The Effects of Manipulating Recognition Variables on the Transition from Menus to Commands

by

J. Morgan Morris and Albert N. Badre

GIT-GVU-91-11
September 1991

Graphics, Visualization & Usability Center

Georgia Institute of Technology
Atlanta GA 30332-0280
The Effects of Manipulating Recognition Variables on the Transition From Menus to Commands

J. Morgan Morris
Georgia State University, Atlanta, GA, 30303, USA

AND

Albert N. Badre
Georgia Institute of Technology, Atlanta, GA 30332-0280, USA

The needs of users vary along with their degree of acquired expertise, with novices gladly accepting much-needed guidance and experts preferring to use the knowledge that they have gained with repeated use. The study of human-computer interaction recognizes this difference, and numerous interaction strategies have been developed to accommodate expertise differences. One such strategy is that of transitional interaction. This strategy accommodates the users' need to move up and down the scale of expertise by allowing them to toggle between recognition and recall modes. The goal of this research is to study in more detail the impact of manipulating recognition mode variables and their effect on inexperienced users' transitions from menu to command interaction.

Two experiments were designed and conducted to meet the above stated goal. The first experiment compared the effects of supporting a recognition mode versus command interaction only. Specifically, a transitional interface to UNIX was compared with a shell-based UNIX interface on measures of user performance and perceived usability. The transitional version resulted in both higher performance and higher perceived usability. The second experiment investigated the effects of menu item selection strategy on the transition from menu to command. The strategy of command completion to a unique form proved superior to typing a paired number as measured by trials to complete and accurate recall.
1. Introduction

A goal of the user interface designer is to build human interfaces that accommodate the changing requirements of the long-term user. Such interfaces should take into account varying levels of user experience and, at the same time, facilitate transitions up and down the scale of expertise. A person is very likely to experience many different user states when interacting with a given system. A user’s level of expertise and requirements change over time; the same user can be a novice at one time and an expert sometime later. The design of user-computer interaction strategies therefore should allow for a smooth transition such that after a time a user finds it natural to be in the experienced state simply by having practiced as a novice [Badre, 1984].

To facilitate transitions along the scale of expertise, the design should provide flexible interface environments to support user interactions that allow a linguistic continuum between recognition strategies, such as menus and prompts for the novice user, and free recall methods, such as the use of commands and macros for the expert. In this continuum, learning the expert’s language of interaction would be a byproduct of using the novice’s recognition mode. Invoking a command procedure for a given action amounts to remembering the selected items in a path of sequentially dependent menus and prompts.

1.1. TRANSITIONALITY: FROM MENUS TO COMMANDS

An effective transitional interface environment incorporates the following principles:

- The interaction strategies should incorporate syntactic and semantic continuity such that after a time a user finds it natural to be in an experienced state simply by having practiced as a novice.

- Experienced and expert users must be able to use the interface without the hardship of having to change established habits.

One way of providing for user transitions with these principles in mind is to allow the user to construct a command by choosing the items from a sequence of menus. In
a menu-to-command system one of the most important components is the arrangement and sequencing of the menus. Given well-designed menu sequences and allowed to construct command components without the necessity of recalling their names, a novice user is exposed to the contents of the system with little effort. Highly experienced users may be unfamiliar with particular commands and options that they have never found reason to use. For example, the `ls` command under UNIX has a considerable number of options: the Berkeley version of UNIX has twenty-one options, as does one version of System V, although they are not the same twenty-one options. When pondering the possible options, the new user's only resort is to consult the `man` pages or to ask a more experienced user what to do. The availability of the options as a part of the interface would help solve the problem of locating such information. Experienced users may have problems, given the plethora of options. As a canonical example, the tendency of UNIX users with the `ls` command is to become familiar with a small subset of the options that they will continue to use. Strictly speaking, such a user would not be an expert relative to the `ls` command, regardless of frequency of use and overall knowledge. Thus, availability of the options in the menu structure as part of the command area on the screen would facilitate familiarity with the full breadth and depth of a command's functionality.

