

## Comparison of Mouse and Keyboard Efficiency

**Richard C. Omanson**  
Infogix

**Craig S. Miller**  
DePaul University

**Elizabeth Young**  
DePaul University

**David Schwantes**  
DePaul University

In many corporate settings, users are required to quickly execute commands. Three methods of issuing commands were compared: 1) selecting a menu item with a mouse (Menu-Mouse condition); 2) selecting a menu item with a keyboard shortcut (Menu-Keyboard condition); and 3) selecting a toolbar item with a mouse (Toolbar-Mouse condition). Users performed one of the three methods across 90 trials and had their speed assessed in blocks of 30 trials. Overall, the Toolbar-Mouse method was the fastest, while the Menu-Keyboard condition showed the most improvement. A GOMS-based model is presented that accounts for differences among methods. This work confirms the use of toolbars for common commands, but also suggests that for heavily-used interfaces, keyboard shortcuts can be as efficient as toolbars and have the advantage of providing fast access to all commands.

### INTRODUCTION

In many corporate settings, users spend hours each day doing computer tasks on which they are required to be efficient. In these settings, the speed at which users issue commands is important. Three common ways to enter commands are via keys, menus, and toolbars. These three methods differ both in terms of device (mouse and keyboard) and whether the commands are organized in a flat or hierarchical (i.e., menu) structure.

additional physical operation, it also involves a mental decision about which menu contains the item. This *categorical* decision is akin to decisions needed for information foraging tasks (Pirolli & Card, 1999) such as web navigation. As a consequence, the hierarchical process imposes two additional costs: that incurred by categorical decisions and that incurred by additional physical selections. In this way, hierarchical menus require decision processes that differ from menu tasks used in previous studies (e.g. Cockburn, Gutwin, & Greenberg, 2007; Hornof, & Halverson, 2003), which only require the literal matching of character strings.

### Effects of Device and Menu Structure

Previous studies on practiced users have indicated that using the keyboard is a faster method for entering commands than the mouse, at least for keyboard methods that only require a single key command. Experiments by Lane, Napier, Peres, and Sandor (2005) found simple keyboard sequences to be faster than drop-down menus (accessed by the mouse) for performing common editing commands in MS Word. Other experiments (Jorgensen, Garde, Laursen, & Jensen, 2002; Karat, McDonald, & Anderson, 1986) also found that using the keyboard to issue a single key command was faster than using the mouse.

For the tasks in these studies, two factors ostensibly explain the faster selection times for the key commands relative to mouse-accessed menus. First, the physical motor operation of pressing a key is substantively faster than pointing with a mouse. For example, Card, Moran and Newell's (1983) GOMS keystroke-level model estimates each key press to take .2 seconds whereas moving the mouse and clicking to take about 1.3 seconds (p. 264). Even if a key command requires pressing and holding an initial control key, the key sequence is still more than three times faster than the point and click sequence.

Second, menu use, at least for the tasks in these studies, requires a multi-step, hierarchical process. Consider Table 1, which contrasts steps needed for a simple key command with steps needed for hierarchical menu selection. Not only does this use of hierarchically-structured menus require an

Table 1. Simple Key Versus Mouse-Based Menu Selection

Selection by Keyboard	Selection by Mouse
1. Recall command key	1. Decide which menu has item
2. Hold control key	2. Look for menu
3. Press command key	3. Point to menu
	4. Click mouse
	5. Look for item
	6. Point to item
	7. Click mouse

Of course, menu selection need not encompass a hierarchical organization. For example, a toolbar is a type of menu that openly displays all selections. At the same time, key sequence commands can be hierarchically organized. For example, many Windows-based applications permit a user to access a menu item by first pressing the Alt key, then a letter designating the menu, and finally a letter designating the menu item. Despite potential costs, hierarchical organizations offer a distinct advantage: they can support a much larger number of selections. When adopting hierarchical organizations in a design, it would be useful to know whether they are necessarily less efficient than mouse-based toolbars and, if so, to what extent. However, since the differences between menus and toolbars have a mixed effect on performance, it isn't always obvious which structure will be the fastest. This is especially true when menus accessed via a keyboard are compared with toolbars accessed via a mouse.

