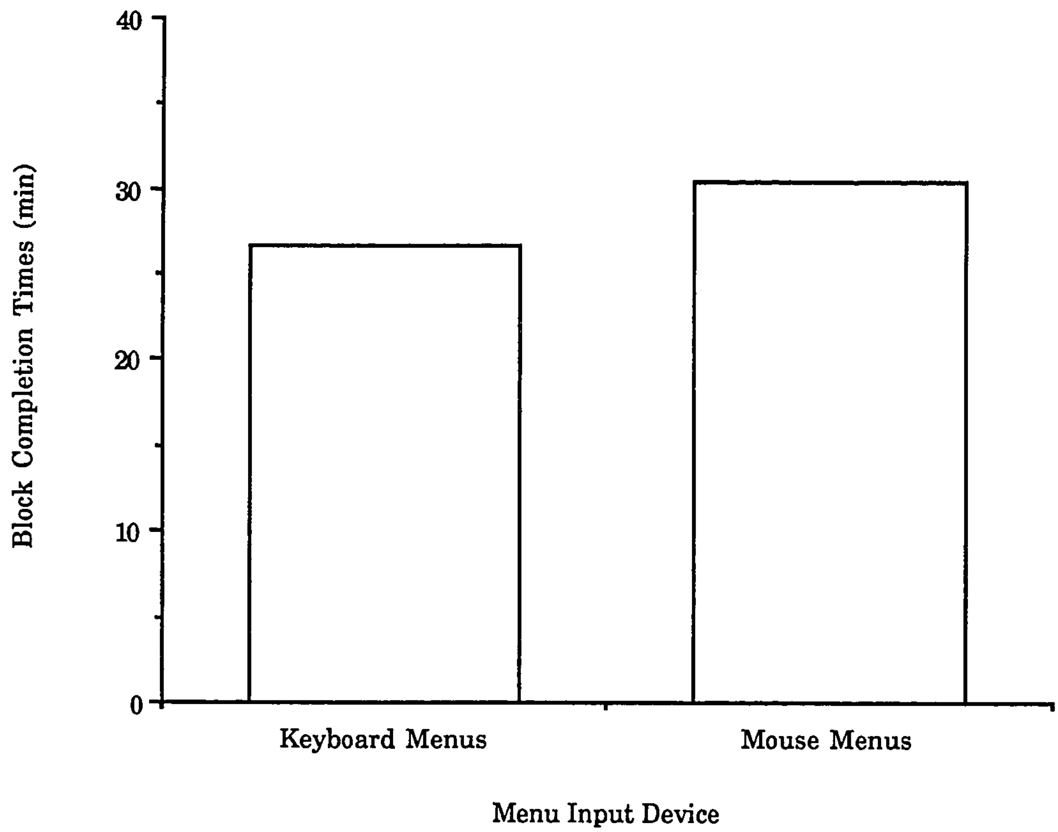

Figure 12. Block completion time by menu input device.

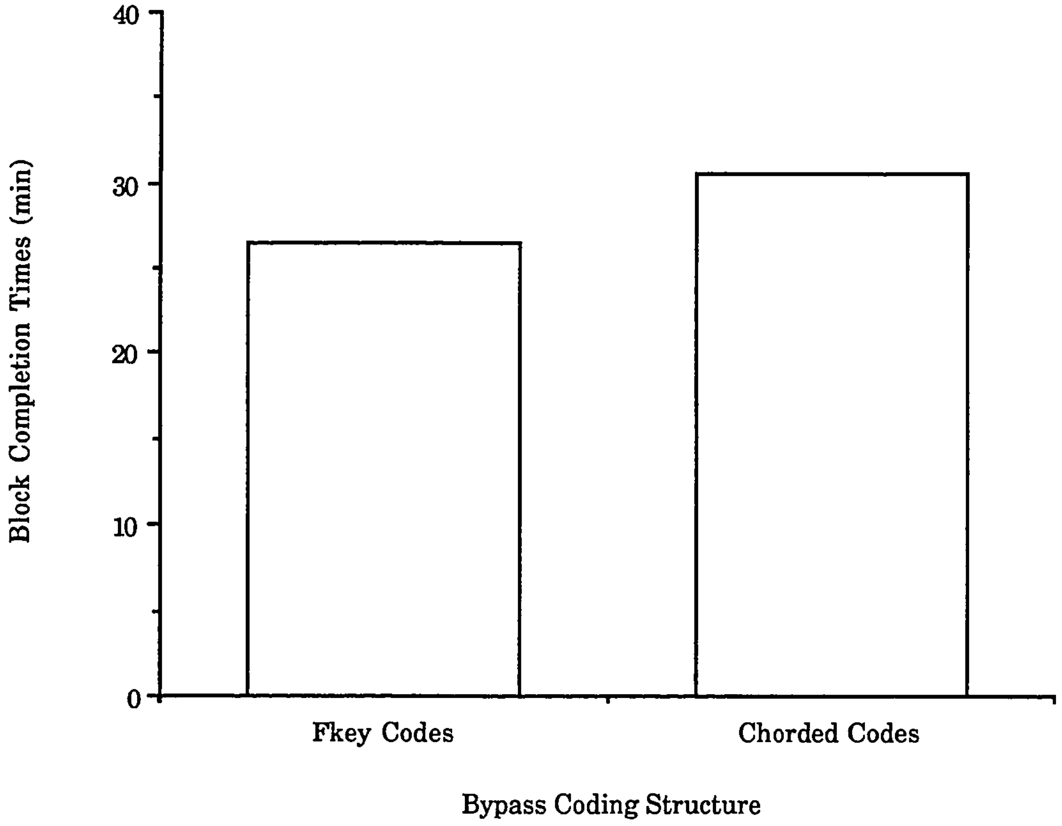

Figure 13. Block completion time by bypass coding structure.

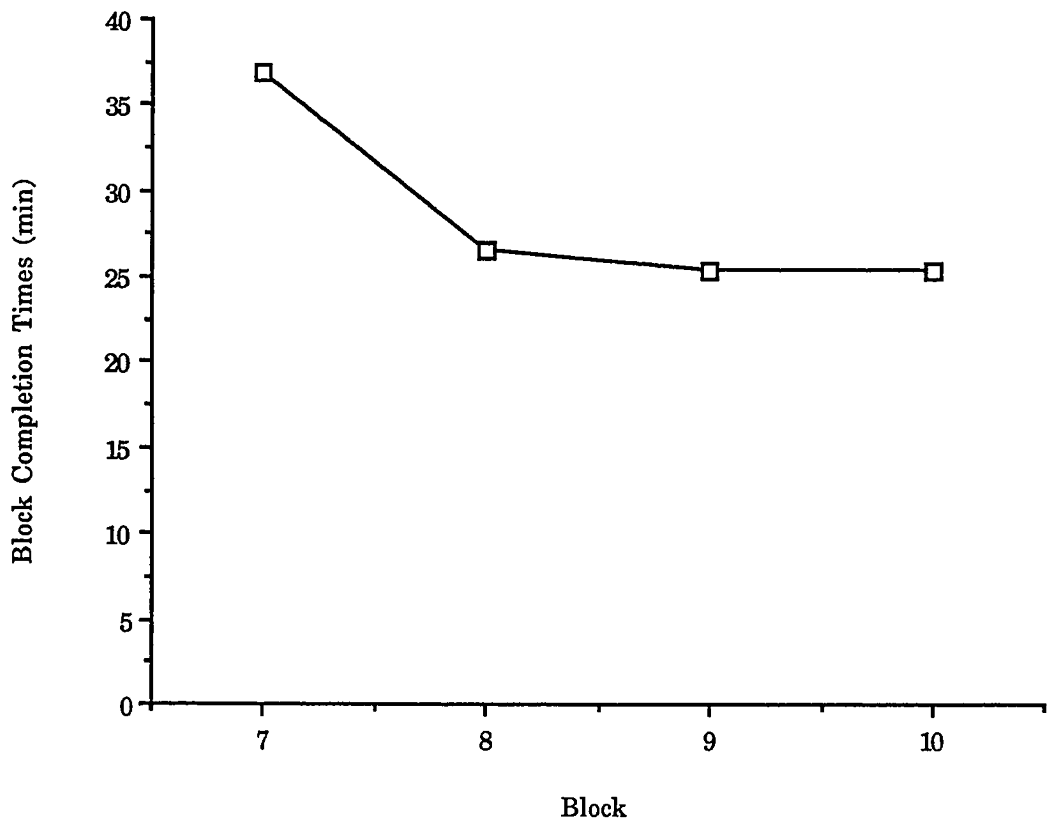

Figure 14. Block completion time by block.

TABLE 30

Newman-Keuls Tests on Block Completion Times by Block

Block	Mean
7	36.85 (A)
8	26.56 (B)
9	25.32 (B)
10	25.30 (B)

NOTE: Means sharing a common letter in parentheses were not significantly different ($p < 0.05$). Times given in minutes.

The results of the ANOVA also showed that the two-way interaction between block and menu input device was significant ($F(3,132) = 8.09$, $p < 0.01$; Figure 15). Simple effects tests were used to break down this interaction.

The simple main effect for menu input device was significant at Blocks seven and eight, but was not significant at Blocks nine and ten (Table 31). The means indicated that the Keyboard Menus groups completed Blocks seven and eight faster (32.87 and 24.79 min, for the respective blocks) than the Mouse Menus groups (40.83 and 28.33 min, for the respective blocks).

Menu designation time. Keystroke data were used to calculate the time between the last user input to precede the entry of a task-specified command to the entry of the menu designating portion of the bypass code or menu interaction sequence. These times were subsequently analyzed with a 2 X 2 X 4 ANOVA, with the between-groups factors representing menu input device and bypass coding structure, and the within-groups factor representing block.

The main effect of menu input device was significant ($F(1,44) = 14.84$, $p < 0.01$; Table 32, Figure 16), as was bypass structure ($F(1,44) = 11.01$, $p < 0.01$; Table 32, Figure 17), and block ($F(3,132) = 82.15$, $p < 0.01$; Table 32, Figure 18). The means indicated that the Keyboard Menus groups accessed target menus faster (2.60 s) than the Mouse Menus groups (3.42 s). The means also indicated that the Fkey Code groups accessed target menus faster (2.65 s) than the Chorded Code groups (3.36 s). Finally, Newman-Keuls tests showed that, on the average, menu designation times were longer for Block seven (3.74 s) than they were for Blocks eight, nine, and ten (2.92, 2.73, and 2.64 s, respectively; Table 33). Menu designation times were also longer for Block eight than they were for Blocks nine and ten.

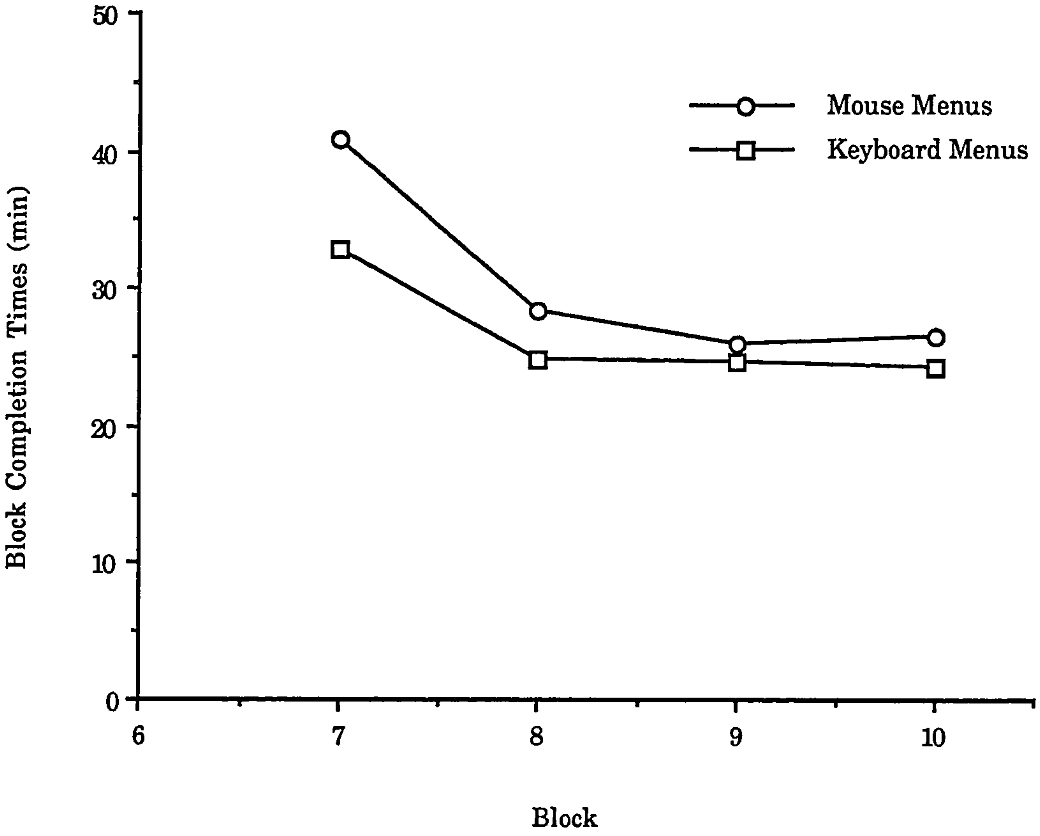

Figure 15. Block completion time by menu input device.

TABLE 31

Summary Table for Block Completion Times Simple-Effects F-Tests of Menu Input Device by Block

SOURCE	MS	df	F
DEVICE at BLOCK 7	761.75	1	9.93 **
SUBJ(DEV*BYPASS) at BLOCK 7	76.72	44	.
DEVICE at BLOCK 8	150.40	1	5.90 *
SUBJ(DEV*BYPASS) at BLOCK 8	25.50	44	.
DEVICE at BLOCK 9	16.20	1	1.00
SUBJ(DEV*BYPASS) at BLOCK 9	16.20	44	.
DEVICE at BLOCK 10	61.35	1	3.04
SUBJ(DEV*BYPASS) at BL0CK 10	20.15	44	.

* $p < 0.05$ ** $p < 0.01$

TABLE 32

ANOVA Summary Table for Menu Designation Time

SOURCE	MS	df	F	
MENU INPUT DEVICE	2301.338	1	14.84	**
BYPASS CODING STRUCTURE	1706.651	1	11.01	**
DEVICE*BYPASS	235.047	1	1.52	
SUBJ(DEVICE*BYPASS)	155.076	44	.	
BLOCK	850.592	3	82.15	**
BLOCK*DEVICE	207.170	3	20.01	**
BLOCK*BYPASS	31.901	3	3.08	*
BLOCK*DEVICE*BYPASS	28.247	3	2.73	*
BLOCK*SUBJ(DEVICE*BYPASS)	10.354	132	.	
OBSERVATION	116.575	71	.	
OBS*DEVICE	7.653	71	.	
OBS*BYPASS	6.921	71	.	
OBS*DEVICE*BYPASS	7.732	71	.	
OBS*SUBJ(DEVICE*BYPASS)	5.669	3124	.	
OBS*BLOCK	105.277	213	.	
OBS*BLOCK*DEVICE	7.009	213	.	
OBS*BLOCK*BYPASS	5.870	213	.	
OBS*BLOCK*DEVICE*BYPASS	6.210	213	.	
OBS*BLK*SUBJ(DEV*BYPASS)	5.210	9075	.	

* $p < 0.05$ ** $p < 0.01$

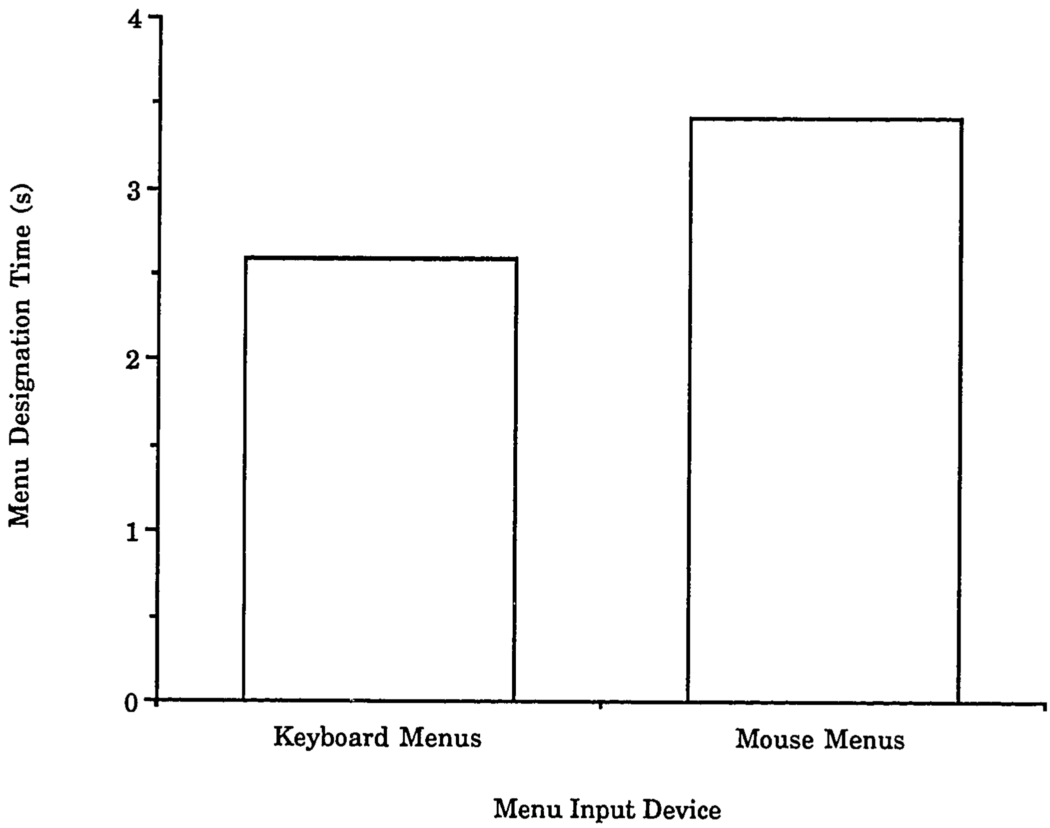

Figure 16. Menu designation time by menu input device.