1.2. RESEARCH GOALS

The overall goal of this study is to investigate the effects of recognition mode variables on the transition from menu to command mode interaction. Two experiments were designed to investigate these issues. The first experiment compared novice user performance on a standard system with that on the same system fitted with a transitional interface. Thus, it compared an interface supporting a recognition mode to one that did not. Subject performance was measured by percent of task completion during the on-line session. Follow-up questionnaires determined levels of anxiety and satisfaction. The leading hypothesis was that users of the transitional interface would exhibit higher percent task completion, lower anxiety, and higher satisfaction.
The second experiment was designed to investigate the effects of menu item selection strategy in a menu-to-command transitional interface. Subjects were assigned to groups who used one of two possible menu item selection strategies. One group's strategy was command completion to a unique form, i.e. they typed enough of the name of the item to make it unique within a menu. The other group typed a paired number. Subjects were presented with a set of tasks to perform, using the menu mode to construct each command. Then the subjects were asked to recall several targeted commands without having the menus available for consultation. The number of trials necessary for complete recall (regardless of order) was recorded. A similar procedure was repeated one week later. Then subjects were required to undergo trials until they achieved 100% accuracy in the order indicated by the target task script. Thus, the number of trials to complete and accurate recall serves as a measure of learning for the selection strategies. The leading hypothesis was that command completion would prove superior in comparison with paired number selection as measured by the number of trials to complete and accurate recall.

1.3. TUNIX: A TRANSITIONAL INTERFACE FOR UNIX

To meet the above goals, Tunix[Morris and Badre, 1988], a transitional interface to UNIX, was developed. Traditional command-line UNIX served as the standard interface. UNIX was chosen due to the wealth of functionality it offers and the well-documented problems with its user interface [Norman, 1981]. Tunix was intended to demonstrate a technique for alleviating the learning problems associated with UNIX.

The Tunix screen is divided into two main areas. Commands appear and execute in the “UNIX Area” located in the uppermost portion of the screen. The bottom portion of the screen is the “Command Area,” where menus and prompts are displayed and the command is constructed to completion. Thus, ordinary interaction with Tunix in menu mode consists of two phases. A command is constructed piecemeal with the support and guidance of menus and prompts in the Command Area. Then the command appears
Directory and File Management

Tier 1
CHOICES:  ls  cd  cat  cp  mkdir  rmdir  rm
Command:  
SELECT:Space EXEC:Enter BRK:CtrlU FORWARD:Tab BACKWARD:BackSpace

Figure 1: First Tier

Press '·' to activate options; CtrlN to bypass options.  ls options
Subtier  List in columnar format.
(1 of 3)
OPTIONS:  C  F  a  i  l  s  t  R  d
Command: ls
SELECT:Space EXEC:Enter BRK:CtrlU FORWARD:Tab BACKWARD:BackSpace

Figure 2: Initial Option Subtier for ls Command

in its entirety in the UNIX Area, along with the results of the command execution.
Figures 1 and 2 describe the sequence by which the commands and options are selected
from the menus to result in the construction of the command statements.

Construction of a command in menu mode may involve any or all of the following
tasks:

1. Moving from item to item to view the description of the command, so that one can
determine whether it is appropriate for the task.

2. Making the desired command the current item.

3. Selecting the current item from the tier.

4. Completing the command when prompted for options (options are also available
   in menus) and other arguments.

5. Executing the command.

Adding a group of options to the command being constructed may involve any or all
of the following tasks:
Directory and File Management

Tier 1

CHOICES:  ls  cd  cat  cp  mkdir  rmdir  rm

Command: _
SELECT:Space EXEC:Enter BRK:CtrlU FORWARD:Tab BACKWARD:BackSpace

Figure 3: First Tier of Experimental Version of Tunix

Information

Tier 2

CHOICES:  who  finger  pwd  cal  date  wc  man

Command: _
SELECT:Space EXEC:Enter BRK:CtrlU FORWARD:Tab BACKWARD:BackSpace

Figure 4: Second Tier of Experimental Version of Tunix

1. Activating the options so that they may be chosen.

2. Moving from item to item to view the description of the option, so that one can determine whether or not it is appropriate for the task.

3. Choosing an item to appear in the current group of options; removing an option from the current group of options.

4. Selecting the current group of options.

Two tiers of Tunix were organized for the purposes of experimentation. The first tier is presented in Figure 3, the second in Figure 4.

2. Experiment 1: A Comparison of Performance and Perceived Usability

Many advantages have been proposed for a system designed to incorporate transitionality [Badre, 1984]. For instance, the needs of new and novice users are accommodated with recognition-based interaction strategies. The needs of more experienced users are considerably different. As an example, experienced programmers must have barriers to their productivity eliminated so that time is not wasted unnecessarily searching for information. The organization of a transitional system and an emphasis on ease of learning
provide a strong foundation for eliminating major obstacles. The following experiment was conducted to validate these intuitively appealing claims. Inexperienced users were given the helpful information they needed to begin learning the system and the concepts necessary to do so. Meanwhile, experienced programmers were allowed to benefit from their knowledge without undue hardship in their attempts to become productive users.