## Effects of Practice

Relative efficiencies are further compounded with the effects of practice. That practice reduces task time has been documented many times and has been shown to be proportional to a power of the amount of practice, which is referred to as the Power Law of Practice (Newell & Rosenbloom, 1980). While practice may reduce the time of individual steps, more substantive reductions may occur by replacing less efficient steps with more efficient ones. One example of step replacement enabled by practice is replacing visual scanning for an item with recalling item location from memory. Cockburn et al. (2007) have modeled the gradual shift from a linear visual search to recalling item location from memory that is brought about by practice. In their model, recalling an item's location has a logarithmic relation to the number of items as described by the Hick-Hyman law (Hick, 1952; Hyman, 1953). The extent to which users shift from linear search to logarithmic recall is determined by their expertise (practice).

A similar type of step replacement is possible when items are selected by entering a keyboard sequence. Rather than searching for menu items to find which letter is in the keyboard sequence, the entire sequence can be recalled from memory.

For this paper, we present a study that investigates practice effects on command entry with respect to device and hierarchical structure. From a practical standpoint, we investigate command methods that an interaction designer might consider: toolbars, mouse-based menus and hierarchical key-sequences. Based on a real-world application in the banking industry, our study has users invoke one of the three command methods to activate a window in an application. These three methods allow us to explore some relative tradeoffs of the factors we have discussed here. We focus on hierarchical key-based commands, partly because they have been less studied and partly because they provide a greater possibility of selections. With this focus, our choice of command methods will provide the basis for further analysis of how device, hierarchical structure, and practice affect efficiency.

## METHOD

### Participants

This study was conducted with 36 participants between the ages of 19 and 61, of which 9 were male and 27 were female. Twenty participants were university students while the remaining 16 were quality assurance testers, technical writers, and interface designers of a software firm. Thirty-two of the participants used Windows as their primary operating system, while the remaining four used an Apple system.

## Materials

The trials were conducted on Windows computers with a monitor resolution large enough to accommodate the experiment application, which logged selections and selection times from each trial. For the experiments, the windows were open to show all selections on the toolbar.

The program allowed users to open 20 different windows using one of three command methods: toolbar, mouse-based menus and hierarchical key-sequences. The windows were accessible from two domain menus (Regions and Business Units). Each domain menu contained 10 alphabetized items corresponding to U.S. states (e.g. Florida, Iowa) and business lines respectively. Selecting an item in these menus opened the corresponding window. Each domain menu could be opened either by a mouse click or by entering an Alt sequence (Alt+underlined letter). Similarly, items within opened menus could be selected either by the mouse or adding the item letter to the Alt sequence used to open the menu (e.g., Alt+R+I displayed the Region menu and opened the Illinois window).

In addition, all 20 windows could be opened from a static toolbar below the domain menus. The toolbar buttons were grouped by content (regions and business units). Toolbar buttons were only accessible by mouse click. The order of the toolbar buttons matched the order used in the domain menus.

## Design and Procedure

Each participant was given background information on the experiment and assigned a window selection method to use for their trials. Participants were randomly assigned to a condition by rotating sequentially through the three conditions (toolbar-mouse, menu-keyboard, or menu-mouse).

After filling out a questionnaire pertaining to their computing habits, participants were shown how to use their assigned method of command entry on the test application. Participants were given four blocks of trials to complete. The first block was a set of five practice trials. The remaining three blocks were sets of 30 test trials. To prevent fatigue, each participant was given a 20 second rest between blocks. By the time the practice trials were completed, all participants had learned to start each trial with their hands on the appropriate input device (keyboard or mouse).

For each block of trials, a prompt was displayed in the center of the screen. Clicking the OK button started the block of trials. For each trial within a block, the program displayed the name of the target window (e.g., Georgia) and the method to use to find it (e.g., Toolbar) at the bottom of the screen. When the target window was opened, the next trial in the block began. Selections and trial times were automatically recorded.

At the end of the experiment, the university students were given a \$5 gift card as compensation for their time.

The measure for the window-finding task was the time (in milliseconds) taken to bring the target window into focus.

## RESULTS

### Experiment Trials

To examine the effects of condition and trial on selection time, means for each block of trials were computed for each participant. These results are shown in Figure 1.

About 1% of the trials showed times that were at least 5 seconds slower than adjacent trials. As it is likely that during these trials, the participant lost concentration or was distracted, a log transformation was performed on the data to mitigate outlier effects (see Keppel, 1973, p. 557).