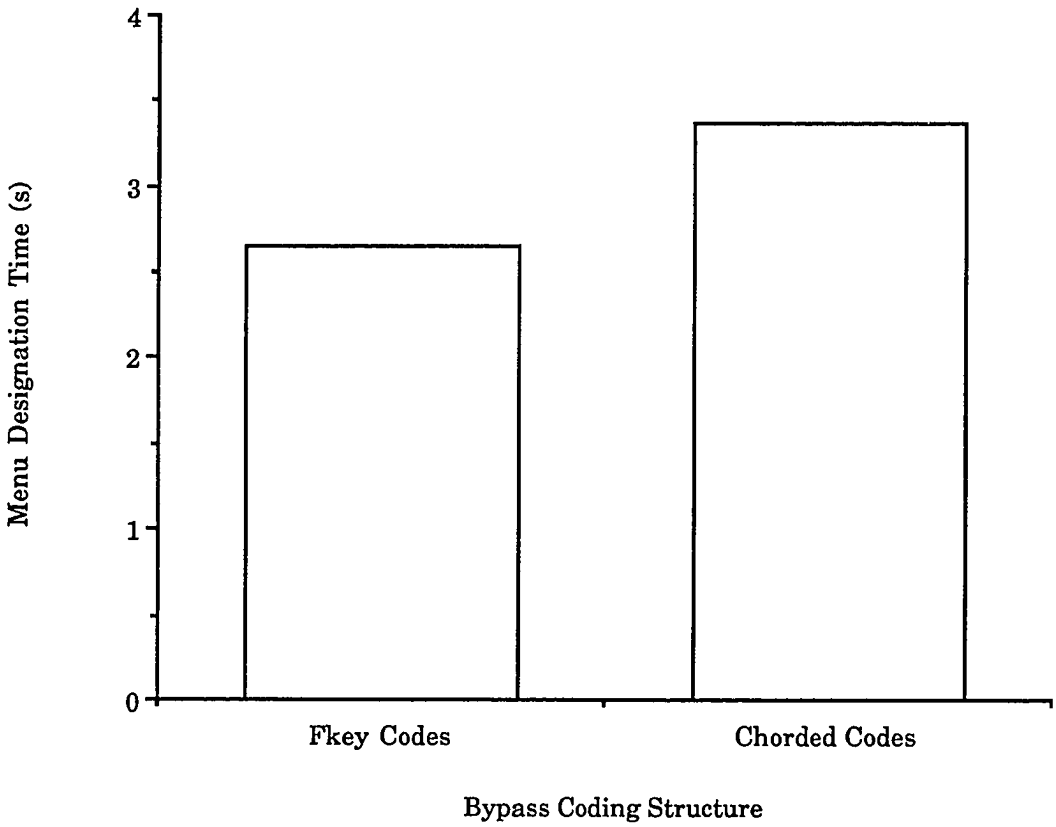

Figure 17. Menu designation time by bypass coding structure.

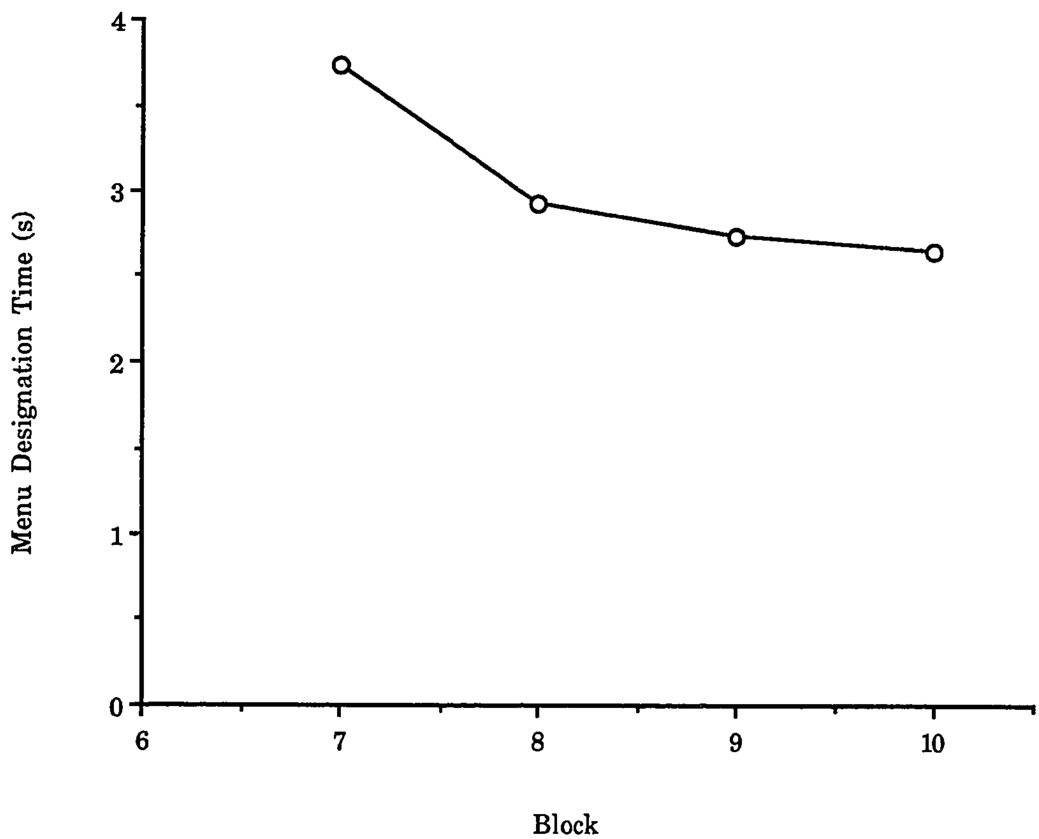

Figure 18. Menu designation time by block.

TABLE 33

Newman-Keuls Tests on Menu Designation Times by Block

Block	Mean
7	3.74 (A)
8	2.92 (B)
9	2.73 (C)
10	2.64 (C)

NOTE: Means sharing a common letter in parentheses were not significantly different ($p < 0.05$). Times given in seconds.

The two-way interaction between block and menu input device was significant ($F(3,132) = 20.01$, $p < 0.01$; Table 32, Figure 19) as was the interaction between block and bypass structure ($F(3,132) = 3.08$, $p < 0.05$; Table 32, Figure 20). Simple effects tests were used to break down these interactions.

The simple main effect for menu input device was significant at all four blocks (Table 34). The means indicated that during each of the four blocks the Keyboard Menus groups entered the menu designating portion of the bypass codes faster (2.99, 2.49, 2.47, and 2.43 s, for the respective blocks) than the Mouse Menus groups (4.50, 3.35, 2.98, and 2.86 s, for the respective blocks).

A break down of the interaction between block and bypass structure revealed that the simple main effect of bypass structure was significant at all four blocks (Table 35). In this case, the means indicated that during each of the four blocks the Fkey Codes groups entered the menu designating portion of the bypass codes faster (3.25, 2.62, 2.41, and 2.34 s, for the respective blocks) than the Chorded Codes groups (4.24, 3.22, 3.05, and 2.95 s, for the respective blocks).

The results of the ANOVA for menu designation times also revealed a significant three-way interaction between device, bypass, and block ($F(3,132) = 2.73$, $p < 0.05$; Table 32, Figure 21). The interaction effect was first broken down by bypass coding structure. Significant simple interaction effects were then broken down by block.

Simple effects tests indicated that there was a significant simple main effect for device and block for those groups that used Fkey Codes (Table 36). The means indicated that the Fkey Code/Keyboard Menus group entered the menu designation portion of the bypass codes faster (2.11 s) than the Fkey

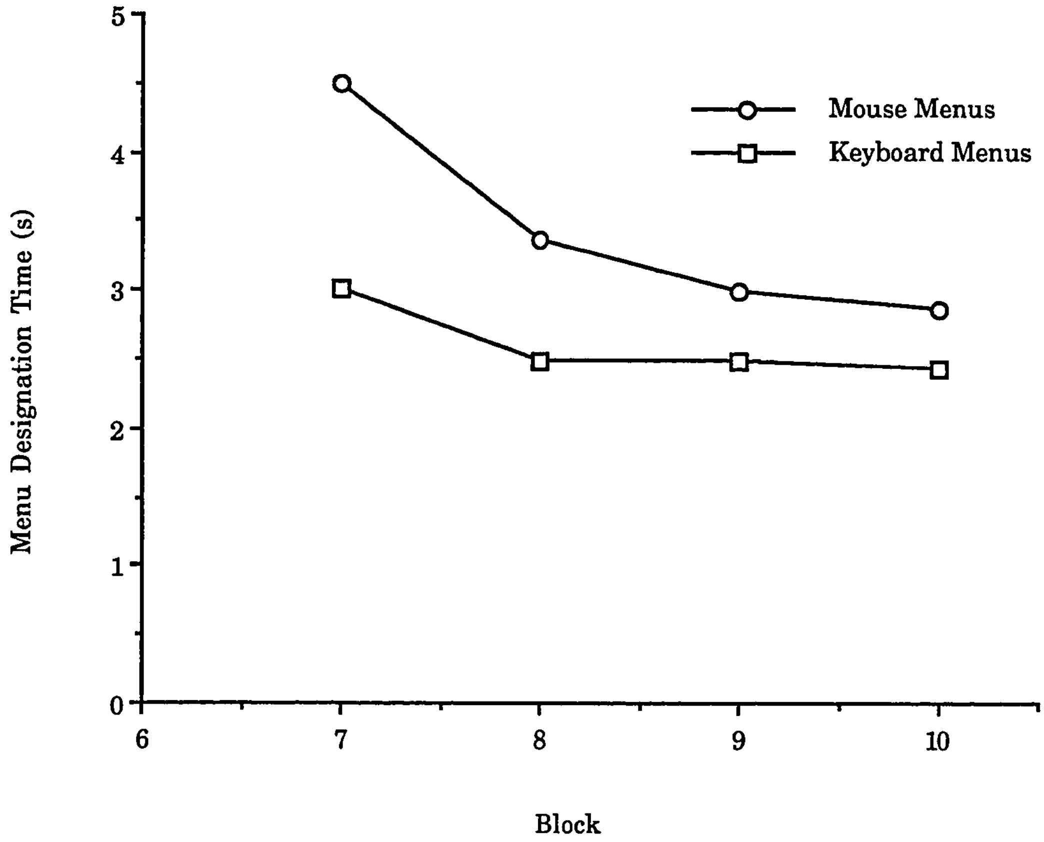

Figure 19. Menu designation times for menu input device by block interaction.

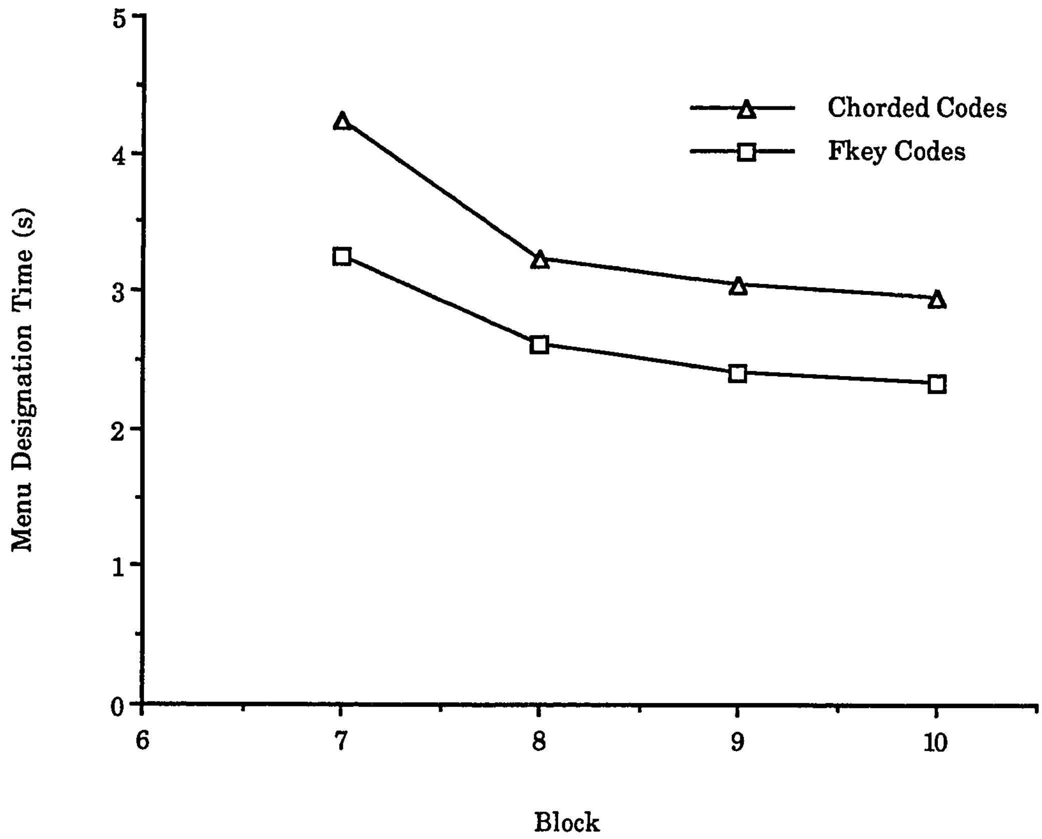

Figure 20. Menu designation times for bypass coding structure by block interaction.

TABLE 34

Summary Table for Menu Designation Times Simple-Effects F-Tests of Menu Input Device by Block

SOURCE	MS	df	F	
DEVICE at BLOCK 7	11916.39	1	25.23	**
SUBJ(DEVICE*BYPASS) at BLOCK 7	75.96	44	.	
DEVICE at BLOCK 8	625.46	1	10.43	**
SUBJ(DEVICE*BYPASS) at BLOCK 8	59.95	44	.	
DEVICE at BLOCK 9	218.42	1	7.34	**
SUBJ(DEVICE*BYPASS) at BLOCK 9	29.77	44	.	
DEVICE at BLOCK 10	162.58	1	7.95	**
SUBJ(DEVICE*BYPASS) at BL0CK 10	20.45	44	.	

* $p < 0.05$ ** $p < 0.01$

TABLE 35

Summary Table for Menu Designation Times Simple-Effects F-Tests of Bypass Coding Structure by Block

SOURCE	MS	df	F
BYPASS at BLOCK 7	829.61	1	10.92 **
SUBJ(DEV*BYPASS) at BLOCK 7	75.96	44	.
BYPASS at BLOCK 8	311.17	1	5.19 *
SUBJ(DEV*BYPASS) at BLOCK 8	59.95	44	.
BYPASS at BLOCK 9	344.60	1	11.57 **
SUBJ(DEV*BYPASS) at BLOCK 9	29.77	44	.
BYPASS at BLOCK 10	316.97	1	15.50 **
SUBJ(DEV*BYPASS) at BL0CK 10	20.45	44	.

* $p < 0.05$ ** $p < 0.01$

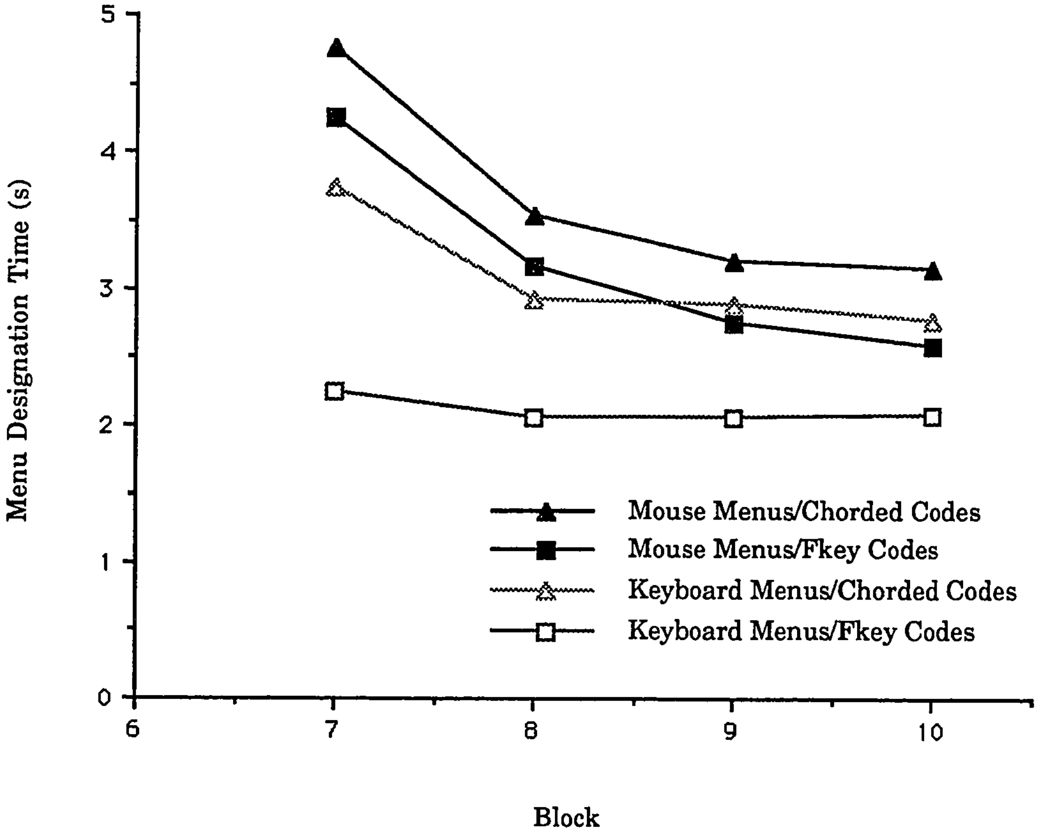

Figure 21. Menu designation times for menu input device by bypass coding structure by block interaction.

TABLE 36

Summary Table for Menu Designation Times Simple-Effects F-Tests for the Fkey Code Groups

SOURCE	MS	df	F
MENU INPUT DEVICE	1968.650	1	15.15 **
SUBJ(DEVICE)	129.958	22	.
BLOCK	291.665	3	28.29 **
BLOCK*DEVICE	189.485	3	18.38 **
BLOCK*SUBJ(DEVICE)	10.354	66	.
OBSERVATION	116.575	71	.
OBS*DEVICE	7.653	71	.
OBS*SUBJ(DEVICE)	5.330	1562	
OBS*BLOCK	105.277	213	.
OBS*BLOCK*DEVICE	7.009	213	.
OBS*BLK*SUBJ(DEVICE)	5.210	4577	.

* $p < 0.05$ ** $p < 0.01$

Code/Mouse Menus group (3.19 s). Newman-Keuls tests showed that, on the average, the menu designation times for the Fkey Code groups were longer for Block seven (3.25 s) than for any of the blocks that followed. The mean menu designation times for the Fkey Code groups were also longer for Block eight (2.62 s) than for Blocks nine and ten (2.41 and 2.34, respectively; Table 37).

A significant simple interaction between device and block was also found for those groups that used Fkey Codes (Table 36). Simple effects tests indicated that the main effect for device was significant at all four blocks (Table 38). The means indicated that during each of the four blocks the Fkey Code/Keyboard Menus group entered the menu designating portion of the bypass codes faster (2.25, 2.06, 2.06, and 2.08 s, for the respective blocks) than the Fkey Code/Mouse Menus group (4.25, 3.17, 2.75, and 2.59 s, for the respective blocks).

For those groups that used Chorded Codes, simple effects tests revealed a significant simple main effect for block (Table 39). Newman-Keuls tests showed that the menu designation times for the Chorded Code groups were longer for Block seven (4.24 s) than for any of the blocks that followed. The mean menu designation times for the Chorded Code groups were also longer for Block eight (3.22) than they were for Blocks nine and ten (3.05 and 2.95 s, respectively; Table 40).

The simple interaction between device and block was also found to be significant for those groups that used Chorded Codes (Table 39). A break down of the simple interaction between device and block revealed a simple main effect for device at Block seven (Table 41). The means indicated that the Block seven menu designation times were shorter for the Chorded Code/Keyboard Menus group (3.74 s) than they were for the Chorded

TABLE 37

Newman-Keuls Tests on Menu Designation Times by Block for Fkey Codes Groups

Block	Mean
7	3.25 (A)
8	2.62 (B)
9	2.41 (C)
10	2.34 (C)

NOTE: Means sharing a common letter in parentheses were not significantly different ($p < 0.05$). Times given in seconds.