Another important aspect of usability is the mental state of the user while using the system. A helpful system should have users who are pleased with certain features of the system and with progress made while using it. A poorly designed system may leave the user without any desire to use the system again. Both subjective issues and performance measures are critical in assessing the usability of a system. This experiment also attempts to obtain more detailed knowledge about user mental states while using both a standard system (UNIX) and a system designed to incorporate transitionality (Tunix). One important aspect of computer use is the amount of anxiety experienced by the user while using a system. Presumably, the more helpful, easy-to-learn, and easy-to-use a system is, the less stress a user undergoes while using the system. Immediately preceding and following each on-line session subjects were administered anxiety tests to measure a change in anxiety attributed to interaction. Another instrument was administered that was designed to address specific usability issues.

2.1. METHOD

2.1.1. Participants

Thirty-two Georgia Institute of Technology students were used in this experiment. Due to the verbal nature of the material, subjects were restricted to those speaking English as their native language. The subjects came from two main groups: one-half selected from a pool of subjects who had little experience with computers, and one-half from a pool of subjects who were experienced programmers but had no UNIX experience. The subjects were selected primarily from undergraduate psychology courses. They
received academic credit in exchange for their participation.

The participants were first asked to read and sign consent forms. Then they were asked to fill out information sheets in order to determine their experience level. Assignment to the ‘inexperienced’ group required less than two years and less than two languages worth of experience. Assignment to the ‘experienced’ group required two or more years experience and knowledge of two or more languages. Subjects were screened and only those representative of these groups were actually picked to participate in this experiment.

2.1.2. Apparatus

An experimental version of Tunix was implemented on an AT&T 3B20. The first two tiers, “Directory and File Management” and “Information,” were used for the task scripts for those subjects assigned to the Tunix group. The on-line monitor was activated to capture the commands entered by the subjects to aid in the data collection process. Subjects assigned to the UNIX group worked on a VAX 11/780 to take advantage of the script command to capture their on-line session. All interaction took place using an AT&T 4425 terminal.

2.1.3. Design

This experiment was designed as a generalized randomized block design, blocked by computer experience. The two groups of subjects differed in prior computer experience: “inexperienced” and “experienced.” The two treatment levels of system choice were standard (UNIX) and transitional (Tunix).

2.1.4. Procedure

Potential subjects completed the information questionnaires. Subject selection involved using the background and experience information they provided. Sixteen subjects were placed in each of the blocks according to their level of computer experience. They were then randomly assigned to one of the treatment levels, resulting in eight sub-
jects per treatment level per group. Thus, 32 subjects were blocked into two levels of experience, then assigned to one of two treatments of system choice.

After being seated and allowed a brief calming period, each subject was given the state portion of the State-Trait Anxiety Inventory, Form Y-1 [Spielberger et al., 1983] to assess their pre-treatment anxiety level. Each subject was then given a written tutorial relating to the UNIX file system to ensure sufficient background to undertake the tasks. Next, each subject in the transitional system treatment received a brief written tutorial introducing the Tunix interface. Those receiving the standard system treatment were given a tutorial describing the UNIX interface. This tutorial included information about the “man -k keyword” command used to find commands related to a keyword, and about the “man command-name” command used to locate information about a specific command.

After the tutorial, the subject was seated at a terminal and asked to complete a brief sample task. Upon successful completion of this task, a task script describing five command procedures was handed to the subject. Subjects were informed that they would have 15 minutes to execute all command procedures in the task script, and that they were being taped on audio cassette to track questions and comments of interest during interaction. The task script consisted of five UNIX command procedures. For example, the following item, representing the UNIX statement “ls -alF” appeared in the task script. The descriptions in the task script were worded carefully to closely match both the man pages and the Tunix descriptions.

(2) List the contents of the current directory. Use command options to do the following:

- ensure that all of the files, including hidden files (files whose names begin with a '.') are displayed on the screen
- list the files in long format, so that detailed information about each file is displayed
• display the file type for each file (i.e. directories are marked with a trailing ‘/’, executable files with a ‘*’)

During execution of the task script, the subject was allowed to ask questions pertaining to the location of information, but the experimental technician did not provide specific answers. UNIX Subjects were given paper and pencil in the event they wanted to record any information. This was often the case as the UNIX user paged through the on-line help manual to locate information.

Following completion of the task script, the subject again received the state portion of the State-Trait Anxiety Inventory, Form Y-1 to assess post-treatment subject anxiety. Finally, a user opinion survey was given to assess the level of perceived usability of the system in question. This survey was developed in-house, since no known surveys dealt with this particular situation. Many of the questions were based on those found in [Shneiderman, 1987].