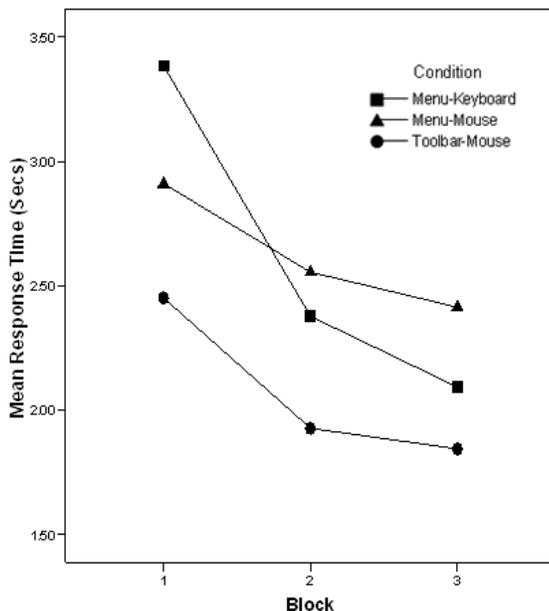


Figure 1. Response time of the Menu-Keyboard, Menu-Mouse, and Toolbar- Mouse Conditions.

The data were analyzed using a 3 (Condition) X 3 (Block) mixed ANOVA on the log transformed data. The main effect of Condition,  $F(2, 33) = 4.002, p = .028$ , the main effect of Block,  $F(2,66) = 153.992, p < .001$ , and the Condition X Block interaction,  $F(4,66) = 9.659, p < .001$ , were significant.

With each block, participants in each condition got faster. However, the rate of improvement was greater in the Menu-Keyboard Condition than in the other two conditions. Scheffe post-hoc tests showed that in Block 1, response times in the Toolbar-Mouse Condition were faster than those in the Menu-Keyboard Condition,  $p = .017$ . In contrast, in Block 2 and Block 3, response times in the Toolbar-Mouse Condition were only reliably faster than those in the Menu-Mouse Condition,  $p = .013$  and  $p = .035$  respectively.

Within each condition, correlated  $t$  tests indicated that response times in each subsequent block were faster than those in the previous block,  $p < .015$ , except in the Toolbar-Mouse Condition, where there was no statistical difference between the response times in Blocks 2 and 3,  $p = .173$ .

## DISCUSSION

The results indicate a clear advantage of using the toolbar over mouse-based menu access, both for initial performance and after three blocks of practice. While the menu-keyboard method achieves the greatest improvement in performance, it is still slower than the toolbar method throughout the three blocks. Although performance starts leveling off after three blocks, it is possible that performance for the menu-keyboard method would eventually surpass that of the toolbar method with sufficient practice. To better understand the operational costs of the three methods and assess their long-term, asymptotic performance, we turn to a componential analysis of all three methods for their performance in the third block. In the next section, we apply established performance constants for practiced users to show that users are near asymptotic performance for the toolbar method and the menu-mouse method. For the menu-keyboard method, the analysis identifies operational costs that need to be reduced for it to equal or surpass toolbar performance.

### Model of Method Effects

A GOMS (Goals, Operators, Methods, Selection) model at the Keystroke Level, GOMS\_KLM (Card et al., 1983), is used as a starting point with refinements derived from work on menu modeling by Cockburn et al. (2007) and categorical recall by Medin and Smith (1981). As much as possible, we rely on previously established principles and time estimates for hypothesizing constituent steps.

For user performance during the final trials, the three methods can be decomposed as shown in Table 2. By the third block of trials, the users are familiar with the menus and the items they contain. When an item is presented, users classify the item as a member of one of the menus and recall the letter associated with the item.

Table 2. Steps in each Command Entry Method after Category and Location Memorization

Menu-Keyboard	Menu-Mouse	Toolbar-Mouse
1. Categorize item	1. Recall item location	1. Recall item location
2. Hold Alt key	2. Point mouse to menu	2. Point mouse to item
3. Press menu letter	3. Click mouse	3. Click mouse
4. Recall item letter	4. Point mouse to item	
5. Press item letter	5. Click mouse	

In the menu-mouse condition, a strategy of recalling the location of the item on the screen when the menu is exposed is used. Evidence for this type of location memory comes from eye movement research (e.g., Hornof & Halverson, 2003) that found people look at where an item will be prior to item actually appearing. This same strategy of recalling an item's location can also be used for toolbar items.