TABLE 38

Summary Table for Menu Designation Times Simple-Effects F-Tests of Menu Input Device by Block for Fkey Codes Groups

SOURCE	MS	df	F
DEVICE at BLOCK 7	1706.89	1	31.44 **
SUBJ(DEVICE*BYPASS) at BLOCK 7	54.28	44	.
DEVICE at BLOCK 8	514.47	1	7.20 *
SUBJ(DEVICE*BYPASS) at BLOCK 8	71.50	44	.
DEVICE at BLOCK 9	205.67	1	11.29 **
SUBJ(DEVICE*BYPASS) at BLOCK 9	18.22	44	.
DEVICE at BLOCK 10	110.07	1	6.52 *
SUBJ(DEVICE*BYPASS) at BL0CK 10	16.88	44	.

* $p < 0.05$ ** $p < 0.01$

TABLE 39

Summary Table for Menu Designation Times Simple-Effects F-Tests for the Chorded Code Groups

SOURCE	MS	df	F
MENU INPUT DEVICE	567.73	1	3.15
SUBJ(DEVICE)	180.193	22	.
BLOCK	590.828	3	56.83 **
BLOCK*DEVICE	45.932	3	4.42 **
BLOCK*SUBJ(DEVICE)	10.397	66	.
OBSERVATION	69.089	71	.
OBS*DEVICE	5.633	71	.
OBS*SUBJ(DEVICE)	6.007	1562	
OBS*BLOCK	59.959	213	.
OBS*BLOCK*DEVICE	5.353	213	.
OBS*BLK*SUBJ(DEVICE)	5.640	4498	.

* $p < 0.05$ ** $p < 0.01$

TABLE 40

Newman-Keuls Tests on Menu Designation Times by Block for Chorded Code Groups

Block	Mean
7	4.24 (A)
8	3.22 (B)
9	3.05 (C)
10	2.95 (C)

NOTE: Means sharing a common letter in parentheses were not significantly different ($p < 0.05$). Times given in seconds.

TABLE 41

Summary Table for Menu Designation Times Simple-Effects F-Tests of Menu Input Device by Block for Fkey Codes Groups

SOURCE	MS	df	F
DEVICE at BLOCK 7	437.639	1	4.48 *
SUBJ(DEVICE*BYPASS) at BLOCK 7	97.642	44	.
DEVICE at BLOCK 8	163.555	1	3.38
SUBJ(DEVICE*BYPASS) at BLOCK 8	48.340	44	.
DEVICE at BLOCK 9	44.843	1	1.09
SUBJ(DEVICE*BYPASS) at BLOCK 9	41.322	44	.
DEVICE at BLOCK 10	59.493	1	2.48
SUBJ(DEVICE*BYPASS) at BL0CK 10	16.88	44	.

* $p < 0.05$ ** $p < 0.01$

Code/Mouse Menus group (4.76 s). The simple effects tests for the remaining blocks were not significant (Table 41).

Command designation time. Keystroke data were used to calculate the time from the input of the menu designating portion of a task-specified command selection sequence to the entry of the command designating portion of the sequence. These times were subsequently analyzed with a 2 X 2 X 4 ANOVA, with the between-groups factors representing menu input device and bypass coding structure, and the within-groups factor representing block.

The main effect of menu interaction device was significant ($F(1,44) = 6.50$, $p < 0.05$; Table 42, Figure 22) as was the effect of block ($F(3,132) = 29.09$, $p < 0.01$; Table 42, Figure 23). The means indicated that the Keyboard Menus groups entered the command designating portion of the bypass codes faster (0.92 s) than the Mouse Menu groups (1.12 s). Newman-Keuls tests showed that, on the average, the command designation times were longer for Block seven (1.19 s) than they were for Blocks eight, nine, and ten (1.01, 0.94, and 0.94 s, respectively; Table 43).

The two-way interaction between block and menu input device was significant ($F(3,132) = 12.17$, $p < 0.01$; Table 42, Figure 24) as was the interaction between block and bypass structure ($F(3,132) = 2.68$, $p < 0.05$; Table 42, Figure 25). Simple effects tests were used to break down these interactions.

The simple main effect for menu input device was significant at Blocks seven and eight, but was not significant at Blocks nine and ten (Table 44). The means for Blocks seven and eight indicated that the Keyboard Menus groups entered the command designating portion of the bypass codes faster

TABLE 42

ANOVA Summary Table for Command Designation Time

SOURCE	MS	df	F
MENU INPUT DEVICE	128.896	1	6.50 *
BYPASS CODING STRUCTURE	19.846	1	1.00
DEVICE*BYPASS	27.447	1	1.38
SUBJ(DEVICE*BYPASS)	19.823	44	.
BLOCK	47.236	3	29.09 **
BLOCK*DEVICE	19.763	3	12.17 **
BLOCK*BYPASS	4.358	3	2.68 *
BLOCK*DEVICE*BYPASS	0.931	3	0.57
BLOCK*SUBJ(DEVICE*BYPASS)	1.624	132	.
OBSERVATION	4.823	71	.
OBS*DEVICE	0.956	71	.
OBS*BYPASS	0.896	71	.
OBS*DEVICE*BYPASS	0.685	71	.
OBS*SUBJ(DEVICE*BYPASS)	0.680	3124	.
OBS*BLOCK	4.732	213	.
OBS*BLOCK*DEVICE	0.815	213	.
OBS*BLOCK*BYPASS	0.637	213	.
OBS*BLOCK*DEVICE*BYPASS	0.564	213	.
OBS*BLK*SUBJ(DEV*BYPASS)	0.652	9075	.

* $p < 0.05$ ** $p < 0.01$

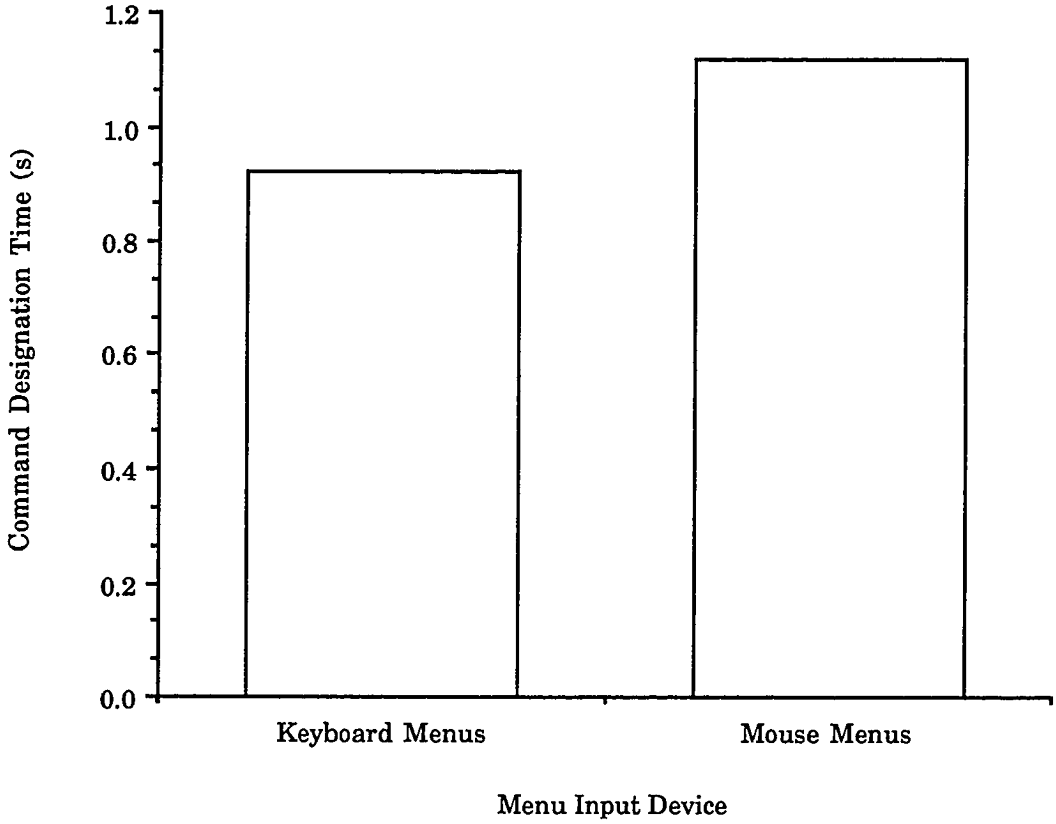

Figure 22. Command designation time by menu input device.

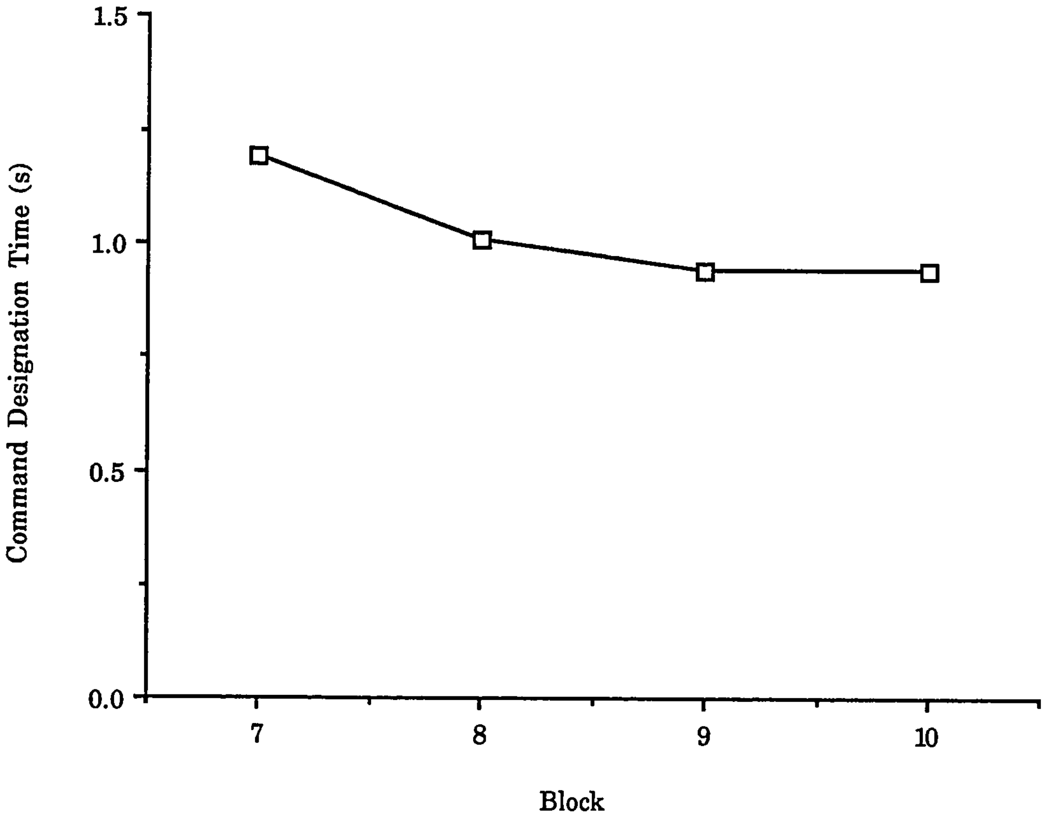

Figure 23. Command designation time by block.

TABLE 43

Newman-Keuls Tests on Menu Designation Times by Block

Block	Mean
7	1.19 (A)
8	1.01 (B)
9	0.94 (B)
10	0.94 (B)

NOTE: Means sharing a common letter in parentheses were not significantly different ($p < 0.05$). Times given in seconds.

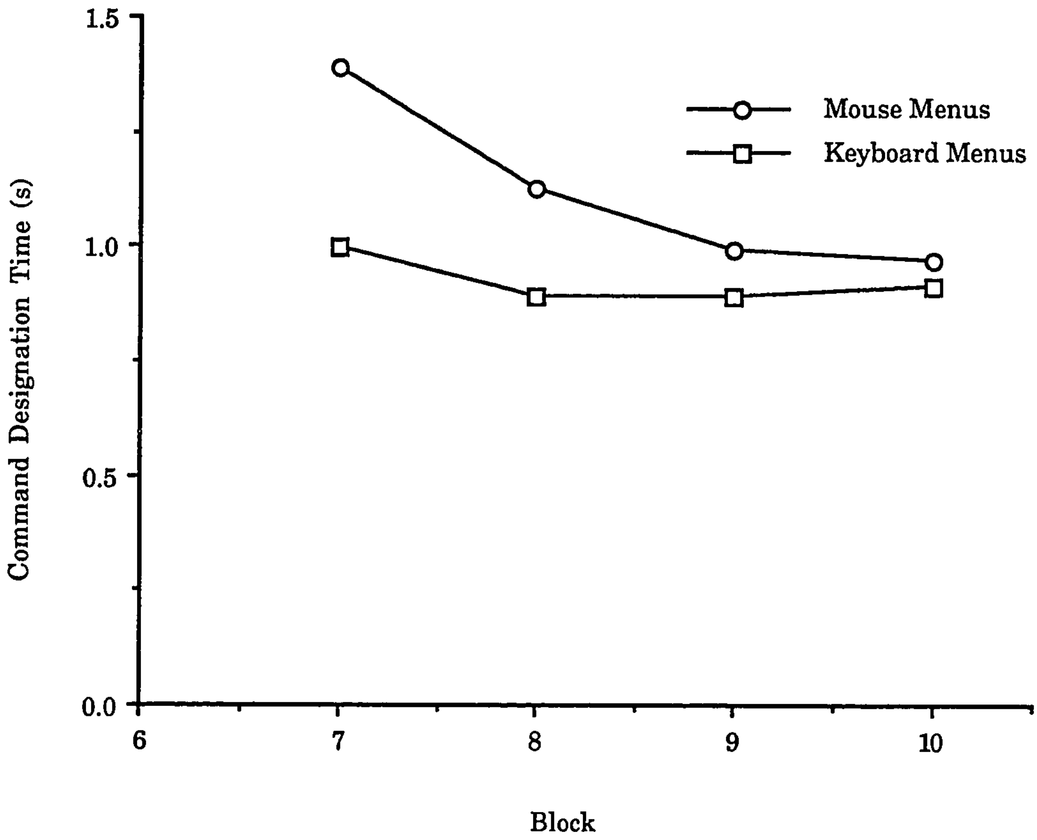

Figure 24. Command designation times for menu input device by block interaction.

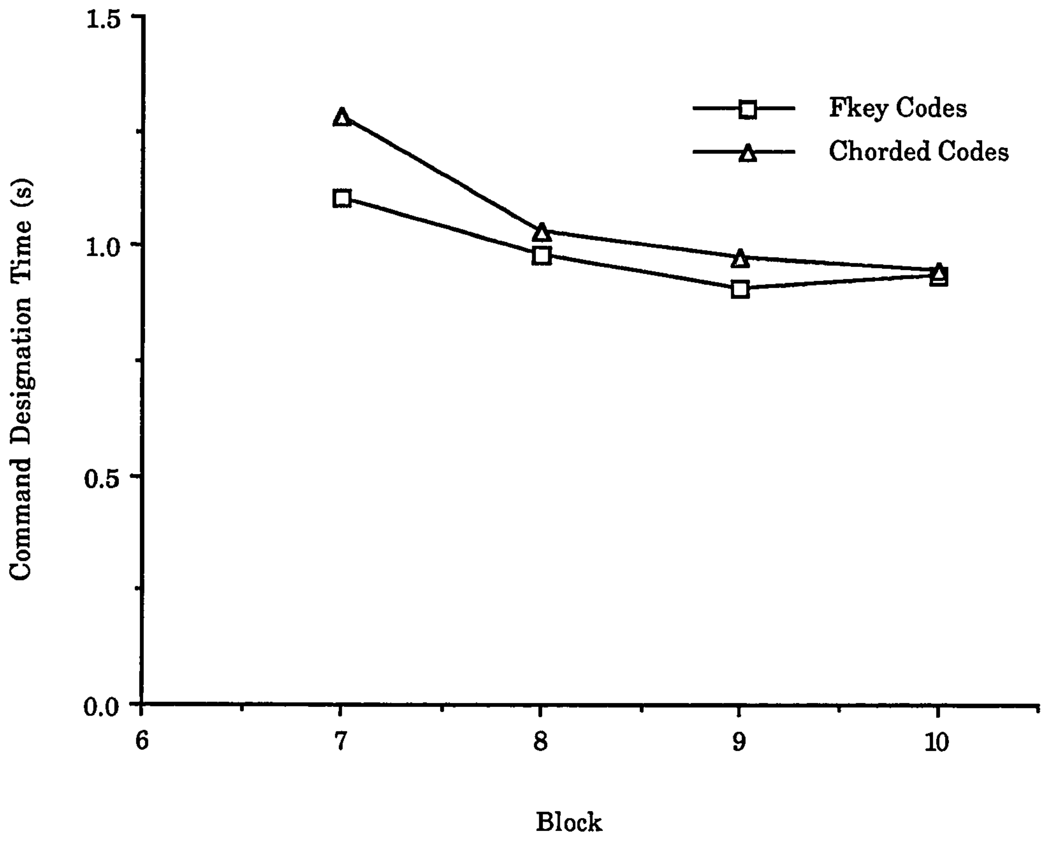

Figure 25. Command designation times for bypass coding structure by block interaction.

TABLE 44

Summary Table for Command Designation Times Simple-Effects F-Tests of Menu Input Device by Block

SOURCE	MS	df	F	
DEVICE at BLOCK 7	131.42	1	15.85	**
SUBJ(DEV*BYPASS) at BLOCK 7	8.29	44	.	
DEVICE at BLOCK 8	45.54	1	6.16	*
SUBJ(DEV*BYPASS) at BLOCK 8	7.39	44	.	
DEVICE at BLOCK 9	8.29	1	1.97	
SUBJ(DEV*BYPASS) at BLOCK 9	4.20	44	.	
DEVICE at BLOCK 10	2.93	1	0.61	
SUBJ(DEV*BYPASS) at BL0CK 10	4.81	44	.	

* $p < 0.05$ ** $p < 0.01$

(1.00 and 0.89 s, for the respective blocks) than the Mouse Menu groups (1.39 and 1.12 s, for the respective blocks.

Despite the significant interaction of block and bypass structure none of the simple main effects tests were found to be significant (Table 45).

Combined menu and command designation time. Keystroke data were used to calculate the time between the last user input to precede the entry of a task-specified command to the entry of the command designating portion of the bypass code or menu interaction sequence. These times were subsequently analyzed with a 2 X 2 X 4 ANOVA, with the between-groups factors representing menu input device and bypass coding structure, and the within-groups factor representing block.

The main effect of menu interaction device was significant ($F(1,44) = 14.94$, $p < 0.01$; Table 46, Figure 26), as was bypass structure ($F(1,44) = 8.89$, $p < 0.01$; Table 46, Figure 27), and block ($F(3,132) = 87.73$, $p < 0.01$; Table 46, Figure 28). The means indicated that the Keyboard Menus groups entered both the menu and command designating portion of bypass codes faster (3.52 s) than the Mouse Menus groups (4.54 s). The means also showed that the Fkey Code groups entered both the menu and command designating portion of the bypass codes faster (3.63 s) than the Chorded Code groups (4.42 s). Finally, Newman-Keuls tests showed that, on the average, the combined menu and command designation times were longer for Block seven (4.93 s) than they were for Blocks eight, nine, and ten (3.92, 3.67, and 3.58 s, respectively). Combined menu and command designation times were also longer for Block eight than they were for Blocks nine and ten (Table 86).