The experimental task required 15 minutes. The tutorial and testing usually lasted 35 minutes. The entire procedure was designed to take approximately an hour.

2.2. RESULTS

The analysis was conducted using MYSTAT [MYSTAT, 1988], a widely available personal version of SYSTAT, when parametric procedures were required. The nonparametric tests were performed by hand with the assistance of a calculator.

2.2.1. Performance

The primary response measure used to indicate performance was the percent of command items touched. A command item was either a command name or an option name, and an item was “touched” if it had either been executed or marked for execution before the time limit expired. For instance, if a subject using the UNIX system wrote an option name on the provided piece of paper, and was unable to execute the command, that option name was considered touched. Similarly, a subject using Tunix may have selected
items from the menus without being able to execute the command. The five command procedures contained ten items. Figure 5 depicts the means and standard deviations for each cell.

The zero standard deviations for the transitional system column preclude use of ANOVA for this data. Note that everyone who used the transitional system touched every item. This is strong enough to make a statement without further statistical analysis. Thus, the transitional system led to superior performance over the standard UNIX interface.

In conclusion, the system choice, regardless of experience level, contributed to the performance of tasks in this experiment, while previous experience did not prove to be a factor. The results are similar for other measures of performance. The percentage of subjects finishing the task script in the allotted time is shown in Figure 6. Overall, 15 of 16 subjects who used the transitional interface completed the task script, while only four of 16 did so using the standard interface.

2.2.2. Attitude

Attitude measures included two tests, one designed to assess anxiety and one for perceived usability. Prior to the experiment, the sensitivity of the anxiety instrument to the possibly low levels of anxiety attributable to using an unfriendly computer system was in doubt. The subjects' technical background was considered likely to reduce their anxiety in this situation. For this reason, a user opinion survey was developed and
System Choice

<table>
<thead>
<tr>
<th>Experience</th>
<th>Standard</th>
<th>Transitional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inexperienced</td>
<td>12.5 (1/8)</td>
<td>87.5 (7/8)</td>
</tr>
<tr>
<td>Experienced</td>
<td>37.5 (3/8)</td>
<td>100 (8/8)</td>
</tr>
</tbody>
</table>

Figure 6: Percent of Subjects Who Finished in Each Group

Table 1: Descriptive Statistics for Change in Anxiety By Experience

<table>
<thead>
<tr>
<th>Experience</th>
<th>Mean Change</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inexperienced</td>
<td>+5.1875</td>
<td>-6</td>
<td>+25</td>
</tr>
<tr>
<td>Experienced</td>
<td>+2.1875</td>
<td>-14</td>
<td>+17</td>
</tr>
</tbody>
</table>

administered in order to estimate the users’ perceptions of system usability.

The normative values for the STAI state scale are 36.47 (SD = 10.02) for male college students, and 38.76 (SD = 11.95) for female college students. (Higher scores indicate greater anxiety.) The pre-task anxiety values were 32.333 (SD = 9.388) for male subjects and 31.636 (SD = 4.501) for female subjects. The overall pre-test anxiety value was 32.094 (SD = 7.969). The pre-test scores are below the normative values, and indicate that the subjects were not anxious prior to the experimental task.

None of these results proved significant. The degree of anxiety induced by an on-line session is small in scale as compared with the anxiety evident in situations normally evaluated by the STAI. Thus, any individual anxiety detected may be the result of events occurring before the session, e.g. a disastrous calculus exam, or to individual differences such as fear of computers. Others seemed anxious participating in the experiment. In support of the appropriateness of the STAI for this experiment, Guynes [Guynes, 1988] demonstrated that anxiety can be detected in an on-line session using the STAI. No difference between system types was apparent from the data.
Table 2: User Opinion Survey

1. ___ The system commands are easy to use.

2. ___ I feel competent with and knowledgeable about the system commands that I used.

3. ___ I would like to use the system again.

4. ___ I felt very frustrated when I used the system.

5. ___ I believe I could use the system without referring to the printed manual or the on-line help facility (the `man` command).

6. ___ I feel that the system provides a great deal of helpful information that I need to use it effectively.

10. ___ I enjoyed using the system.

A slight possibility for a difference exists between levels of experience (see Table 1). At any rate, the results do not demonstrate significance between the system types or levels of experience for this experiment.