Figure 2 presents models for the last block of each method, where the models predict the correct order of observed times and come within .2 seconds of the observed

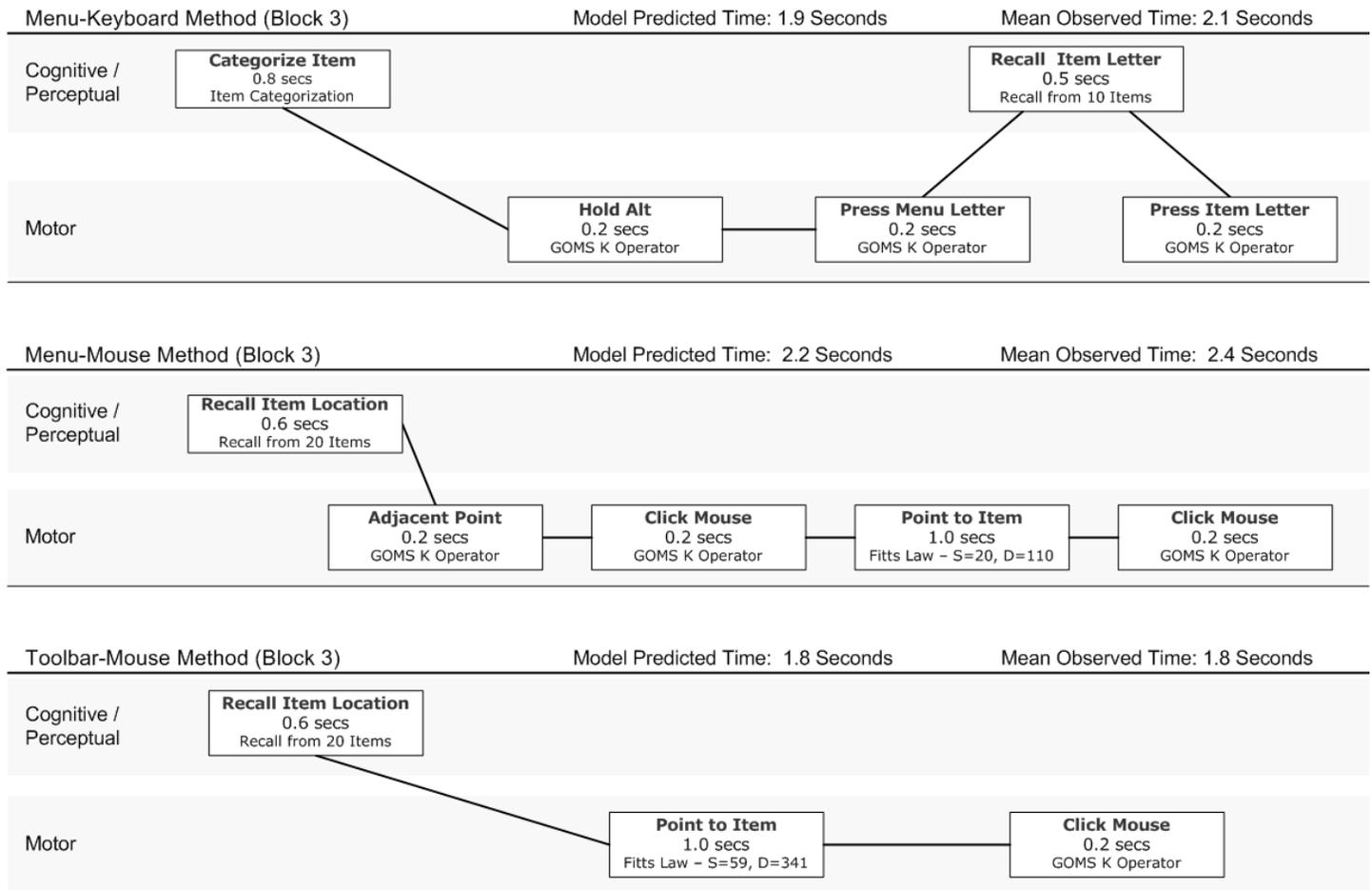


Figure 2. Models of Ending Performance on the Three Methods

times. Justifications for the time costs are presented in the remainder of this section.

The predicted time to categorize an item is based on categorization experiments by Medin and Smith (1981) who report response times near 1 second for the fastest tasks. Removing the time for the mouse click (0.2 seconds) gives an estimated time of 0.8 seconds,

The time to press a key is predicted to be 0.2 seconds, which assumes users did not have to look to find the key and corresponds to Card et al.'s (1983) K operator.