TABLE 45

Summary Table for Command Designation Times Simple-Effects F-Tests of Bypass Coding Structure by Block

SOURCE	MS	df	F
BYPASS at BLOCK 7	26.46	1	3.19
SUBJ(DEVICE*BYPASS) at BLOCK 7	8.29	44	.
BYPASS at BLOCK 8	2.10	1	0.28
SUBJ(DEVICE*BYPASS) at BLOCK 8	7.39	44	.
BYPASS at BLOCK 9	4.28	1	1.02
SUBJ(DEVICE*BYPASS) at BLOCK 9	29.77	44	.
BYPASS at BLOCK 10	0.09	1	0.02
SUBJ(DEVICE*BYPASS) at BL0CK 10	20.45	44	4.81

* $p < 0.05$ ** $p < 0.01$

TABLE 46

ANOVA Summary Table for Combined Menu and Command Designation Times

SOURCE	MS	df	F	
MENU INPUT DEVICE	3519.513	1	14.94	**
BYPASS CODING STRUCTURE	2094.577	1	8.89	**
DEVICE*BYPASS	123.505	1	0.52	
SUBJ(DEVICE*BYPASS)	235.508	44	.	
BLOCK	1297.432	3	87.73	**
BLOCK*DEVICE	353.848	3	23.93	**
BLOCK*BYPASS	58.477	3	3.95	**
BLOCK*DEVICE*BYPASS	36.803	3	2.49	
BLOCK*SUBJ(DEVICE*BYPASS)	14.788	132	.	
OBSERVATION	155.683	71	.	
OBS*DEVICE	11.747	71	.	
OBS*BYPASS	7.939	71	.	
OBS*DEVICE*BYPASS	8.493	71	.	
OBS*SUBJ(DEVICE*BYPASS)	6.646	3124	.	
OBS*BLOCK	135.256	213	.	
OBS*BLOCK*DEVICE	9.431	213	.	
OBS*BLOCK*BYPASS	7.189	213	.	
OBS*BLOCK*DEVICE*BYPASS	6.204	213	.	
OBS*BLK*SUBJ(DEVICE*BYPASS)	6.163	9075	.	

* $p < 0.05$ ** $p < 0.01$

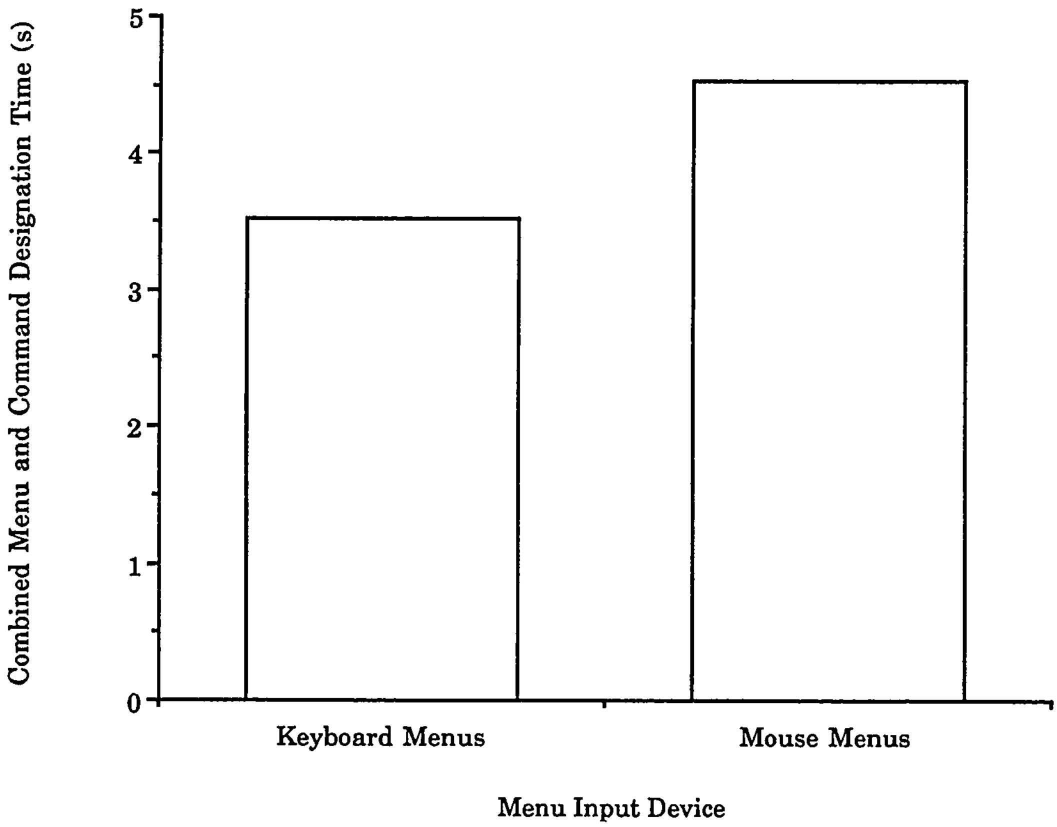

Figure 26. Combined menu and command designation time by menu input device.

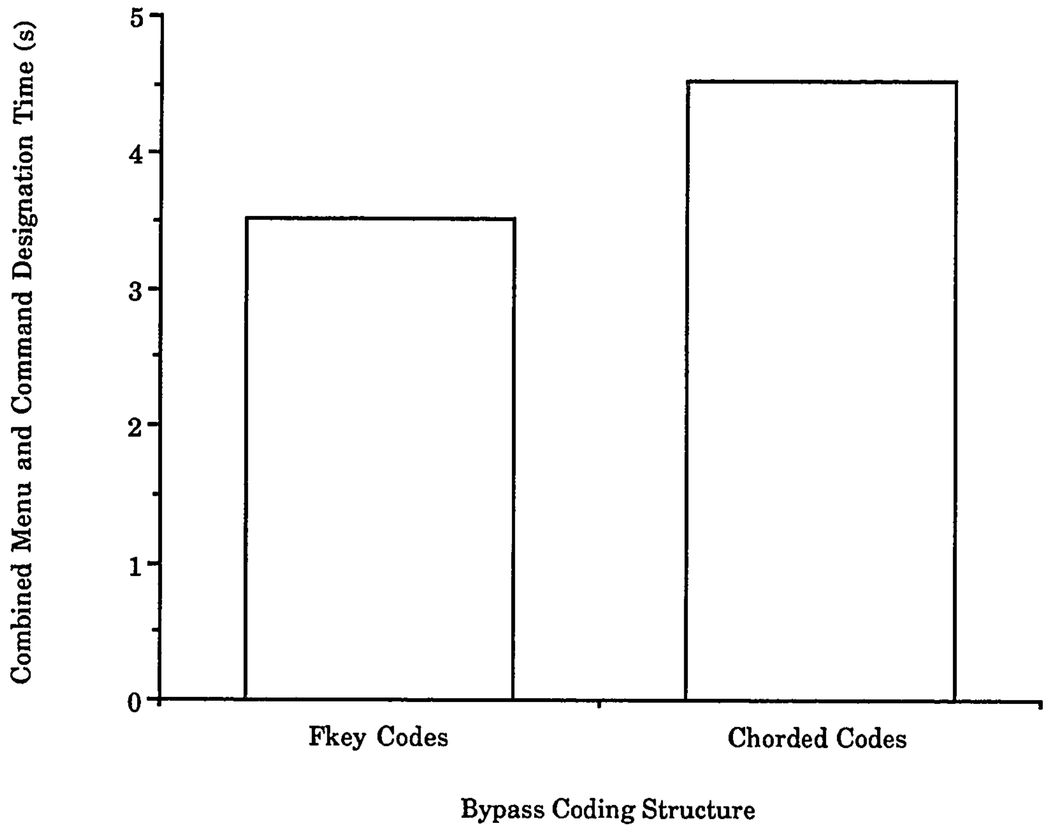

Figure 27. Combined menu and command designation time by bypass coding structure.

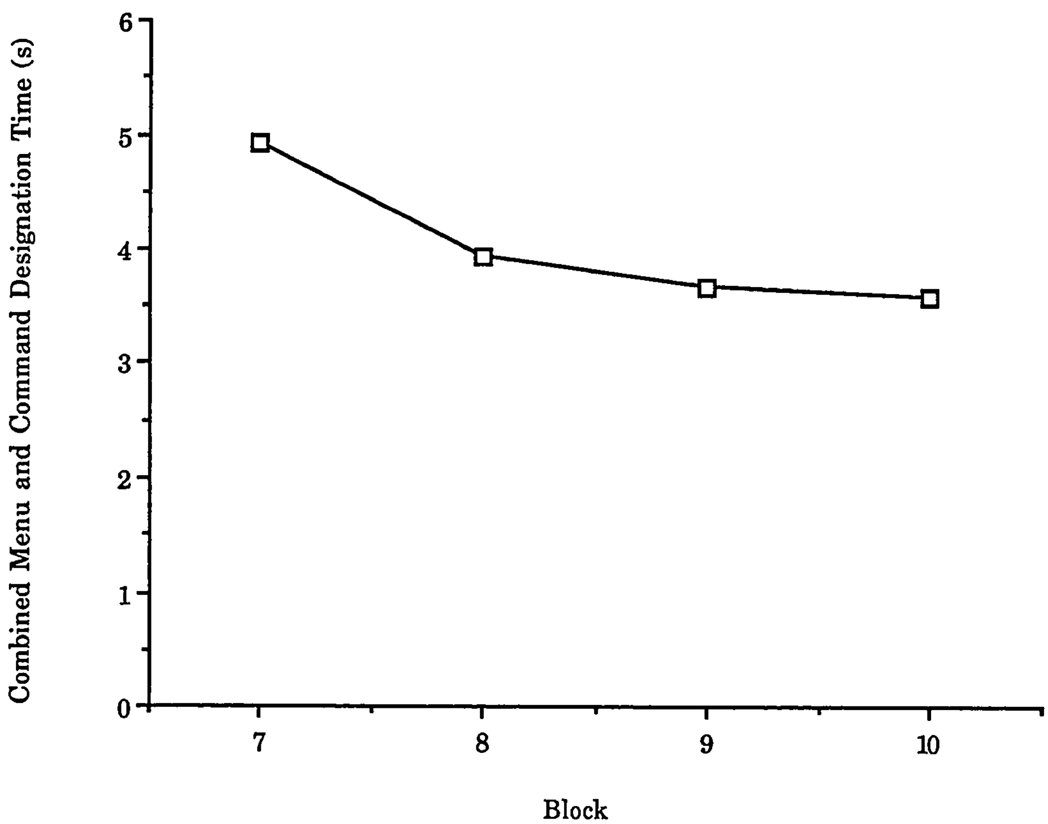

Figure 28. Combined menu and command designation time by block.

TABLE 47

Newman-Keuls Tests on Combined Menu and Command Designation Times by Block

Block	Mean
7	4.93 (A)
8	3.92 (B)
9	3.67 (C)
10	3.58 (C)

NOTE: Means sharing a common letter in parentheses were not significantly different ($p < 0.05$). Times given in seconds.

The two-way interaction between block and menu input device was significant ($F(3,132) = 23.93$, $p < 0.01$; Table 46, Figure 29), as was the interaction between block and bypass structure ($F(3,132) = 3.95$, $p < 0.01$; Table 46, Figure 30). Simple effects tests were used to break down these interactions.

The simple main effect for menu input device was significant at all four blocks (Table 48). The means showed that the Keyboard Menus groups entered both portions of the code faster on all four blocks (3.99, 3.38, 3.36, and 3.34 s, for the respective blocks) relative to the Mouse Menus groups (5.89, 4.47, 3.97, and 3.83 s, for the respective blocks).

A break down of the interaction between block and bypass structure revealed that the simple main effect of bypass structure was significant at Blocks seven, nine, and ten (Table 49). The means for these blocks showed that the groups using Fkey Codes entered the entire bypass code sequence faster (4.35, 3.31, and 3.27 s, for the respective blocks) than the groups using Chorded Codes (5.52, 4.02, and 3.89 s, for the respective blocks).

Errors. As was the case for the menu selection phase of the experiment, video tape and keystroke recordings were used to identify and tally three types of errors. Errors were recorded if subjects: selected commands not specified in the task materials (Extra Actions), substituted unspecified commands for task-specified commands (Substitutions); or omitted task-specified commands (Omissions). Errors were cross-tabulated by type, menu input device, and bypass coding structure (Table 50). The resulting frequency counts were analyzed through a series of Chi-Square tests.

For each error type, a Chi-Square test of independence was performed on the menu input device by bypass coding structure contingency table.

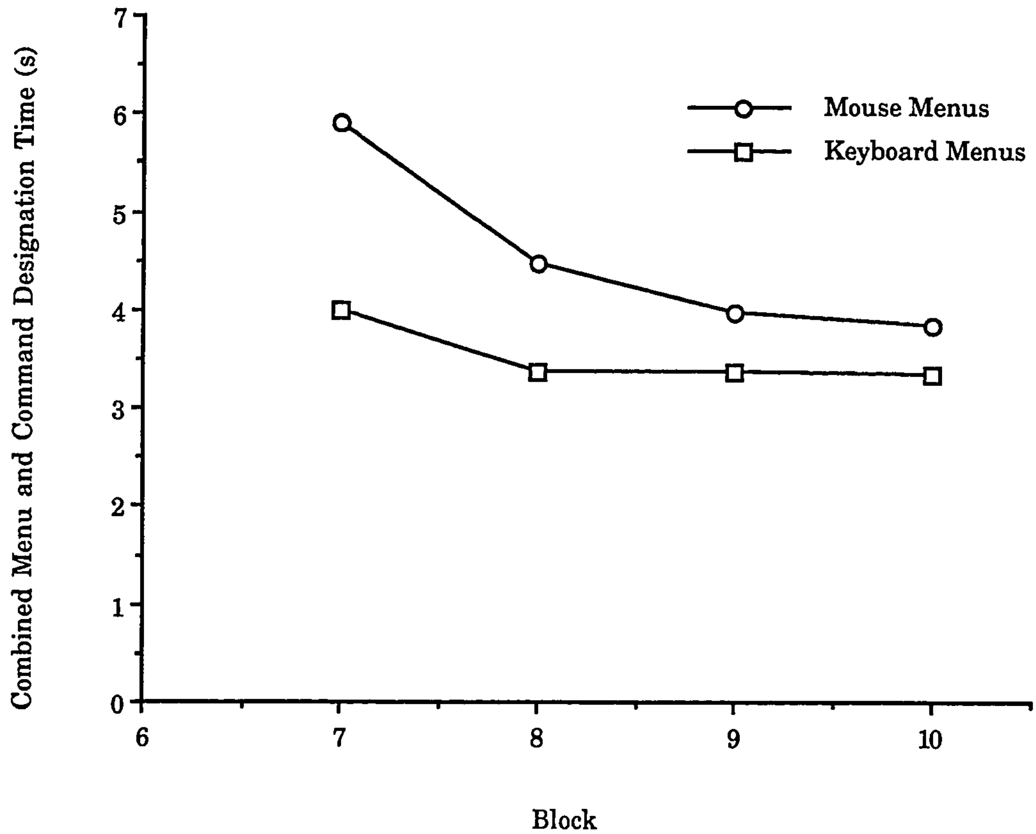

Figure 29. Combined menu and command designation times for menu input device by block interaction.

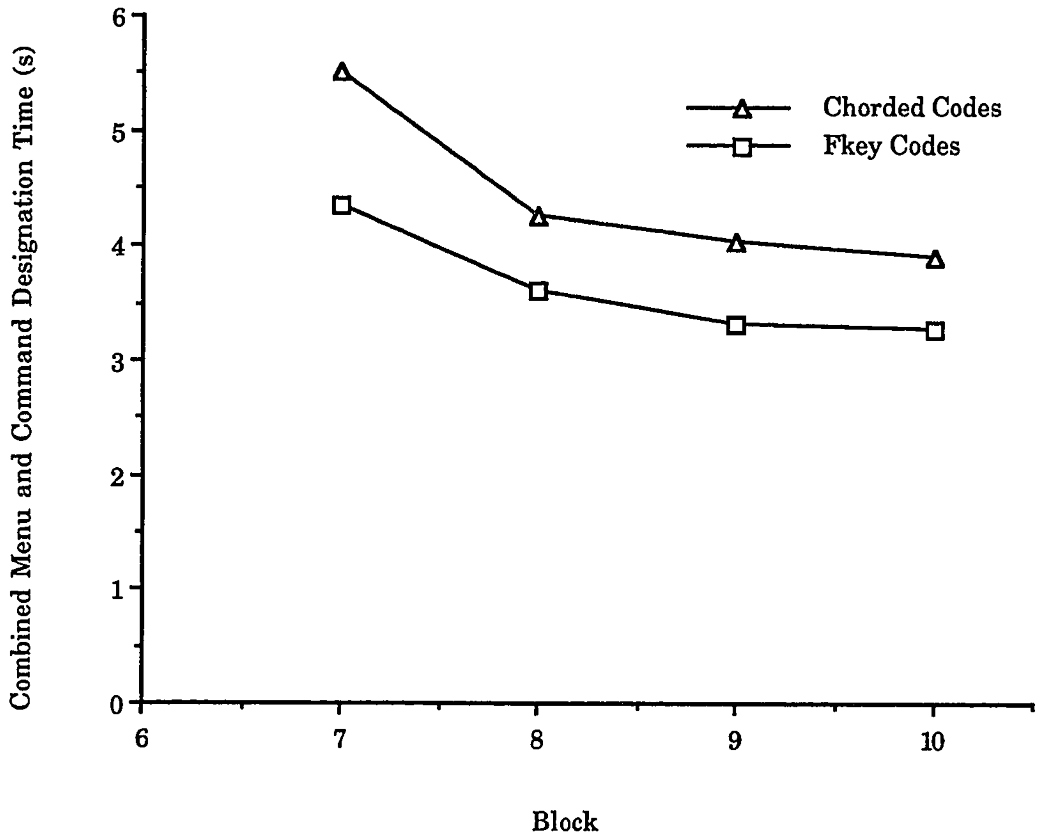

Figure 30. Combined menu and command designation times for bypass coding structure by block interaction.

TABLE 48

Summary Table for Combined Menu and Command Designation Times

Simple-Effects F-Tests of Menu Input Device by Block

SOURCE	MS	df	F
DEVICE at BLOCK 7	3051.52	1	28.24 **
SUBJ(DEV*BYPASS) at BLOCK 7	108.05	44	.
DEVICE at BLOCK 8	1008.55	1	10.67 **
SUBJ(DEV*BYPASS) at BLOCK 8	94.51	44	.
DEVICE at BLOCK 9	311.80	1	7.18 *
SUBJ(DEV*BYPASS) at BLOCK 9	43.45	44	.
DEVICE at BLOCK 10	209.18	1	6.18 *
SUBJ(DEV*BYPASS) at BL0CK 10	33.86	44	.