The usability results proved to be more encouraging, although claims must be cautious. After all, the instrument was developed for this experiment and was not empirically validated. The original questionnaire, a Likert-type scale [Anastasi, 1982], contained twelve items to be marked: ‘1’ strongly agreed, ‘2’ agreed, ‘3’ neutral, ‘4’ disagreed, and ‘5’ strongly disagreed. Only seven of the items were considered for analysis due to repetition. The sum of the scores for these items is called the corrected usability score. High values indicate a greater amount of satisfaction or perceived usability. Table 2 represents the items considered for analysis.
System Choice

<table>
<thead>
<tr>
<th></th>
<th>Standard</th>
<th>Transitional</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inexperienced</td>
<td>20.500 (3.586)</td>
<td>25.250 (2.816)</td>
</tr>
<tr>
<td>Experienced</td>
<td>24.375 (3.503)</td>
<td>26.750 (3.655)</td>
</tr>
</tbody>
</table>

Figure 7: Mean (Standard Deviation) for Corrected Usability Scores

Table 3: Results of ANOVA for Corrected Usability Scores

<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>DF</th>
<th>MS</th>
<th>F</th>
<th>P</th>
<th>Sig</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experience Level</td>
<td>57.781</td>
<td>1</td>
<td>57.781</td>
<td>4.980</td>
<td>.034</td>
<td>*</td>
</tr>
<tr>
<td>System Choice</td>
<td>101.531</td>
<td>1</td>
<td>101.531</td>
<td>8.751</td>
<td>.006</td>
<td>*</td>
</tr>
<tr>
<td>Interaction</td>
<td>11.281</td>
<td>1</td>
<td>11.281</td>
<td>0.972</td>
<td>.333</td>
<td></td>
</tr>
<tr>
<td>Error</td>
<td>324.875</td>
<td>28</td>
<td>11.603</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The maximum usability score was 35. Figure 7 summarizes the means and standard deviations for each cell. The results of the ANOVA are summarized in Table 3. (The nonparametric analysis failed to show significance for experience level by a small margin.) Thus, the accepted analysis is that both experience level and system choice proved to be significant. Experienced subjects are more likely to give a system a higher usability rating, regardless of the system. Transitional system users are more likely to give the system a higher rating, regardless of their experience. There is no interaction between the two factors.

Statements 1, 5, and 10 were most responsible for the differences. Those using the transitional interface were more likely to feel that the system’s commands were easy to use than were the standard interface users (statement 1). Additionally, they felt that they could use the system more readily without accessing other forms of help (statement 5). Finally, more experienced users were most likely to respond that they enjoyed using the system (statement 10), whether they used Tunix or standard UNIX. The means and
Table 4: Means per Statement for Corrected Usability Scores

<table>
<thead>
<tr>
<th></th>
<th>Inexperienced</th>
<th>Inexperienced</th>
<th>Experienced</th>
<th>Experienced</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Standard</td>
<td>Transitional</td>
<td>Standard</td>
<td>Transitional</td>
</tr>
<tr>
<td>S1</td>
<td>2.75 (1.165)</td>
<td>1.75 (.707)</td>
<td>2.00 (.756)</td>
<td>1.75 (.463)</td>
</tr>
<tr>
<td>S2</td>
<td>3.5 (1.195)</td>
<td>2.75 (.707)</td>
<td>3.00 (1.195)</td>
<td>2.375 (1.061)</td>
</tr>
<tr>
<td>S3</td>
<td>2.25 (.886)</td>
<td>2.25 (.463)</td>
<td>1.75 (.463)</td>
<td>1.875 (.835)</td>
</tr>
<tr>
<td>S4</td>
<td>3.00 (1.195)</td>
<td>3.75 (.707)</td>
<td>3.375 (.916)</td>
<td>3.875 (.991)</td>
</tr>
<tr>
<td>S5</td>
<td>4.75 (.463)</td>
<td>3.5 (1.069)</td>
<td>4.5 (1.069)</td>
<td>2.75 (1.035)</td>
</tr>
<tr>
<td>S6</td>
<td>2.25 (.886)</td>
<td>2.125 (.641)</td>
<td>1.5 (.756)</td>
<td>1.75 (.707)</td>
</tr>
<tr>
<td>S10</td>
<td>3.00 (.756)</td>
<td>2.125 (.641)</td>
<td>2.25 (.886)</td>
<td>2.625 (.744)</td>
</tr>
</tbody>
</table>

Table 5: Corrected Usability Scores By Completion of Task

<table>
<thead>
<tr>
<th>Group</th>
<th>Sample Size</th>
<th>Mean</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incomplete</td>
<td>13</td>
<td>21.538</td>
<td>3.597</td>
</tr>
<tr>
<td>Complete</td>
<td>19</td>
<td>26.053</td>
<td>3.188</td>
</tr>
</tbody>
</table>

standard deviations for each group-treatment combination are shown in Table 4. Taken together, these statements indicate a difference in perceived usability by system choice, with Tunix receiving the higher rating, and by experience.