* $p < 0.05$ ** $p < 0.01$

TABLE 49

Summary Table for Combined Menu and Command Designation Times

Simple-Effects F-Tests of Bypass Coding Structure by Block

SOURCE	MS	df	F
BYPASS at BLOCK 7	1152.37	1	10.66 **
SUBJ(DEVICE*BYPASS) at BLOCK 7	108.05	44	.
BYPASS at BLOCK 8	364.42	1	3.86
SUBJ(DEVICE*BYPASS) at BLOCK 8	94.51	44	.
BYPASS at BLOCK 9	425.64	1	9.80 **
SUBJ(DEVICE*BYPASS) at BLOCK 9	43.45	44	.
BYPASS at BLOCK 10	327.57	1	9.68 **
SUBJ(DEVICE*BYPASS) at BL0CK 10	20.45	44	.

* $p < 0.05$ ** $p < 0.01$

TABLE 50

Cross-tabulation of Errors by Error Type, Device, and Bypass Coding Structure

		CODING STRUCTURE		
ERROR TYPE	DEVICE	FKEY	CHORDED	TOTAL
EXTRA ACTIONS	KEYBOARD	70	83	153
	MOUSE	80	147	227
	TOTAL	150	230	380
SUBSTITUTIONS	KEYBOARD	1	21	22
	MOUSE	10	70	80
	TOTAL	11	91	102
OMISSIONS	KEYBOARD	47	20	67
	MOUSE	35	54	89
	TOTAL	82	74	156

These tests indicated that menu input device and bypass coding structure were independent in terms of frequency of observed Substitutions (Chi-Square = 1.135, df = 1, $p > 0.05$), but that device and structure were related in terms of frequency of observed Extra Actions and Omissions (Chi-Square = 4.225, df = 1, $p < 0.05$ and Chi-Square = 14.565, df = 1, $p < 0.05$, respectively).

In light of the finding that device and structure were not related in terms of frequency of Substitutions committed, binomial comparisons were conducted on the device and structure totals. These tests revealed that the number of Substitutions was not evenly distributed across device or structure. Seventy eight percent of the observed Substitutions were committed by the Mouse Menus groups (total = 80) and twenty two percent were committed by the Keyboard Menus groups (total = 22, $p < .05$). Eighty nine percent of the observed Substitutions were committed by the Chorded Codes groups (total = 91) and eleven percent were committed by the Fkey Codes groups (total = 11, $p < .05$).

In light of the finding that device and structure were related in terms of frequency of Extra Actions, binomial comparisons were conducted on cell totals. These tests revealed that the Chorded Code/Mouse Menus group committed significantly more Extra Actions (total = 147) than did the Fkey Code/Mouse Menus group (total = 80, $p < .05$). In contrast, no difference was found between the number of Extra Actions committed by the Chorded Code/Keyboard Menus group and the Fkey Code/Keyboard Menu group (totals = 83 and 70, respectively).

Finally, since device and structure were related in terms of frequency of observed Omissions, binomial comparisons were conducted on cell totals. These tests revealed that the Chorded Code/Mouse Menus group committed

significantly more Omissions (total = 54) than did the Fkey Code/Mouse Menus group (total = 35, $p < .05$). Furthermore, the Fkey Code/Keyboard Menus group committed more Omissions (total = 47) than did the Chorded Code/Keyboard Menu group (totals = 20, $p < .05$).

The three error types were also collapsed and re-tabulated by menu input device and bypass coding structure. A Chi-Square test of independence was, in turn, performed on the menu input device by bypass coding structure contingency table of overall errors. This test indicated that device and structure were related in terms of the overall frequency of observed errors (Chi-Square = 26.22, df = 1, $p < 0.05$).

In lieu of the finding that device and structure were related in terms of frequency of overall errors, binomial comparisons were conducted on cell totals. These tests revealed that the Chorded Code/Mouse Menus group committed significantly more errors overall (total = 306) than did the Fkey Code/Mouse Menus group (total = 125, $p < .05$). In contrast, no overall difference was found between the frequency of errors committed by the Chorded Code/Keyboard Menus group and the Fkey Code/Keyboard Menu group (totals = 124 and 118, respectively).

Subjective rankings. The two-choice rank-order items were analyzed with binomial comparison tests. The three-choice items were analyzed with Chi-Square tests for goodness of fit to a uniform distribution. A rank-order difference was found for eight of the 22 two-choice items (Table 51).

Subjects indicated that they liked selecting commands with the bypass codes more than they did with the menus. Relative to menus, subjects also rated bypass codes as a more convenient and a more natural feeling method of command selection. Furthermore, subjects indicated that they felt they

TABLE 51

Binomial Tests on Two-Choice Items

	Control Style		
Dimension	Bypass	Menus	Difference
I selected commands faster with ...	41	7	34 *
It was easier to use ...	26	22	4
It was more convenient to use ...	31	15	16 *
It was more awkward to use ...	21	27	6
I felt more confident with ...	18	30	12
It was easier to learn to select with ...	8	40	32 *
It was more frustrating to use ...	24	24	0
It felt more natural to use ...	34	14	20 *
The best way to select commands is ...	28	20	8
It required more concentration to use ...	38	10	28 *
I did a better job when I used ...	28	19	9
I finished the editing task faster with ...	38	10	28 *
It felt more cumbersome to use ...	18	30	12
I felt more in control when using ...	22	26	4
I felt more anxious when using ...	36	12	24 *
Overall the style I like most was ...	32	16	16 *

* $p < 0.05$

selected commands faster with the bypass codes and that they finished the editing tasks faster when using bypass codes. On the other hand, subjects indicated that they felt it was easier to learn how to select commands with menus and that using menus did not require as much concentration as did bypass codes. Finally, subjects also indicated that they felt less anxious when selecting commands from menus than they did when selecting commands with bypass codes.

For all six three-choice items the responses were found to be unevenly distributed across the three response categories (i.e., bypass, menu, or both). In each case, the majority of the subjects indicated that it would be advantageous to have "both" menu and bypass code access to commands (Table 52).

TABLE 52

Chi-Square Test for Goodness of Fit to a Uniform Distribution on Three-Choice Items

Dimension	Control Style			Chi-Square
	Codes	Menus	Both	
easier to learn	0	6	42	64.50 *
easier to make selections	6	12	30	19.50 *
less frustrating to use	1	13	34	34.87 *
less awkward to use	6	14	28	15.50 *
perform better with	11	11	26	9.37 *
prefer	11	7	30	18.87 *

* $p < 0.05$

DISCUSSION

Research findings and anecdotal evidence strongly suggest that there are benefits to be derived from blending menu and bypass code-based styles of control. An examination of the current generation of graphic user interface platforms reveals that these software environments do, in fact, blend menu and bypass code-based styles of control. Further examination of these platforms reveals that the command selection techniques they employ are very similar (i.e., an auxiliary pointing device serves as the focal means for making menu-based command selections, and ill-organized modifier key/letter key pairings are used to make bypass code-based command selections). For a number of reasons, the overall control strategy employed by these platforms appears questionable at best. First, the decision to focus on pointing as the method for selecting commands from menus appears to have been made without the benefit of empirical support. Second, the decision to rely on broad, flat bypass coding structures appears extremely ill-advised given research findings (including findings from basic verbal learning research). Finally, there is little continuity between the menu-based and bypass code-based styles of control used in these systems. The present investigation addressed these issues through a two phased experiment.

Menu Selection Phase

In the menu selection phase of this study three menu-based command

selection techniques were compared, namely, Fkey Menus, Chorded Menus, and Mouse Menus. The Fkey Menus technique was designed so that menus were accessed by pressing spatially mapped function keys and commands were selected by pressing a letter that was underlined in the command names. With the Chorded Menus technique, menus were accessed by simultaneously pressing a modifier key and a letter that was underlined in the menu titles. Commands were then selected by pressing a letter that was underlined in the command names. In the case of the Mouse Menus technique, menus were accessed by clicking on the menu title and commands were selected by dragging a highlight cursor over the desired command and then releasing the mouse button. The objective of this phase of the experiment was to glean design principles that could be used to improve the speed, accuracy, and acceptability of menu-based control techniques.

Fkey Menus versus Mouse Menus. In comparing these two menu interaction techniques it was found that the Fkey Menus yielded significantly faster block completion times, menu access times, and command selection times. It was also found that the Fkey Menus technique produced fewer errors than the Mouse Menus technique. In particular, it was found the Fkey Menus technique resulted in relatively fewer substitution and omission errors.

On the average, the block times for the Fkey Menus technique were found to be 21 percent faster than those of the Mouse Menus technique. This translated into an unexpectedly large time savings of 12.6 minutes per task hour in favor of the Fkey Menus technique.

An examination of the menu access times for the two techniques revealed that they did not differ initially, but that the access times for the

Fkey Menus technique quickly became faster than those of the Mouse Menus technique. By the end of the first phase of the experiment, the average menu access time for the Fkey Menus technique was more than 33 percent faster than that of the Mouse Menus technique. Overall, 13 percent of the block time differences between these two techniques were accounted for by the differences in the menu access times.

An examination of the command selection times for the two techniques revealed that, although not initially different, the command selection times for the Fkey Menus technique quickly became faster than those of the Mouse Menus technique. By the end of the menu selection phase, the average command selection time for the Fkey Menus technique was more than 43 percent faster than that of the Mouse Menus technique. Overall, seven percent of the block time differences between the two techniques were accounted for by the differences in the command selection times.

On the last block of the menu selection phase, the Fkey Menus group accessed target menus and selected target commands an average of 1.72 seconds faster than the Mouse Menus group. As was the case with the average block time difference, the size of this difference was unexpectedly large.

As mentioned above, relatively more errors were also committed by the Mouse Menus group. It might be expected that this would have contributed to the relatively longer block completion times observed for this group. However, the additional errors committed by the Mouse Menus group were predominantly substitution and omission errors, which should, in fact, have had a minimal effect on the block time differences. The substitution errors, which were a one-for-one replacement of a non-specified for a specified command, should not have substantially affected the block

completion time differences. The fact that the Mouse Menus group omitted relatively more task-specified commands should have decreased rather than increased the average block completion time differences.

Fkey Menus versus Chorded Menus. In comparing the Fkey Menus and the Chorded Menus techniques it was found that the Fkey Menus technique yielded significantly faster block completion times, menu access times, and command selection times. It was also found that the Fkey Menus technique produced significantly fewer substitution errors than the Chorded Menus technique.

The block times for the Fkey Menus technique were found to be an average of 23 percent faster than those of the Chorded Menus technique. This translated into an average time savings of 13.8 minutes per task hour in favor of the Fkey Menus technique.

An examination of menu access time differences for the two techniques revealed that the menu access times for the Fkey Menus technique were initially 34 percent faster than those of the Chorded Menus technique. By the end of this phase of the experiment the relative difference was 35 percent in favor of the Fkey Menus technique. Overall, 20 percent of the block time differences between the two techniques were accounted for by the differences in the menu access times.

As predicted, there was no difference between the command selection times for these two techniques. This finding was expected, given that the two techniques handled command selection in exactly the same fashion (i.e., the subject pressed a key corresponding to an underlined letter in the command name).

On the last block of this phase of the experiment, the Fkey Menus group accessed target menus and selected target commands an average of

1.23 seconds faster than the Chorded Menus group. Again, the size of the observed difference was surprising.

Also, as mentioned above, the Chorded Menus technique produced more substitution errors in comparison with the Fkey Menus technique. Again, however, errors of this type should not have had a major effect on the overall block time differences.

Chorded Menus versus Mouse Menus. In comparing the Chorded Menus and Mouse Menus techniques it was found that: the two techniques did not differ in terms of block completion times; the menu access times for the Mouse Menus technique were initially faster than those of the Chorded Menus technique, but this difference soon disappeared; and the command selection times for the Chorded Menus technique were faster than those of the Mouse Menus technique. It was further found that the Mouse Menus technique produced more omission errors.

An examination of the command selection times for these two techniques revealed that, although initial command selection times were not significantly different, the command selection times for the Chorded Menus technique quickly became faster than those of the Mouse Menus technique. By the end of this phase of the experiment, the average command selection time for the Chorded Menus technique was more than 36 percent faster than that of the Mouse Menus technique.

Finally, as mentioned previously, the Mouse Menus group omitted a relatively larger proportion of the commands that were specified in the task materials. Again, however, errors of this type should not have had a major effect on the overall block time differences.

Subjective evaluations. As a whole, the subjective responses did not favor any one condition in particular. It was found, however, that relative

to the Chorded Menus group, the Fkey Menus groups felt it was easier to remember the location of commands in the menu structure. It was also found that, relative to the Fkey Menus group, the Mouse Menus group felt that the command selection process was more invigorating.

Summary and conclusions. Based on experimental comparisons of keyboard and mouse-based techniques for selecting command from menus, Karat, McDonald, and Anderson (1986) concluded that "...less 'natural' devices such as keyboards, can in some circumstances be preferred and lead to better performance than more 'natural' pointing devices such as mice" (p. 87). The results of the present investigation clearly support this assertion.

The results of the menu selection phase of the experiment indicated that the Fkey Menus technique was superior to the Mouse and Chorded Menus techniques in terms of both speed and accuracy. It was found that, relative to pointing and clicking with the mouse or pressing a chorded key/letter key sequence, menus were accessed much faster with spatially mapped function keys. It was also found that commands were selected from displayed menus much faster by pressing compatible letter keys as opposed to dragging a cursor to target items and releasing the mouse button to select the items. Finally, it was found that, relative to the other techniques tested (i.e., Mouse Menus and Chorded Menus) the technique based on spatially mapped menu access keys and semantically mapped command selector keys (i.e., the Fkey Menus technique) produced relatively faster combined access and selection times, faster block completion times, and fewer errors.

The error differences that were observed were, in fact, limited to substitution and omission errors, which should not have had a direct

impact on the timing differences. However, the fact that the Mouse Menus group omitted relatively more commands than the other two groups does suggest a possible explanation for at least a portion of the timing differences. In particular, it suggests that the Mouse Menus group tended to "lose their place" in the task materials. If this occurred frequently, it would presumably have increased the average time that these subjects spent reorienting to the task materials, thereby producing a general increase in block completion times, as well as, a possible increase in menu access and command selection times.

Assuming that the Mouse Menus technique did, in fact, produce a disorientation effect relative to the task materials, it may also have caused subjects to 'lose their place' in the on-screen document as well. Again, if this occurred frequently it presumably would also have increased block completion times by increasing the average amount of time the subjects spent trying to find or re-find a specific point or section within the on-screen document.

The subjective data also suggest a possible explanation for some of the error and timing differences. It was found that, relative to the Fkey Menus group, the Chorded Menus group felt they had more trouble remembering the location of commands within the menu structure. If this was indeed the case, it would stand to reason that it would have had a direct impact on the menu access and command selection times as well as the number of errors that were committed by this group. Such difficulties could account for the relatively poor initial performance of this group, as well as some of the long term performance differences that were observed.

The above mentioned difficulties may also have caused the Chorded Menus group to spend more time trying to remember the steps required to

perform a given editing task before actually trying to carry them out. This sort of pre-planning activity would not have been reflected in the menu access or command selection times but would have contributed to the overall block completion times for this group.

The three techniques may also have varied in terms of the basic stimulus-response compatibility between the tasks and task materials and the steps required to access target menus and make command selections. The longer menu access times for the Mouse and Chorded Menus groups suggests that it took these groups longer to translate the intention to access a given menu into the actions required to have it displayed. Furthermore, the longer command selection times that were observed for the Mouse Menus group suggest that it took longer for this group to translate the intention to select a command from a displayed menu into the actions required to make the selection.

It may be that some of the timing differences were simply due to differences in motor requirements for the three techniques. However, the work of Card, Moran, and Newell (1983) suggests that these differences would only account for a small portion of the differences observed in this study.

Finally, the menu selection sequences used in either or both of the keyboard-based techniques may have begun to function at some point like a command language. In other words, after some period of time the subjects in these groups may have begun to make command selections with these keyboard sequences without looking at the menu. While this may have occurred to a certain extent, it does not appear to account for many of the present findings. For example, the performance advantages for the Fkey Menus technique were apparent in the early stages of the experiment -

presumably before the subjects had time to commit many of the selection sequences to memory.

Irrespective of the cause, the fact remains that within the context of a relatively face valid text editing task the Fkey Menus produced faster performance times and fewer errors compared to the traditional Mouse Menus technique or the widely provided Chorded Menus technique. The only finding in favor of traditional point/click/drag method for interacting with command bar menus was that the Mouse Menus subjects found it to be relatively more "invigorating" than the Fkey Menus subjects found the function key/letter key method. However, compared to the Fkey Menus group, there was no other indication that the Mouse Menus group reacted more favorably to the technique they used.

In conclusion, the findings of this phase of the experiment indicate that the best overall design alternative for interacting with command bar menus consists of spatially mapped keys for accessing menus and semantically mapped keys for selecting commands.

Menu Bypass Phase

In the menu bypass phase of the experiment, two bypass coding structures (Fkey Codes and Chorded Codes) were crossed with two menu input devices (Keyboard Menus and Mouse Menus). In the case of Fkey Codes, spatially mapped function keys were used to designate the menus in which the target command was located. These mappings served as the first level in the coding structure. Compatible letter codes were then used to designate specific commands. In the case of Chorded Codes, chorded key/letter key pairings served as the first level in the hierarchical coding

structure. Compatible letter codes were again used to designate specific commands.

The menu input device factor was manipulated so as to vary the continuity between the menu selection technique the subjects had used in the menu selection phase of the experiment and the bypass coding technique they used in the menu bypass phase. In the case of Keyboard Menus, subjects used the same key sequences to make bypass code-based command selections that they had previously used to make menu-based command selections. Thus, there was a high degree of continuity between the menu- and bypass code-based styles of control. In the case of Mouse Menus, subjects had previously used a mouse to select commands from menus, and had not used the key sequences that functioned as the bypass codes. Thus, there was a low degree of continuity between the control technique they used in the menu selection phase of the experiment and the bypass technique they were asked to use in the menu bypass phase.

The experimental conditions resulting from the crossing of structure and device thus varied in terms of the degree of continuity between menu selection technique and menu bypass technique, and also differed in terms of the type of mnemonic used to generate the first portion of the bypass codes (i.e., spatially mapped function keys versus lexically meaningful letter keys). The objective for this phase of the experiment was to glean design principles that could be used to ease the transition from menu to code usage and to improve the usability and memorability of bypass codes.

Fkey Code versus Chorded Codes. In comparing the Fkey Codes and the Chorded Codes structures it was found that Fkey Codes yielded significantly faster block completion and menu designation times. It was

also found that Fkey Codes produced significantly fewer substitution errors than Chorded Codes.

On the average, the block times for Fkey Codes were found to be 13 percent faster than those of Chorded Codes. This translated into an average time savings of 7.8 minutes per task hour in favor of Fkey Codes.

An examination of menu designation time differences revealed that the menu designation times for Fkey Codes were initially 23 percent faster than those of Chorded Codes. By the end of the menu bypass phase of the experiment the relative difference was 21 percent. Overall, 21 percent of the block time differences between the two coding structures were accounted for by the differences in the menu access times.

As predicted, there was no difference between the command selection times for these two bypass coding structures. This finding was expected given that these two structures handled command selection in exactly the same fashion (i.e., the subject pressed a compatible letter key that denoted the desired command).

Furthermore, as mentioned above, Chorded Codes produced more substitution errors in comparison with Fkey Codes. However, errors of this type should not have had a major effect on the overall block time differences.

Keyboard Menus versus Mouse Menus. Comparisons of the Keyboard Menus and Mouse Menus conditions revealed that Keyboard Menus yielded block completion times that were initially faster than those of Mouse Menus. However, this difference disappeared by the end of the third block of menu bypass trials. The command designation times followed a similar pattern. They were initially faster for the Keyboard Menus, however, the difference disappeared by the end of the third block of menu bypass trials.

It was further discovered that Keyboard Menus produced significantly fewer substitution errors than Mouse Menus. Finally, it was found that the effect of menu input device on menu designation times, omission errors, and errors overall was affected by the bypass coding structure that was used.

An examination of the menu designation times revealed that they were initially 19 percent faster for Keyboard Menus in comparison with Mouse Menus. However, as mentioned above, the benefit of having used Keyboard Menus was relatively short lived (for three trial blocks).

An examination of the command designation times for these two menu input devices revealed that the command designation times for Keyboard Menus were initially 28 percent faster than those for Mouse Menus. Again, however, the benefit of having used Keyboard Menus was relatively short lived (for three trial blocks). Overall, six percent of the block time differences between the input device conditions were accounted for by the differences in the command designation times.

Also, as previously mentioned, the Keyboard Menus groups committed significantly fewer substitution errors than the Mouse Menus groups. The total number of substitution errors committed by the Keyboard Menus groups was less than 23 percent of the total for the Mouse Menus groups.

The menu designation times for the Keyboard and Mouse Menus groups suggest that when Chorded Codes were used to bypass menus the benefit of having used Keyboard Menus lasted only one block. The weak impact of menu input device on performance with Chorded Codes suggests one of two possibilities. Either there was little positive transfer of learning from the first to the second phase of the experiment or the Chorded Codes

were extremely easy to learn and use. A review of the other pertinent timing and error data suggests that the former is true.

An examination of the menu designation times for the Keyboard and Mouse Menus conditions revealed that when Fkey Codes were used the benefit of having used Keyboard Menus lasted throughout the second phase of the experiment. This effect suggests there were strong positive transfer of learning effects for the Keyboard Menus/Fkey Codes group. In general, the other timing and error data suggest that this was the case.

Bypass coding structure and menu input device were also found to be related in terms of the number of Extra Action and Omission errors that were committed. The Chorded Codes/Mouse Menus group committed far more Extra Action errors relative to the Fkey Codes/Mouse Menus group. The same trend held for the Omission errors. In addition, it was found that the Chorded Codes/Mouse Menus group committed far more errors overall and far more Omission errors compared to the Fkey Codes/Mouse Menus group.

Subjective evaluations. The responses to the forced-choice questionnaire administered after the second phase of the experiment indicated that subjects liked selecting commands with the bypass codes more than with menus. Subjects also rated bypass codes as a more convenient and a more natural feeling method of command selection. Furthermore, subjects indicated that they felt they selected commands faster with the bypass codes and that they finished the editing tasks faster when using bypass codes. On the other hand, subjects indicated that they felt it was easier to learn how to select commands with menus and that using menus did not require as much concentration as using bypass codes.

Finally, subjects indicated that they felt less anxious when selecting commands from menus than when selecting commands with bypass codes.

When given the choice of having either menus, bypass codes, or both, subjects expressed a general preference for having both styles of control. In particular, they felt that software which offered both styles of control would be easier to learn, easier to use, and less frustrating than software that provided only one style of control.

Summary and conclusions. In general, the results of this phase of the experiment suggest that bypass codes consisting of spatially mapped menu designator keys and semantically mapped command designator keys (i.e., Fkey Codes) were easier to use and more memorable than codes based on semantically mapped menu designator keys and semantically mapped command designator keys (Chorded Codes). The results of this phase of the experiment also suggest that having a high degree of continuity of action between the menu-based and bypass code-based styles of control had only a short lived positive effect on performance. However, the benefits of high intra-style continuity still represent a potential advantage for the novice or intermittent user.

The above mentioned conclusions are supported by the following findings: 1) spatially mapped function keys (i.e., Fkey Codes) consistently produced faster menu designation times compared to semantically mapped chorded key/letter key sequences (i.e., Chorded Codes); 2) the Fkey Codes groups completed the task blocks faster than the Chorded Codes groups; 3) fewer substitution errors were committed with the Fkey Codes than were committed with the Chorded Codes; 4) the Fkey Codes group that had not used the bypass code sequences prior to this phase committed fewer Extra Action errors than did the Chorded Codes group that had not previously

used the bypass codes; 5) the Fkey Codes group that had not used the bypass code sequences prior to this phase committed fewer Omission errors than did the Chorded Codes group that had not previously used the bypass codes; and 6) the block time and command selection time advantages for having had experience using the codes to make menu-based command selections disappeared by the end of the third block of menu bypass trials.

It was also found that the subjective reactions to the menu and command language-based styles of control were somewhat consistent with the lore and conventional wisdom put forth in the literature (see Tables 1-6). The subjects in this study felt that menu-based styles of control were easier to learn, required less concentration to use, and made them feel less anxious. However, they thought that command languages (i.e., the bypass codes) felt more natural, were more convenient to use, and lead to faster task times and better task performance. The subjects of this study, who were quasi-expert users by the time the comparison questionnaire was administered, also responded that, "Overall the method for selecting commands that I liked the most was ... " the command language (i.e., the bypass code-based style).

In conclusion, the findings of this phase of the experiment indicate that the best overall design alternative for bypassing command bar menus consists of spatially mapped keys for designating menus and semantically mapped keys for designating commands. The findings further indicate that a menu-based style of control should be provided with this coding strategy to facilitate user acceptance.

Future Research

In future studies, the generalizability of the present findings to other

types of software applications (e.g., spreadsheets, graphics packages, etc.) should be investigated. The generalizability of the present findings to other input devices (e.g., speech recognizers, trackballs, touchscreens, etc.) and other menu configurations (e.g., pop-up menus, cascading menus, etc.) should also be studied.

In future studies, retrospective verbal protocol and eye tracking measures might be used to clarify the underlying reasons for selection errors and the underlying reasons for differences in performance times for different control strategies. Eye tracking and verbal protocol measures might also be used to gain insight into, and ascertain differences in, the cognitive strategies and visual search behaviors that subjects employ as they attempt to find and relate specific points or areas within task materials, specific points or areas within the on-screen work space, and specific points and areas within the command bar and command bar menus.

REFERENCES

Antin, J. F. (1988). An empirical comparison of menu selection, command entry and combined modes of computer control. Behavior & Information Technology, 7(2), 173-182.

Badre, A. N. (1984). Designing transitionality into the user-computer interface. In G. Salvendy (Ed.), Human-computer interaction (pp. 27-34). Amsterdam: Elsevier Science Publishers.

Bayerl, J. P., Millen, D. R., & Lewis, S. H. (1988). Consistent layout of function keys and screen labels speeds user responses. Proceedings of the Human Factors Society 32nd Annual Meeting, 344-346.

Bertino, E. (1985). Design issues in interactive user interfaces. Interfaces in Computing, 3, 37-53.

Bosser, T. (1987). Learning in man-computer interaction (ESPRIT/HUFIT Project No. 385 Vol. 1). New York: Springer-Verlag.

Card, S. K. (1984). Visual search of computer command menus. In H. Burma & D. Bouwhuis (Eds.), Attention and performance: Vol. 10, control of language processes (pp. 97-108). Hillsdale, NJ: Lawrence Erlbaum Associates.

Card, S., Moran, T., & Newell, A. (1983). The psychology of human-computer interaction. Hillsdale, N. J.: Erlbaum.

Davies, S., Lambert, A. , & Findlay, J. (1989). The effects of the availability of menu information during command learning in a word processing application. Behavior and Information Technology, 2, 135-44.

Eason, K. D., & Damodaran, L. (1981). The needs of the commercial user. In M. J. Commbs, & J. L. Alty (Eds.), Computing skills and the user interface (pp. 115-139). London: Academic Press.

Foley, J. D., Wallace, V. L., & Chan, P. (1984). The human factors of computer graphics interaction techniques. IEEE CG&A, pp. 13-48.

Gilfoil, D. M. (1982). Warming up to computers: A study of cognitive and affective interaction over time. Proceedings of CHI'82 Human Factors in Computer Systems , 245-250.

Green, T. R. G., & Payne, S. J. (1984). Organization and learnability of computer languages. International Journal of Man-Machine Studies, 21, 7-18.

Hall, P. A. V. (1982). Computer dialogues for many levels of competence. In J. A. Larson (Ed.), End user facilities in the 1980's, Computer Society Press.

Hammond, N., & Barnard, P. (1984). Dialogue design: Characteristics of user knowledge. In A. Monk, (Ed.), Fundamentals of human-computer interaction (pp. 127-164), London: Academic Press.

Heffler, M. J. (1981). A human-computer interface that provides access to a diverse user community. Proceedings of the 14th Hawaii International Conference on System Sciences, 2, 601-610.

Hutchins, E. L., Hollan, J. D., & Norman, D. A. (1985). Direct Manipulation Interfaces. In D. A. Norman, & S. W. Draper (Eds.), User centered system design: New perspectives in human-computer interaction (pp. 311-338), New York: Ablix Publishing.

Karat, J., McDonald, J. E., Anderson, M. (1986). A comparison of menu selection techniques: Touch panel, mouse and keyboard. International Journal of Man-Machine Studies, 25, 73-78.

Larson, J. A. (1982). End User Facilities in the 1980's (pp. 1-3, 439-441), New York: Computer Society Press, .

Laverson, A., Norman, K., & Shneiderman, B. (1987). An evaluation of jump-ahead techniques in menu selection. Behavior and Information Technology, 6(2), 97-108.

Liebelt, L. S., McDonald, J. E., Stone, J. D., & Karat, J. (October, 1982). The effect of organization on learning menu access. Proceedings of the 26th Annual Meeting of the Human Factors Society, 546-550.

Maguire, M. (1982). An evaluation of published recommendations on the design of man-computer dialogues. International Journal of Man-Machine Studies, 16, 237-261.

Margono, S., & Shneiderman, B. (1985). A study of file manipulation by novices using commands vs. direct manipulation. Proceedings of the 26th Annual Technical Symposium, Washington, DC Chapter of ACM, 154-159.

Mount, F. E., Rudisill, M., & Schulze, L. J. H., (1984). Cursor controller comparison part A phase I. Human Factors Engineering Section (NASA No. 19974). Lockheed Engineering and Management Services Company Inc. .

Nielsen, J. (1987). A user interface case study of the macintosh. In G. Salvendy (Ed.), Cognitive engineering in the design of human-computer interaction and expert systems (pp. 241-248). Amsterdam: Elsevier Science Publishers B. V.

Norman, D. A. (1984). Stages and levels in human-machine interaction. International Journal of Man-Machine Studies, 21, 365-375.

Norman, D. A. (1983). Design principles for human-computer interfaces. Proceedings of CHI'83 Human Factors in Computing Systems, 1-10.

Norman, D. A. (1981). Learning and Memory. San Francisco: W. H. Freeman and Company.

Paap, K. R., & Roske-Hofstrand, R. J. (1988). Design of menus. In M. Helander (Ed.), Handbook of human-computer interaction, (pp. 205-233). Elsevier Science Publishers: New York.

Perlman, G. (1985). Making the right choices with menus. In B. Shackel (Ed.), Proceedings IFIP INTERACT'84 Human-Computer Interaction, 317-321.

Perlman, G. & Sherwin, L. C. (1988). Designing menu display format to match input device format. SIGCHI Bulletin, 20(2), 78-82.

Ramsey, H. R. & Atwood, M. E. (1979, September). Human Factors in Computer Systems: A Review of the Literature (Technical Report SAI-79-111-DEN). Englewood, Colorado: Science Applications, Inc. (NTIS No. AD A075 679).

Shneiderman, B. (1988). We can design better user interfaces: A review of human-computer interaction styles. Ergonomics, 31, 699-710.

Shneiderman, B. (1987). Designing the user interface. Reading, MA: Addison-Wesley.

Shneiderman, B. (1986). Designing menu selection systems. Journal of the American Society for Information Sciences, 31, 57-70.

Shneiderman, B. (1983). Direct manipulation: A step beyond programming languages. IEEE Computer, 16(8), 57-69.

Shneiderman, B. (1982). The future of interactive systems and the emergence of direct manipulation. Behavior and Information Technology, 1, 237-256.

Shneiderman, B. (1980). Software Psychology - Human Factors in Computers and Information Systems, Winthrop Publishers, Inc.

Smith, S. L., & Mosier, J. N. (1986). Guidelines for designing user interface software (Technical Report ESD-TR-86-278, USAF). Electronic Systems Division, Hanscom Air Force Base, MA (NTIS No. ADA177198).

Streitz, N. A. (1987). Cognitive compatibility as a central issue in human-computer interaction: Theoretical framework and empirical findings. In G. Salvendy (Ed.), Cognitive engineering in the design of human-computer interaction and expert systems (pp. 75-82). Amsterdam: Elsevier Science Publishers B. V.

Streitz, N. A., Spijkers, W. A. C., & van Duren, L. L. (1987). From novice to expert user: A transfer of learning experiment on different interaction modes. Proceedings IFIP INTERACT'87 Human-Computer Interaction (pp. 841-846). New York: Elsevier Science.

Streitz, N., Lieser, A., & Wolters, A. (1989). The combined effects of metaphor worlds and dialogue modes in human-computer interaction. In F. Lkix, N. A. Streitz, Y. Waren, and H Wandke (Eds.), Man-computer interaction (pp. 75-88). North-Holland: Elsevier Science.

Taylor, R. A. (1986). Using multiple dialog modes in a user-system interface to accommodate different levels of user experience: An experimental study (Dissertation University of Texas at Austin). Dissertation Abstracts International, , .

Walker, N. & Olson, J. R. (1988). Designing keybindings to be easy to learn and resistant to forgetting even when the set of commands is large. Proceedings of CHI'88 Human Factors in Computing Systems, 201-206.

Whiteside, J., Jones, S., Levy, P. S., & Wixon, D. N. (1985). User performance with command, menu, and iconic interfaces. Proceedings of CHI' 85 Human Factors in Computing Systems, 185-191.

Ziegler, J. E. & Fahnrich, K. P. (1988). Direct manipulation. In M. Helander (Ed.), Handbook of human-computer interaction, (pp. 123-132). Elsevier Science Publishers: New York.

Zipf, G. K. (1949). Human behavior and the principle of least effort. Cambridge, Mass.: Addison-Wesley.

Appendix A: Instructions for Use of Fkey Menus

PLEASE NOTE

Copyrighted materials in this document have not been filmed at the request of the author. They are available for consultation, however, in the author's university library.

175-214

University Microfilms International

Section 1.0

BUSINESS ENVIRONMENT

ALIGN CENTER

Investor considerations

- Free enterprise and free-trade systems.
- Absence of bureaucratic interference from government.
- No policy of nationalization.
- Highly motivated, educated and skilled work force.
- Low unemployment.
- Overall, tariffs on imports are low.

Industrial climate CUT

The economy is increasingly becoming more service-oriented, with lesser emphasis on manufacturing, as evidenced by their relative share of gross national product (GNP)--*see <Service Industries> in Chapter 1.* The high technology industry is experiencing unprecedented growth that is expected to continue, due in large part to a large, mobile and well-educated work force, as well as an economic system that encourages and rewards the entrepreneur.

The American government and most of the population strongly support the free enterprise system and the free-trade system. However, there is growing sentiment for protectionism in certain quarters, based on the perception that the business and trade practices followed by foreign competitors and governments are "unfair" or "nonreciprocal."

PASTE

Framework of industry

The U.S. economy is broad-based, encompassing almost every type of industry imaginable. It is a mixed economy consisting of upward of

1

ten million small businesses operated through various forms of business entities. However, the share of total assets owned and national product produced is shifting from "small business" to "big business." To put matters in perspective, the dominance of big business is confined mainly to the manufacturing and public utility sectors. In manufacturing, less than 200 companies produce more than one-half of all manufacturing output. In transportation, communications, electricity, and gas, 120 companies produce three-quarters of the output. On the other hand, in the construction, wholesale and retail trade, and the services industries, the big business proportion of output is small.

Most businesses in the United States are owned by a family or a partnership. This is typically true of farms, most retail and service establishments, and even many small manufacturing concerns.

Aims of government policy CUT

The current administration has attempted to stimulate business expansion by stabilizing interest rates and inflation. The main thrust of government ~~economic and~~ fiscal policy is to promote economic growth. PASTE Inflation declined from 12.4 percent in 1981 to 1.1 percent in 1986 and the prime interest rate dropped from a high of 21 percent in 1980 to 7.5 percent in early 1987. There have been other attempts to increase business expansion through deregulation and export promotion.

Deficit reduction, unemployment and defense spending have become

major issues for both the President and Congress. The budget deficit reached a record $220.7 billion in 1986, with a projected fiscal 1987 deficit of $173.2 billion. For unemployment trends, *see Chapter 1 <General description> of the economy.* The President has requested that approximately 28 percent of the federal budget of $1,015.6 billion be appropriated for defense in 1987.

Trend toward nationalization/privatisation: The federal, state and local governments have little direct involvement in manufacturing and services. U.S. laws, the policies of the administration and Congress, and public sentiment all oppose government ownership of business. However, public acceptance of official intervention to rescue failing companies that provide essential services-such as railroads, utilities or defense equipment-is growing. On three separate occasions, the federal government has stepped in to assist large corporations on the brink of financial collapse. It provided loan guarantees for Lockheed Aircraft Corp. in 1971, for Chrysler Corp, in 1979-80, and for Continental Bank in 1984.

Regional/special industry development: Most states actively promote new investment, offering assistance in obtaining financing and advisory services. For more details, *see Chapter 4 <Investment incentives>.*

Free-trade zones: The United States has over 100 Foreign Trade Zones within the country. For details, *see Chapter 4 <Investment Incentives>.*

Financial services: U.S. government policy is neutral regarding special financial services for industry. For information on available financial services, *see Chapter 7 <Banking and Finance>*.

Labor/management relations

A skilled labor force is readily available in the United States, as are well-educated managerial and supervisory personnel. Because of their size and influence, labor unions are an important economic force in the country. However, in recent years, labor union membership has fallen dramatically. High labor costs, rather than fringe benefits, characterize labor costs in the United States. For more details, *see Chapter 10 <Labor Relations and Social Security>*.

Overseas trade relations

Membership in trade blocs: The United States does not belong to any trading bloc, but is a member of the General Agreement on Tariffs and Trade (GATT).

Export: The United States encourages the export of goods overseas by offering a number of export-linked incentives. Export credit insurance is also available. For more details, *see Chapter 4 <Investment Incentives>*.

BOLD

Trade barriers: The importation of certain articles is either prohibited or restricted. Restrictions include imports subject to compulsory licensing, imports limited by absolute quota and imports controlled by

raising tariffs after quotas have been filled (*tariff-rate import quotas*). Some of the rules limit entry to certain ports; restrict routing, storage or use; or require treatment, labeling or processing as a condition of release from customs.