One interesting comparison is the mental state of those subjects who completed the entire task script compared to those who did not. Table 5 provides a comparison between these groups for perceived usability, while Table 6 provides a comparison based on change in anxiety. The usability scores prove significant (F=13.953, P = .001) while the anxiety scores do not (F = 1.363, P = .252). The usability results attest

Table 6: Anxiety Scores By Completion

<table>
<thead>
<tr>
<th>Group</th>
<th>Sample Size</th>
<th>Mean</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incomplete</td>
<td>13</td>
<td>5.692</td>
<td>8.107</td>
</tr>
<tr>
<td>Complete</td>
<td>19</td>
<td>2.316</td>
<td>7.9186</td>
</tr>
</tbody>
</table>
Table 7: Results of ANOVA for Number of Questions Asked

<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>DF</th>
<th>MS</th>
<th>F</th>
<th>P</th>
<th>Sig</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experience Level</td>
<td>57.781</td>
<td>1</td>
<td>57.781</td>
<td>3.624</td>
<td>.067</td>
<td></td>
</tr>
<tr>
<td>System Choice</td>
<td>108.781</td>
<td>1</td>
<td>108.781</td>
<td>6.824</td>
<td>.014</td>
<td>*</td>
</tr>
<tr>
<td>Interaction</td>
<td>.781</td>
<td>1</td>
<td>.781</td>
<td>.049</td>
<td>.826</td>
<td></td>
</tr>
<tr>
<td>Error</td>
<td>446.375</td>
<td>28</td>
<td>15.942</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

that those who finish are likely to give the system a higher usability rating. Even so, since the majority of those who finished (15 of 19) came from the Unix group, the most significant factor is still the system choice.

2.2.3. Information Availability

Several measures addressed the amount of additional information, i.e. information not included as part of the interface, needed to perform the task script. For the standard interface user, this included the number of questions asked and the number of times help was accessed. The only measurable quantity for the transitional interface user was number of questions asked, since they were not permitted to use the on-line help system. The overall number of questions per group-treatment combination is summarized in Figure 8. These were also compared using both parametric and nonparametric methods. The results are provided in Table 7.

Clearly, the number of questions asked varies according to the system used, with
those using the standard system asking more questions than those using the transitional interface. A large part of the difference is accounted for by questions involving the man pages and command syntax. Standard interface users asked a total of 132 questions, of which 39 involved the help system and 27 command syntax. Transitional interface users did not use the man pages, and asked only eight questions regarding command syntax out of a total of 63 questions. As should be expected, an indication exists that inexperienced users asked more questions than did experienced ones, although the difference was not significant (p = .067).

Those using the standard interface accessed the on-line help system in order to get the information they needed to build and execute commands successfully. The mean number of help accesses for this group is as follows: 9.5 (SD = 2.330) for the inexperienced group, and 11.25 (SD = 3.495) for the experienced group. Apparently, the experienced group was more familiar with the requirements of the task, and was able to accomplish more in the given time.

3. Experiment 2: The Effects of Selection Strategy on Recall

One key feature of a menu-based interface is the strategy used in selecting items from the menu. Shinar and Stern [Shinar and Stern, 1987] studied the effects of three menu selection strategies: (1) keying an associated number, (2) typing the first letter of the item, and (3) moving the cursor to the item. They found reaction time (RT) to be consistently longer for those using paired numbers. Inexperienced users performed better initially with cursor movement, but after further use typing the first letter became the superior method. The authors cited several reasons for the superiority of letter coding: “Letter coding permits (a) consistent selection of the same options across various menus of different lengths and contents; (b) consistent selection of options during stages of system development when menus are changing; and (c) direct transfer to coded command entry in applications in which novice users become more experienced.” Point ‘(c)’ forms an intriguing study involving a transitional system. In a transitional use context, where
one learns by using menus and then recalls by invoking a command sequence, the menu item selection strategy may have a significant effect on recall and remembering. Possibly, the cognitive processing involved in selection of a menu item via typing in an associated, yet unrelated key, e.g. a paired number, is somewhat different from typing in part or all of the menu item.

The purpose of this experiment was to determine the effects of computer experience and menu item selection strategy on acquisition, recall, and remembering of command procedures for the longitudinal user in a transitional use context. Two selection strategies were examined. They were command completion to a unique form and typing a paired number. The leading hypothesis was that the effect of selection strategy on acquisition and recall of command procedures was significant. Furthermore, command completion would provide superior results to the paired number case.

3.1. METHOD

3.1.1. Participants

The subjects were 16 Georgia Institute of Technology students who received academic credit in exchange for their participation. The subjects were selected from undergraduate psychology classes. They had little or no experience with computers and less than two years programming experience. Of course, no person with any background in UNIX was selected.

3.1.2. Apparatus

As in Experiment 1, Tunix was used as the experimental system. The first tier, “Directory and File Management,” was the focus of the experimental tasks, while the next tier, “Information,” was used for the tutorial procedure.

3.1.3. Design

The design of this experiment was a comparison of two samples, provided by the
menu item selection strategy treatments. The two levels of treatment were command completion (name) and paired number.

3.1.4. Procedure

Eight subjects were randomly assigned to each group. During the training session, subjects were given tutorials on both the UNIX file system and the Tunix user interface. Then, subjects were seated at a terminal to complete a sample task.

Following the tutorials, each subject was given a task script representing a sequence of four typical UNIX file manipulation command procedures. The subject was asked to work through the task script using the assigned menu item selection strategy, executing each command procedure successfully. The order of the command procedures, and the order of option descriptions, was randomly varied for each on-line session. Immediately after this on-line session, the subject was moved to a table to perform an alphabetic distractor task. Next, the subject was shown a task script similar to the on-line task script and asked to reconstruct on paper three of the command procedures from memory. The total number of items (command and option names) to recall was 10.

Each sequence of interaction/distraction/recall formed one trial. If an error was made in reconstructing the procedure, then the subject completed another trial. For the first session, trial repetition continued until: (1) a subject recalled all of the items correctly in less than five trials, or (2) after five trials a subject was able to recall at least eight of the 10 items correctly. During the first session, the subjects would become familiar with the physical experimental environment, the experimental system, the experimental procedure, and the command items used in the experiment.

The subjects were asked to return one week later to take part in a second session. This time they completed trials until they were able to recall all 10 items correctly, regardless of the order used to reproduce them. Until this point, the subjects had seen the same target task script during each command procedure recall phase of a trial. They then completed more trials with a different target task script until they were able to recall all of the items provided in the order specified by the task script. The total number of
 trials was recorded during the second session as the response variable.

3.2. RESULTS

The number of trials to ordered recall during the second session were analyzed using the Mann-Whitney test [Conover, 1980]. (These calculations are relatively simple and were performed using a hand calculator.) The results showed a significant difference between the groups (T1 = 2.17, p < .05). The data were also analyzed using a t-test. These results also proved significant. However, the Mann-Whitney test was used because of the sample size.

The primary response measure was number of trials to complete ordered recall, with time to completion collected as a support measure. As shown in Figure 9, the mean number of trials for the paired number group was 3.875. They took an average time of 20:09 (SD = 10:56) to complete, as depicted in Figure 10. The mean number of trials for the name group was 2.375, taking an average time of 12:03 (SD = 5:56) to finish. The time to completion during the second session was analyzed with the Mann-Whitney test using a significance level of α = .05. The test statistic fell on the ω_{0.975} quantile, formally indicating acceptance of the null hypothesis of no difference. This borderline rejection is close enough to indicate that a true difference exists. The means and standard deviations support this conclusion.

The data from both sessions were explored more deeply to determine what con-
tributed to the significant results. The Mann-Whitney test ($\alpha = .05$) was used throughout.

The number of trials to 100% recall during the first session was analyzed for differences between the two groups. (One trial was added to the total for the subjects who did not reach 100% accuracy.) No differences were detected for the first session ($T_1 = .9727$).

During the second session, subjects were required to achieve two levels of recall accuracy. The first level was 100% recall with option order irrelevant. The number of trials to this level was analyzed, and the results proved significant ($T_1 = 2.1623$). The second level of recall accuracy was number of trials to 100% recall with option order significant. The differences between the two levels was computed and analyzed. The results did not prove significant ($T_1 = 1.09$).

Thus, no immediate effect of selection strategy on recall was detected during the first session. Subjects using the command completion to uniqueness strategy performed better during the first level of recall than did the paired number group. This indicates that forgetting over time is easier for the paired number group. Another contribution to the overall significance of trials to complete ordered recall may have been the use of nonproductive learning strategies. More subjects in the paired number group had difficulty with the option order than did those in the command completion group. Thus, the superiority of command completion selection over paired number selection may be attributed to two factors: less forgetting and more productive learning strategies.