The United States has a wide array of legal measures at its disposal that can be used to protect domestic industry from injurous imports. The principal laws are those governing antidumping and countervailing duties, market disruption caused by imports from communist countries and procedures for retaliation against unfair trade practices. Proposed legislation would strengthen current laws and provide for mandatory import surcharges under certain conditions for goods coming into the United States from countries maintaining trade surpluses.

The overall tariff burden is not high. The average tariff on industrial products is about 4.3 percent for 1987. Most tariffs are levied ad valorem, but a few are still levied on a specific basis. The tariff negotiations resulted in a change from specific to ad valorem levies on a wide range of raw and semiprocessed materials.

For additional information on importing, *see Chapter 8 <Exporting to the United States>*.

CLICK OPEN SECTION 2.0

Section 2.0

EXPORTING TO THE UNITED STATES

→ ALIGN CENTER

Tips for exporters

- U.S. Customs will provide binding rulings, prior to exportation, on likely duty rate and value.
- Exporters should seek advice of knowledgeable U.S. customs brokers or trade consultants prior to shipment.
- Strict enforcement of antidumping and countervailing duty laws.
- Related-party sales present unique problems with regard to customs valuation and U.S. tax laws.
- Duty deferrals available by using Customs bonded warehouses and foreign trade zones (free ports) PRINT
- Reduced duty rates for developing countries, certain geographical regions.
- Certain developing countries and countries in specific geographical regions often entitled to reduced rates of duty.

Introduction

More and more companies are becoming involved in exporting to the United States. However, they are finding that these activities, while profitable, are also posing considerable challenges: confusing classification codes; burdensome procedural demands; and complex and frequently changing tax and legal requirements. In order to gain a deeper under standing of the U.S. import process and what, if any, actions can be taken by exporters to avoid unnecessary risks and difficulties, it may prove useful to strategically plan the initial entry into the market with the above-noted points in mind.

Import process

The import process consists of the following essential elements:

JUSTIFY

1. A sale of merchandise from a foreign exporter to a U.S. importer takes place.
2. When the merchandise reaches a port of entry, documents must be filed with Customs in order toassign a tentative value and tariff classification to the product.
3. Customs examines the merchandise to see whether there are any restrictions on importation (such as quotas); to ensure compliance with Customs and other agency regulations (such as proper marking or other means of identification); and to uncover any prohibited items (such as contraband or counterfeit goods).
4. If everything is in order, the customs entry can be "liquidated"--that is, the customs duty is determined and paid, and the merchandise is released. If there are restrictions on the merchandise or problems with documentation, Customs may hold the goods until the situation is resolved.

Import restrictions

The importation of certain classes of merchandise may be prohibited or restricted to protect the economy and security of the United States, to safeguard consumer health and well-being, and to preserve domestic plant and animal life. Some commodities are also subject to an import quota or a restraint under bilateral trade agreements and arrangements.

Many of these prohibitions and restrictions on importations are subject, in addition to Customs requirements, to the laws and regulations administered by other U.S. government agencies with which Customs cooperates in enforcement. These laws and

regulations may, for example, prohibit entry; limit entry to certain ports; restrict routing, storage or use; or require treatment, labeling or processing as a condition of release. Customs clearance is given only if these additional requirements are met. This applies to all types of importations, including those made by mail and those placed in foreign trade zones.

COPY

The foreign exporter should make certain that the U.S. importer has provided proper information to:

(1) permit the submission of necessary information concerning packing, labeling, etc., and

(2) that necessary arrangements have been made by the importer for entry of the merchandise into the United States.

NEW PASTE SELECT ALL BOLD SAVE CLOSE

Prohibited imports: Prohibited imports include certain narcotics; obscene, immoral and seditious matter; merchandise produced by convict or forced labor; and certain other items.

Imports subject to license or permit: Among imports subject to compulsory licensing are arms and ammunition; Krugerrands from South Africa; and products of Cuba, Cambodia, Libya, Nicaragua, North Korea, and Vietnam.

It is impractical to list all restricted articles specifically; however, the following is a short list of commodities that need a license or permit in order to be imported:

Section 2.0

1. Alcoholic beverages.
2. Animals and animal products.
3. Certain drugs.
4. Fruits and nuts.
5. Meat and meat products.
6. Milk, dairy and cheese products.
7. Plants and plant products.
8. Poultry and poultry products.
9. Petroleum and petroleum products.
10. Trademarked articles.
11. Vegetables.

BOLD

Tariff-rate import quotas: Tariff-rate import quotas (i.e., tariffs are raised after the quotas have been filled) are fewer and apply to various goods, such as tuna fish, certain other fish, whole milk, motorcycles, potatoes, and brooms.

Imports limited by absolute quotas: Imports limited by absolute annual quotas include some dairy products, animal feeds, cotton, cotton waste, stainless steel bars, textile articles and wearing apparel, peanuts, sugars, syrups, molasses, cheese, and some beers and wines.

Antidumping and subsidized exports: Exporters should be aware that the United States strictly enforces unfair trade laws, such as the Antidumping Law (sales to the United States at prices lower than sales in the export market) and the Countervailing Duty Law (subsidized exports).

The purpose and effect of these laws are to restrict imports where unfair trade policies are being exercised by foreign corporations and countries. This topic has become one of the leading political and economic issues in the United States.

Import duties

Customs duties: Duty rates may vary depending on each product's country of origin, the type of product and other factors. Items brought into the United States are subject to duty in accordance with their classification in the import tariff schedules. For example, some rate concessions are granted to items from least-developed developing countries. Most products, however, are dutiable at somewhat higher level under the "most favored nation" (MFN) rates--such countries as France, West Germany and Japan. Finally, goods from countries to which these rates have not been extended--such as East Germany, the Soviet Union and Bulgaria--are dutiable at the highest level with full or "statutory" rates.

Overall, the import tariff schedules contain more than 10,000 classifications. Experts estimate that about 60 percent of these classifications may be subject to interpretation; that is, an import

could reasonably fall under more than one tariff classification and, hence, qualify for different duty rates.

Customs valuation: The product description of an import is the primary determinant of its tariff classification. Based on this classification, a duty rate is assessed against the import's value.

Transaction value: The most common yardstick used by Customs to determine that value is the actual sales price, plus certain commissions and other expenses ("transaction value"). However, an exporter must demonstrate that his declared transaction value represents a bona fide business transaction. The exporter can do this by making certain documents available to Customs in order to prove the validity of the transaction. These documents include, but are not limited to, commercial invoices, country of origin certificates, bills of lading, and foreign assembly declarations.

PLAIN TEXT

Customs may make additional adjustments to the transaction value (*for example, adding in selling commissions, royalties and license fees, and rebates to the manufacturer*), particularly in dealings between related parties, such as a foreign parent and its American subsidiary.

Deductive value: Another method that can be used to establish the value of imports is the "deductive value," in which the U.S. resale value is worked back to an ex-factory price. If a product is subject to the use of deductive value, it is valued after resale in the United

States has occurred, with deductions for commissions, profits and general expenses incurred with the sale, as well as the costs of transportation and processing.

PASTE

Computed value: A third method is the "computed value," which simply combines the cost of production plus any profits. A similar benefit as the one for use of the deductive value results from using the product's computed value. With this method, imported products from a related overseas supplier may help establish a lower duty base than the transaction value. General, sales and administrative expenses incurred in the country of production, based on generally accepted accounting principles in that country, can be calculated differently than in the United States. These costs will then determine the value of the product.

COPY

In any case, whether using transaction value, deductive value or computed value, the Customs Service requires supporting evidence and adherence to generally accepted accounting principles.

Foreign trade zones and customs duties: Duties are paid only on those items imported into the United States from foreign trade zones (free ports). When the goods enter U.S. territory, the cost of processing (U.S. labor, overhead and facilities) and profit realized is excluded in determining duty value for Customs appraisal purposes. For more information on these zones, *see <Free-trade zones> in Chapter 4.*

Other taxes: With the exception of liquor and tobacco excise taxes, no other taxes are due by virtue of exportation to the United States. However, once a product is sold within the country, many local jurisdictions collect inventory or sales taxes, which vary from state to state.

Documentation procedures

Imported goods are not legally entered until after the arrival of the shipment within the limits of the port of entry and delivery of the merchandise has been authorized by Customs. This normally is accomplished by the filing of appropriate documents by the importer. (or his agent) SAVE The Customs Service does not notify the exporter of the arrival of the shipment. Imported merchandise not entered through Customs within five days after its arrival or any authorized extension is sent by Customs to a general order warehouse to be held as unclaimed.

To make or file a customs entry, the following documents are generally required:

1. A bill of lading, airway bill or carrier's certificate (naming the consignee for customs purposes) as evidence of the right of the consignee to make entry.
2. A commercial invoice, obtained from the exporter, that shows the value and description of the merchandise should accompany the goods.
3. Entry manifest or Application and Special Permit for Immediate Delivery.
4. Packing lists, if appropriate, and other documents

necessary to determine whether the merchandise may be admitted.

Re-exports

"Drawback" is a partial refund of duties collected on imported items that are subsequently re-exported. For example, if imported products are returned because they do not conform to specifications or are in some way defective, the importer may be eligible for a rebate of up to COPY 99 percent of the original duties paid. Or, if the imported goods are, for whatever reason, exported in essentially the same condition--that is, not substantially modified or exported as part of another product, the importer may also be entitled to a PASTE refund of the original import duties.

Benefits can also be derived when a mixture of imported articles with like domestic raw materials is used to manufacture an entirely new product. When that product is, in turn, exported, the importer may be eligible for a rebate of some percentage of the original import duty.

Although drawback is an incentive to export American products with previously imported foreign parts, drawback claims pose the need for orderly import documentation, available accounting records and an ability to provide an audit trail to Customs in order to be acceptable.

Local representation

Local agent:

Customs consultants and brokers: Customs consultants, with accounting firms or customs brokers, are available to assist exporters and importers with their U.S. importations.

Sales agent: It is not necessary to employ the services of a sales agent if the exporter is communicating directly with his U.S. importer, However, the services of a sales agent are helpful in assisting with the administrative burdens of the import process. In general, there are no problems or restrictions in retaining their services.

COPY PASTE PASTE

Sales subsidiary: The establishment of sales subsidiaries does not cause any inherent difficulties from a U.S. Customs standpoint. However, exporters should be aware that numerous other government agencies should be consulted prior to taking this step.

For liability to taxation on sales of goods to U.S. customers, *see <Imports> in Chapter 16.*

Sources of information

All U.S. embassies maintain a commercial library and, generally, have individuals available who are familiar with most import requirements. Other sources of information outside of the United States include customs experts with the major accounting firms and overseas editions of U.S. newspapers or trade journals.

OPEN
Section 3.0

TAXATION OF INDIVIDUALS

Investor considerations

- Aliens are residents for U.S. tax purposes if considered lawful permanent residents or meet a "substantial presence" test.
- "Dual-status aliens" subject to U.S. tax under both resident and nonresident rules.
- U.S. citizens and residents taxed on worldwide income.
- Nonresident aliens taxed on U.S. source business income at graduated rates.
- Nonresident aliens taxed on U.S. source income not effectively connected with a business at flat rate.
- Certain U.S. source income earned by nonresident aliens tax exempt.
- Employee allowances and tax reimbursements taxable.
- Split-employment contracts for U.S. citizens and residents do not reduce U.S. taxes.
- Business-related and nonbusiness-related deductions available.
- Standard deductions. PASTE
- Exemptions. PASTE
- Relief from double taxation available to U.S. citizens, residents and nonresidents.
- U.S. tax rates depend on "filing status."
- Credits available to U.S. citizens and residents and, to a lesser extent, nonresidents. COPY
- Taxpayers compute tax under both the regular tax system and alternative minimum tax system, paying the larger amount.

Territoriality and residence

U.S. citizens and aliens resident in the United States are subject to the same tax rules; they are ordinarily taxable on their worldwide income, irrespective of source. Nonresident aliens are taxable only on U.S. (and certain foreign) source business income and certain classes of U.S. source nonbusiness income. Business and nonbusiness income are taxed separately, generally in accordance

with the same rules that are applicable to foreign corporations (*see <Separation of business and nonbusiness income> in Chapter 16*).

COPY

An alien will be considered to be a U.S. resident for income tax purposes if the individual:

(1) is a lawful permanent resident of the United States at any time during the calendar year, or

(2) meets the requirements of the "substantial presence" test. This test will be met if an individual, having at least 31 days of U.S. presence in the current year, has 183 or more days of presence in the United States during the current and two preceding years. The two preceding years have special weightings.

NEW PASTE SAVE CLOSE

It is important to note that an alien can be considered a "nonresident" (or nonimmigrant) for immigration purposes and at the same time a "resident" for U.S. tax purposes. However, if an alien is deemed to be a "resident" (or immigrant), he must consider himself a resident for tax purposes as well, or risk losing his immigration status. For a detailed explanation of the taxation of nonresident aliens, see the Price Waterhouse Information Guide "Foreign Nationals in the United States." For a brief explanation of the visa system required for nationals of foreign countries to enter the United States for the purpose of conducting business, *see <Work permits> in Chapter 10.*

<u>Special provisions</u>

Certain exclusions from gross income are available only to nonresident aliens. First, U.S. source gross income from the

operation of ships and/or aircraft is exempt from U.S. taxation if the nonresident is a resident of a foreign country that grants U.S. citizens an equivalent exemption. In other words, the nonresident alien's country of residence must grant U.S. citizens an equivalent exemption with respect to gross income attributable to the operation of ships and/or aircraft within its jurisdiction.

in the United States

Second, compensation received by a nonresident alien who is present as a nonimmigrant for a temporary period under an F visa (student) or J visa (trainee, specialist) is excluded from gross income. However, the exclusion applies only to compensation paid to the nonresident alien by a foreign employer, which includes a foreign office or branch of a U.S. business as well as actual foreign entities.

PRINT

A final gross income exclusion covers amounts received by a nonresident alien in the form of an annuity under a qualified annuity plan, or from a qualified trust administering a pension, profit sharing or stock bonus plan, provided:

1. The annuity is paid in respect of services performed by the alien outside the United States while he or she was a nonresident.

2. At the time the first amount is paid, as an annuity under the plan, to the alien claiming this exclusion, 90 percent or more of the employees or annuitants in the plan are citizens or residents of the United States.

Special provisions also surround the treatment of various types of income, as well as the deductibility of certain expenses and

allowances; these are discussed throughout the remainder of the chapter.

Gross income

BOLD Employee services:

U.S. citizens and resident aliens: U.S. citizens and resident aliens must include in gross income all compensation received (regardless of source, currency paid or residence of payer), including living and housing allowances, tax reimbursements and the fair market value of benefits in kind, such as houses and automobiles. Split-employment contracts to compensate U.S. citizens or resident aliens for work performed in and out of the United States do not reduce the individual's U.S. tax liability since they are subject to taxation on their worldwide income.

Certain items of compensation are excludable from gross income, including:

(1) limited premiums on group term life insurance policies;

(2) meals or lodging furnished for the convenience of the employer; and

(3) benefits received under group legal services plans, educational assistance plans, and accident and health plans.

In addition, citizens and residents may qualify to elect to exclude up to $70,000 of foreign-earned income in 1987. Resident aliens may

only qualify for the exclusion by meeting the physical presence test, which requires that the taxpayer's tax home be in a foreign country and that he or she be present in a foreign country during at least 330 days in a consecutive 12-month period. For further details, refer to the Price Waterhouse Information Guide "U.S. Citizens Abroad." To qualify, a U.S. citizen's tax home must be in a foreign country and the taxpayer must meet either the bona fide resident test or the physical presence test.

Nonresident aliens: A nonresident alien rendering services within the United States for either a U.S. or foreign employer is considered to be engaged in a U.S. business and must include in gross income all compensation received for such services, wherever paid. However, the remuneration is exempt from U.S. tax under the following conditions:

1. The individual is present in the United States for a period not exceeding 90 days in a taxable year.

2. The compensation for the taxpayer's services is not in excess of $3,000, and

3. The services are performed either for a foreign employer not engaged in a U.S. business or for the foreign place of business of a U.S. corporation, partnership or individual.

The compensation may also be exempt by virtue of a tax treaty (*see <Personal services> in Chapter 23*).

Employees of foreign governments: Employees of foreign governments stationed in the United States who are not citizens of the United States are exempt from federal income tax with respect to compensation received for official services performed on behalf of their governments. In order to be eligible for such exemption, the individuals must perform services of a nature similar to services performed by employees of the U.S. government in their countries, and their countries must grant the same exemption to employees working there. Income other than compensation for official services is subject to tax.

COPY

PASTE

Capital gains:

U.S. citizens and resident aliens: Citizens and residents must include capital gains in gross income. The kinds of gains treated as capital in nature are generally the same for both individuals and corporations (*see <Capital gains> in Chapter 15*). A loss of up to $50,000 ($100,000 in the case of a husband and wife filing a joint return) on the sale or exchange of small business stock may be treated as an ordinary, rather than a capital, loss.

The Tax Reform Act of 1986 eliminated the 60 percent capital gains exclusion for 1987 and subsequent years. Formerly when a taxpayer realized a net long-term capital gain (the excess of net long-term capital gain over net short-term capital loss), only 40 percent of the gain was recognized for regular tax purposes. The holding period of the capital asset determines whether the gain or loss is long term or short term. If a capital asset is held more than six months, the gain or

loss is long term. If it is held not longer than six months, the gain or loss is shod term. Beginning in 1987, the full amount of a net long-term capital gain is taxable. However, the Act provides a special transitional adjustment for 1987 whereby a taxpayer's net long-term capital gain is subject to tax at a maximum rate of 28 percent, even though the highest regular tax rate is 38.5 percent. If the taxpayer's marginal regular tax rate is less than 28 percent, the lower rate will be applied to the net long-term capital gain.

Although the capital gain exclusion is repealed, the distinction between capital assets and ordinary assets is preserved, as are the mechanisms for capital gain computation. This is important not only because of the preferable 28 percent maximum tax rate available in 1987 for net long-term capital gain income but also because of the rules applicable to capital losses. Capital losses are allowed only to the extent of capital gains plus $3,000 (see <Deductions> in this chapter). Any capital loss in excess of this amount is carried forward indefinitely until it is fully absorbed.

ITALIC

Capital losses may only be deducted if incurred in business or transactions for profit. Losses on personal-use property are only permitted for casualty or theft losses in excess of $100 per loss, and in excess of 10 percent of the individual's adjusted gross income. Thus, a loss on the sale of a personal residence is not deductible.

Special rules are applicable if a taxpayer sells his principal residence. Specifically, gain on the sale of a principal residence is not taxed if

the taxpayer buys, or builds, another residence and uses it as his or her principal residence within a specified period of time, and the cost of the new residence equals or exceeds the selling price of the old. If the sales price of the old residence exceeds the cost of the new, gain will be recognized to the extent of the difference. If the taxpayer is 55 years of age or older before the date of sale, the taxpayer can exclude up to $125,000 of gain on the sale if he or she owned and used the home as a principal residence for specified periods of time prior to the sale. This $125,000 exemption can be elected only once.