The times for various critical points during the experiment are presented in Table 8. $T_{2(1)}$ indicates the time to complete the first level of the second session. Similarly, $T_{2(2)}$ denotes the time to complete levels one and two of the second session. $T_{tot}$ is the total time to criterion across the experiment, the sum of the time to reach criterion in session one and $T_{2(2)}$. These times provide further support for the superiority of command completion selection.
Table 8: Means (Standard Deviation) for Times in Experiment 3

<table>
<thead>
<tr>
<th>Paired Number</th>
<th>Command Completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{2(1)}$</td>
<td>10:21 (5:12)</td>
</tr>
<tr>
<td>$T_{2(2)}$</td>
<td>20:09 (10:56)</td>
</tr>
<tr>
<td>$T_{tot}$</td>
<td>61:50 (25:59)</td>
</tr>
</tbody>
</table>

4. Discussion of Experimental Results

4.1. EXPERIMENT 1

Several conclusions may be derived from the data analysis. Evidence supports the claim that the availability of information found in the experimental transitional interface is beneficial. For instance, compared with standard UNIX users, (1) subjects using Tunix finished more of the task script in the allotted time, (2) subjects using Tunix voiced greater approval in a user opinion questionnaire, and (3) Tunix users searched for less information and completed more items.

Experiment 1 models a user’s first exposure to the system. Without prior experience, a user might become lost or confused in many ways while trying to use the system. The results support the claim that Tunix helps eliminate one of the biggest obstacles to learning UNIX—finding needed information in a timely fashion. It may be argued that comparing the UNIX man pages with practically any other help system would result in unfavorable results for the UNIX system. For example, Borenstein [Borenstein, 1985] found the man pages to be inferior to both an alternative help system and a human tutor. However, help is integrated into the Tunix interface. Thus, the user may not need to interrupt the current task to search for information, and may concentrate on the current task. The user locates command names and option names in a semantically organized
menu system. Within the menus, command and option descriptions associate the names with functionality. In a more elaborate system than Tunix, a layered help system would be available to answer further questions. Unlike UNIX in many cases, Tunix also provides guidance restricting numerous nongermane attempts at solving the current problem. Linguistic continuity is a key, supplying the standard interface vocabulary and syntax, but with prompts for options and arguments. Tunix constructs the command at the bottom of the screen as the user proceeds, eliminating many doubts about command syntax. Tunix's guidance and availability of information is a boon for both inexperienced and experienced users who need to learn and use UNIX.

Experiment 1 supports these conclusions: (1) Tunix, a system retrofitted with an interface designed to incorporate transitionality, is superior to the standard UNIX interface in terms of performance and perceived usability, and (2) Tunix is a valid transitional interface, and may be used for subsequent studies of factors involved in learning. Thus, the transitional structure provides the foundation for menu-to-command transitions.

4.2. EXPERIMENT 2

A significant difference was found between the two methods of menu item selection strategy. Picking an item using command completion to a unique form results in faster, more efficient acquisition of a set of command procedures. The extension of these results to other transitional use interfaces is possible, but some doubts must be erased before such a strong generalization can be made. The UNIX environment was selected because of the well-known problems associated with learning to use its interface. The degree to which these problems were a factor in the experiment is unknown. For example, UNIX command names are often inappropriate, which could be a factor. During Experiment 2, subjects tended to recall the options first, then the command names and seemed to have particular problems with "ls" and "cat". Further, UNIX has several commands with numerous options; other systems would not be this complex. Before strong generalizations can be made, the questions of command naming and command complexity must
be addressed. No reason compels belief that this stronger generalization will prove to be false.

The overall significance of Experiment 2 may be summarized as: (1) in general, transitional interfaces should avoid paired numbering as the primary method of menu item selection, since evidence exists that command completion to uniqueness leads to quicker and more efficient learning of the command vocabulary, and (2) users of systems with naming problems, as in UNIX, would benefit more from a transitional interface using command completion as its menu item selection strategy than one using paired numbers. The results of Experiment 2 provide further evidence for the superiority of mnemonic or related alphabetic selection strategies. The work of [Shinar and Stern, 1987] and [Perlman, 1984] measured reaction time and selection time respectively. This study adds facilitation of recall to this list of measures for which mnemonic alphabetic keys are superior.

5. Conclusion

The first experiment's results relative to performance and mental state suggest that the transitional framework implemented in Tunix is worthy of further research and development. The availability of menu sequences in the recognition mode of interaction allows novices to use the system productively. The advantages of learning command structures in a command language environment include abbreviated interaction and creation of new commands from frequently executed command sequences. Experiment 2 supported the hypothesis that a command completion to uniqueness menu item selection strategy leads to improved learning performance. Thus, the availability and design of menu sequences can have a significant effect on the long-term user's ability to use and learn command structures in a transitional interface.

References