Gains and/or losses from certain like-kind exchanges of business or investment property and from involuntary conversions may not be taxed (*see <Capital gains> in Chapter 15*).

Nonresident aliens: A nonresident alien is taxed on capital gains connected with a U.S. business in the same manner as citizens and residents (see above). A nonresident's net capital loss connected with a U.S. business is allowed to the extent of the net capital gain arising COPY from a U.S. business plus $3,000 of effectively connected PASTE income. Such net capital loss may not be used to offset a net capital gain that is not connected with a U.S. business or any other U.S. income that is not effectively connected.

Such loss is subject to the same carryover rules applicable to U.S. citizens and residents (also see above). If the gains are not connected with a U.S. business, they are taxable only if from U.S. sources and if the nonresident is present in the United States for at

least 183 days during the taxable year; the statutory rate of tax on net gains (whether long-term or short-term) is 30 percent, although a lower treaty rate may apply. Although gains may be taxable under the Internal Revenue Code, they may be exempt under a tax treaty (*see <Capital gains> in Chapter 23*). A nonresident's capital loss that is not connected with a U.S. business is deductible to the extent of the taxpayer's capital gain not connected with a U.S. business, Any excess of such losses over such gains is not carried forward but lost. The collapsible corporation provisions do not apply to a nonresident alien not engaged in a U.S. business (*see below*). Gains from the sale of real property interests are deemed to be connected with a U.S. business and therefore taxable (*see <U.S. real property interests> in Chapter 13*).

Other income:

U.S. citizens and resident aliens: Items included in gross income and thus taxable to a citizen or resident are similar in nature to items included by a corporation (*see <Gross income> in Chapter 15*). These include dividends, interest, rents, and royalties, Also includable are income from a profession or unincorporated business, remuneration for personal services, pensions, annuities, and certain alimony and separate maintenance payments. Resident aliens should consider the possible favorable U.S. tax ramifications that may arise from the rental of their personal residence in their home country. To the extent that expenses (including depreciation) exceeds rental income, such loss may shelter other income from taxation. The term "income" has a very broad meaning and generally may be taken to include all accretion of wealth recognized by a taxpayer, However,

mere appreciation in the value of an asset is not income until it is recognized by sale, exchange or other conversion. In addition, contributions to qualified deferred compensation plans, and the income accumulated thereon, are not taxable to an employee until the year in which distributed or made available to the individual.

UNDERLINE Nonresident aliens: A nonresident alien is taxable on the same items of income as a citizen or resident if the income is connected with a U.S. business. Only limited items of foreign source income are considered to be connected income (*see <Income subject to regular corporate tax rates> in Chapter 16*). Certain kinds of U.S. source investment and other income not connected with a U.S. business are generally subject to a flat rate of tax of 30 percent of gross income, or lower treaty rate (*see <Income subject to flat rate tax> in Chapter 16*). The sourcing rules for income that is not connected with a U.S. business are discussed under *<Geographical source of income> in Chapter 13.* A nonresident alien, like a foreign corporation, may elect to be taxed on U.S. real property nonbusiness income at the regular tax rates rather than at 30 percent on a gross basis.

Exempt income: The Internal Revenue Code specifically provides for certain exclusions from gross income. The most common of these exclusions are as follows:

1. Generally, life insurance proceeds when paid by reason of the death of the insured.
2. Certain interest on obligations of states or political subdivisions thereof, generally known as tax-exempt

municipal bonds.

3. Gifts, bequests and inheritances.

4. Social security benefits, up to a limit.

PLAIN TEXT

Citizens can in certian cases exclude income from U.S. possessions *(other than Puerto Rico, the Virgin Islands and Guam)* received outside the United States. Any individual who is a resident of Puerto Rico for an entire taxable year is generally exempt from U.S. tax on income from Puerto Rico sources.

Closely held companies

S corporation: Certain closely held corporations may elect not to be taxed on their income but, instead, to have their shareholders taxed on such income. This election avoids double taxation, i.e., taxation at the corporate level and at the shareholder level, and allows the shareholders to benefit currently from corporate losses.

When an election to be treated as an S corporation is made, the corporation is not subject to federal income tax, with limited exceptions for certain capital gains and passive income. The shareholders include in their individual income tax returns their pro rata share of the corporation's current taxable income or loss, regardless of whether such income (loss) was actually distributed or retained. In determining the tax of a shareholder, a pro rata share of the corporation's items of income, loss deduction or credit, the separate treatment of which could affect the tax of the shareholder

CUT

and the nonseparately computed income or loss, is recognized. The at risk loss limitation rules discussed below are generally applicable to an S corporation. The amount of loss from an S corporation that is deductible by a shareholder is limited to the shareholder's basis in the stock and the shareholder's loans to the corporation. PASTE

Only a domestic corporation that is not a member of an affiliated group and that has no more than 35 individual shareholders who are residents or citizens may elect to be an S corporation. Estates and certain trusts may also be shareholders.

Personal holding company: A 38.5 percent penalty tax is imposed on certain closely held corporations classified as personal holding companies because of the nature of their income and the ownership of their stock (*see <Personal holding company tax> in Chapter 15*). This tax generally does not apply to a foreign corporation if all of its shares are owned by nonresident aliens.

Foreign personal holding company and controlled foreign corporation: U.S. shareholders (including citizens or residents) of foreign personal holding companies and controlled foreign corporations are taxable on certain undistributed income of such corporations, (*See Chapter 18 <Taxation of foreign operations>* for a general discussion of these rules.) A foreign personal holding company is excluded from the definition of a personal holding company.

Passive foreign investment company: A foreign corporation is a passive foreign investment company (PFIC) if 75 percent or more of its gross income is "passive income" for the tax year, or at least 50 percent of the average value of its assets during the tax year produce, or are held for producing, "passive income." Excess distributions from a PFIC and gains from the disposition of shares in a PFIC are taxed to a U.S. citizen or resident as ordinary income earned ratably over the period the investment was held except in the case of a passive foreign investment company which is a "qualified electing fund." The tax liability is determined by applying the highest statutory tax rate for a particular year to the gain or excess distribution allocated to that year. The tax due is the sum of the amounts of additional tax for each prior year plus interest on the deferred tax liability for prior years plus the current year's tax liability on amounts allocated to the current year and amounts attributable to years in which the company was not a passive foreign investment company. A PFIC is excluded from the definition of a personal holding company.

FIND
passive foreign investment company
FIND NEXT
CLOSE

Collapsible corporation: A shareholder owning more than 5 percent of the stock in a collapsibal corporation may be denied capital gain treatment on the sale or redemption of his or her stock if more than 70 percent of such gain is attributable to property manufactured, constructed, produced, or purchased by the corporation, and the gain is realized within three years following the completion of the manufacture, construction, production, or purchase of the property. The purpose of the collapsibal corporation rules is to prevent shareholders from converting to capital gain income attributable to

such property, which is ordinary in the hands of the corporation before the corporation itself realizes two-thirds of the income to be derived from its sale.

OPEN SECTION 3.1

Section 3.1

ALIGN LEFT ←

Deductions

Deductions for individuals fall into two different categories. Some deductions, generally those of a business nature, are subtracted from gross income in arriving at "adjusted gross income." Others are deducted from adjusted gross income in arriving at "taxable income." Adjusted gross income is of special significance because certain deductions are subject to limitations expressed as a percentage of adjusted gross income (*see below*). In addition, deductions to determine adjusted gross income are allowed even if a blanket deduction is elected (also *see below*). Taxable income is the base on which the tax is computed.

Business:

U.S. citizens and resident aliens: Certain deductions that are generally of a business nature are allowed in arriving at adjusted gross income. The major deductions are as follows:

1. Expenses, deductions and losses in carrying on a business or profession, other than as an employee, subject to the "passive activity" rules and the "at-risk" rules discussed below. (These deductions are similar in nature to those allowed · a corporation; *see <Business expenses> in Chapter 15.*)

2. Expenses attributable to the production of rents or royalties, subject to the "passive activity" rules and the "at-risk" rules discussed below.

3. Losses from the sale or exchange of investments or income-producing property, except that a net loss from the sale of capital assets is only deductible to the extent of $3,000. (Each dollar of short-term and long-

term capital loss may be deducted from ordinary income in full to the $3,000 limit. Any unused loss may be carried forward until exhausted.)

4. Contributions to a qualified retirement plan for the self-employed and for certain retirement savings under certain circumstances. When deductible, both are subject to annual limitations.

5. Certain alimony and separate maintenance payments.

Passive activity limitations: The passive activity rules operate to limit the deductibility of passive activity losses in an effort to prevent certain taxpayers from using "tax shelter" losses to reduce their other taxable income. Portfolio income (dividends, interest and capital gains) is not treated as income from a passive activity and therefore, in general, may not be offset by passive activity losses. Generally, passive activity losses and credits may be used only to the extent of passive activity income. Disallowed losses from a passive activity may be carried forward indefinitely to future tax years. If sufficient passive activity income is not generated in such future years to absorb the losses carried forward, then the losses are allowed in full when the taxpayer disposes of his entire interest in the activity that created such losses.

CUT

PASTE

A taxpayer's activities are considered passive unless he "materially participates" in the business. Activities conducted through a limited partnership are considered passive to limited partners. Any rental business (except hotel operations) is also considered a passive activity. However, if an individual "actively" participates in a rental

real estate activity (which may consist of several properties), losses of up to $25,000 a year from that particular activity may be allowed as a deduction against other nonpassive activity income. This allowance is phased out for taxpayers with adjusted gross income in excess of $100,000, and is generally unavailable to limited partners. The allowance is reduced by 50 percent of each dollar of adjusted gross income in excess of $100,000. Therefore, a taxpayer with $150,000 of adjusted gross income would not be entitled to a current deduction.

The passive activity provisions are effective for tax years beginning after December 31, 1986 (subject to some complicated transitional rules), and apply to individuals, estates, trusts, personal service corporations, and closely held regular corporations. Partnerships and S corporations are not subject to the limits directly since they are treated as conduits and pass their activities through to their owners who may be subject to the limits.

At-risk limitations: The at-risk rules generally operate in a manner to prevent taxpayers from deducting losses (incurred with respect to a business or income-producing activity) in excess of the amount actually at risk. For example, if an individual invests $10,000 in a business, $5,000 of which was obtained through a nonrecourse loan, deductible losses are limited to $5,000. These at-risk limitation rules apply to a lesser extent to real property activities conducted by a taxpayer.

A nonresident alien is entitled to business-related deductions

connected with his or her U.S. business income, subject to the same limitations applicable to a U.S. citizen and resident alien.

CUT

Nonbusiness: A blanket deduction, called the standard deduction, is available and may be used where it exceeds the total amount of itemized deductions otherwise allowed. Certain deductions that are not generally of a business nature and several that are may be subtracted from adjusted gross income in arriving at taxable income. PASTE The amount of the standard deduction varies, depending on the taxpayer's filing status (see "Filing status" below) and will be indexed for inflation beginning in 1989. The standard deduction amounts shown in <table I>.

ALIGN RIGHT <Table I>

Standard deduction Filing status

	1987 $	1988 $
Married/joint return	3,760	5,000
Head of household	2,540	4,400
Single	2,540	3,000
Married/separate return	1,880	2,500

An additional standard deduction amount of $600 for each elderly (65 or older) or blind individual filing a joint return is allowed. For single taxpayers, the additional amount is $750.

A taxpayer may elect to deduct the items listed below

(itemized deductions) instead of the standard deduction:

1. Certain expenses incurred in moving to a new job location.

2. Certain expenses of an employee, including business travel expenses while away from home, business transportation expenses *(but not expenses of commuting),* and expenses of outside salesmen. These expenses are combined with expenses of producing investment income (see item 9 below) and certain other deductions (see item 8 below) and may be deducted only to the extent they exceed in aggregate 2 percent of the taxpayer's adjusted gross income. Business travel and entertainment expenses includable in the amount subject to the 2 percent limitation are first subject to the 80 percent limitation (*see <Travel and entertainment> in Chapter 15*).

3. Interest expense is classified in several different ways and is deductible, depending on its classification, as follows:
   a. Personal interest-Consists of interest incurred for personal purposes, such as for credit cards and automobiles used for personal purposes. The deduction for this type as interest is being phased out over a four-year period, such that 65 percent is deductible in 1987, with lesser percentages applicable in later years.

   b. Mortgage interest-Consists of interest on the taxpayer's principal and second homes. The interest deduction, in general, is limited on loans up to the cost of the homes (purchase price plus improvements). Additional interest is deductible as long as the debt is incurred for educational or medical expenses.

   c. Investment interest-Consists of interest incurred on loans to acquire assets generating investment

income, such as stock. Interest is deductible to the extent of "net investment income."

d. Passive activity interest-Consists of interest incurred in connection with a taxpayer's operations that are subject to the "passive activity" rules discussed above. In general, this interest is deductible, subject to the passive activity limitations.

e. Trade or business interest-Consists of interest incurred on loans in connection with a taxpayer's trade or business activities that are not categorized as "passive activities" *(see above)*. Such interest is fully deductible.

4. Certain state and local taxes, e.g., income, property and certain foreign taxes. Sales tax is not deductible in general but in certain instances may be capitalized and deducted through depreciation.

5. Contributions to U.S. charities, subject to certain limitations.

6. Medical expenses (including medical insurance premiums), subject to certain limits.

7. Casualty or theft losses on nonbusiness property, subject to certain limitations.

8. Expenses incurred in connection with the determination, collection or refund of any tax (such as tax preparation and consulting fees), subject to the 2 percent limitation discussed under item 2 above. COPY

9. Other expenses incurred for the production of income but not connected with a business (such as investment advisor fees and trustee fees), subject to the 2 percent limitation. PASTE

Appendix G: Bipolar Rating Scales

Section 3.1

Nonresident aliens: Nonresident aliens may only deduct the expenses listed in items 1, 2, 3(d), 3(e), 5, and 7 above to the extent they are sourced to the United States. These deductions are subject to the same limitations applicable to U.S citizens (and residents). A taxpayer who claims nonresident status for any part of the taxable year may not claim the standard deduction. No deductions are allowed with respect to income that is not connected with a U.S. business.

QUIT

36

Appendix G: Bipolar Rating Scales

Subject #: ______

Software Evaluation

We would like you to rate the software you just used by completing the following statements. Complete each statement by placing a mark in the box above the adverb that best expresses your opinion.

1. Learning how to select commands from the menus was ...

Hard								Easy
	Extremely	Quite	Slightly	Neutral	Slightly	Quite	Extremely	

2. Remembering which menu a particular command is in is ...

Hard								Easy
	Extremely	Quite	Slightly	Neutral	Slightly	Quite	Extremely	

3. The time it takes to find and select commands you use often is ...

Short								Long
	Extremely	Quite	Slightly	Neutral	Slightly	Quite	Extremely	

4. The time it takes to find and select commands you don't use very often is...

Short								Long
	Extremely	Quite	Slightly	Neutral	Slightly	Quite	Extremely	

5. Selecting commands with this software is ...

Hard								Easy
	Extremely	Quite	Slightly	Neutral	Slightly	Quite	Extremely	

6. Selecting commands with this software is ...

Inconvenient								Convenient
	Extremely	Quite	Slightly	Neutral	Slightly	Quite	Extremely	

7. When I executed a command selection I felt ...

Confident								Unsure
	Extremely	Quite	Slightly	Neutral	Slightly	Quite	Extremely	

8. Using this software is ...

Invigorating								Tiring
	Extremely	Quite	Slightly	Neutral	Slightly	Quite	Extremely	

Subject #: ______

9. On the last block of text I edited with this software I felt ...

Uncomfortable	Extremely	Quite	Slightly	Neutral	Slightly	Quite	Extremely	**Comfortable**

10. Overall I would rate the way you select commands with this software as ...

Bad	Extremely	Quite	Slightly	Neutral	Slightly	Quite	Extremely	**Good**

11. Overall I would rate the way you select commands with this software as ...

Slow	Extremely	Quite	Slightly	Neutral	Slightly	Quite	Extremely	**Fast**

12. Overall I would rate the way you select commands with this software as ...

Unfriendly	Extremely	Quite	Slightly	Neutral	Slightly	Quite	Extremely	**Friendly**

Appendix H: Comparison Questionnaire

Subject #: ______

Menu and Bypass Code Comparisons

As you began this experiment you told the computer what to do by selecting commands from menus. Midway through the experiment you began to use codes that bypassed the menus. In other words you told the computer what to do by inputting command codes that you had memorized, as opposed to selecting commands from a menu.

• Overall I thought it was **faster** to select commands with:
____ Bypass Codes ____ Menus

• Overall I thought it was **easiest** to select commands with:
____ Menus ____ Bypass Codes

• Overall I thought the most **convenient** way to select commands was:
____ Menus ____ Bypass Codes

• Overall I thought it was more **awkward** to select commands with:
____ Bypass Codes ____ Menus

• Overall I was more **confident** in my command selections when using:
____ Bypass Codes ____ Menus

• Overall I think it was **easier to learn** how to select commands with:
____ Bypass Codes ____ Menus

• Overall I thought it was more **frustrating** to select commands with:
____ Bypass Codes ____ Menus

• Overall it felt more **natural** to me to select commands with:
____ Bypass Codes ____ Menus

• Overall I think the **best way** to select commands is with:
____ Bypass Codes ____ Menus

• Overall I felt I had to **concentrate** more when selecting commands with:
____ Bypass Codes ____ Menus

• Overall I felt I did a **better job** at the editing tasks when I used:
____ Bypass Codes ____ Menus

• Overall I think I **finished the editing tasks faster** when I used:
____ Bypass Codes ____ Menus

• Overall it felt more **cumbersome** to me to have to select commands with:
____ Bypass Codes ____ Menus

Subject #: _____

• Overall I felt more **in control** when I was selecting commands with:
____ Bypass Codes ____ Menus

• Overall I felt more **anxious** when selecting commands with:
____ Bypass Codes ____ Menus

• Overall the method for selecting commands I **liked** the most was:
____ Bypass Codes ____ Menus

Future Designs

• Overall I think software would be **easier to learn** if it offered:
____ Bypass Codes ____ Menus ____ Both

• Overall I think it would be **easiest to select** commands if you had:
____ Bypass Codes ____ Menus ____ Both

• Overall I think software would be **less frustrating** to use if it offered:
____ Bypass Codes ____ Menus ____ Both

• Overall I think software would be **less awkward** to use if it offered:
____ Bypass Codes ____ Menus ____ Both

• Overall I feel people would **perform better** at computer tasks if they could select commands by using:
____ Bypass Codes ____ Menus ____ Both

• Overall I would **prefer** software that let you select commands with:
____ Bypass Codes ____ Menus ____ Both

VITA

Monty Hammontree was born in Athens, Georgia, on March 10, 1961. In May of 1984 he received the Bachelor of Arts degree in Psychology from David Lipscomb College. In August of 1986 he received the Master of Arts degree in Industrial Psychology from Middle Tennessee State University. In the fall of 1986 he entered the Graduate School of Old Dominion University, and subsequently received the Doctor of Philosophy degree in Industrial/Organizational Psychology in August, 1991. Monty performed his doctoral internship at the User Systems Engineering Center of Texas Instruments in Dallas, Texas. In April of 1991 he joined the technical staff at Texas Instruments as a Human Factors Engineer.