Recalling an item's letter or location is hypothesized to conform to the Hick-Hyman law of decision time (Hick, 1952; Hyman, 1953):

$$T_d = b_d * \log_2 (1/p_i) + a_d \tag{1.0}$$

In this formula,  $p_i$  is the probability of an item, which is initially  $1/n$ , where  $n$  is the number items. Normally, these probabilities change as items are selected, where selected items are more probable. However, since our task targets were randomized, the probability of an item remains  $1/n$ . Thus, we can use a simplified version of formula (1.0):

$$T_d = b_d * \log_2(n) + a_d \tag{1.1}$$

The terms  $b_d$  and  $a_d$  are empirically derived constants, which Cockburn et al. (2007) found to be 0.08 and 0.24 respectively. Using their constants, the time to select a target from a memorized location or to recall a letter associated with an item can be predicted using the following formula:

$$T_d = 0.08 * \log_2(n) + .24 \tag{1.2}$$

The predicted times derived from formula (1.2) for recalling a toolbar item location (from a set of 20) is 0.59 seconds.

The predicted time to move the mouse is estimated by Fitts Law (Fitts, 1954) using a formula suggested by Card et al. (1983, p. 241):

$$T_m = K + I * \log_2(D/S + .5) \text{ sec} \tag{2.0}$$

In this formula,  $D$  is the distance to the target,  $S$  is the size of the target, while  $K$  and  $I$  are empirically-derived constants for which Card et al. (1983) found to be 0.8 and 0.1 respectively (p. 262). Using their constants, a reasonable approximation of the time to move the mouse can be predicted using the following formula:

$$T_m = 0.8 + 0.1 * \log_2(D/S + .5) \text{ sec} \tag{2.1}$$

The predicted times derived from formula (2.1) for moving the mouse to a menu item ( $S=20$  pixels,  $D=100$  pixels) is 1.04 seconds; and for moving the mouse to a toolbar item ( $S=59$  pixels,  $D=341$  pixels) is 1.04 seconds. Since the active areas of the two menus were contiguous, the K operator (.2 seconds) was used to estimate the time to point to a menu.

## CONCLUSION

Based on previously established principles for practiced users, we have constructed a model for the last block of each method (see Figure 2), each of which predict the correct order of observed times and come within .2 seconds of the observed times.

In some circumstances, steps of a process are executed in parallel rather than sequentially (John & Kieras, 1996). For the toolbar and menu-mouse methods, each step depends on the previous step and thus does not permit any parallelism. However, the menu-keyboard method potentially permits some parallelism. In particular, users may hit the Alt key as they mentally recall the first menu selection key. Additionally, they may start to recall the second key as they physically press the first selection key. Alternatively, users may chunk the 2-key sequence as one cognitive unit.

At best, performance for the menu-keyboard menu will reach 1.2 seconds by subsuming the Alt-key press (.2s) and dropping the second letter recall (.5s). An open question is whether this improvement in performance would ultimately occur for this method. It is possible that the hierarchical menu structure makes it difficult to learn the sequence as a single unit. Further research would be useful for identifying conditions in which keyboard sequences can be learned as one cognitive chunk. In any case, the results and analysis reveal difficulties for learning hierarchical key sequences for invoking commands.

The learning and performance advantage for the mouse-based toolbar method contrasts with the findings of Jorgensen et al. (2002), Karat (1986), and Lane et al. (2005). The difference between our findings and these studies is the use of a categorized menu system which creates an additional selection process in using Alt sequences versus issuing a single keystroke or chord (i.e., control sequence) examined by the earlier studies. Indeed, our model explains why the Toolbar-Mouse method is faster – because it involves fewer selections and avoids cognitive operations requiring a categorical decision.

This work can be directly applied to practical interface design. Our findings show that a single toolbar provides the most efficient access to commands, although key-based menu access starts to be competitive after 90 trials of practice. The toolbar consequently provides the best access to commands for novices and intermediate users. For the most practiced users, our analysis suggests that toolbar access would remain competitive, except possibly under idealized learning circumstances for the key-based menu.

A user interface design may also want to consider the likely location of hands before a selection is made. For our study, users prepared the location of their hands in the optimal place: on the keyboard for the keyboard sequence and on the mouse for the toolbar selection. The “homing” cost of moving hands is estimated at .4 seconds (Card et al., 1983) and could give either method a slight edge in terms of efficiency.

Once the number of commands for a task outgrows the available space for a single toolbar, the keyboard/menu system has the clear advantage since a greater number of commands can be accessible through a keyboard Alt sequence. The menu-keyboard method is shown to be more efficient after practice when compared to the menu-mouse system of selection, so the keystroke sequences should be used to select from a large number of task-related commands.

## ACKNOWLEDGMENTS

We thank Donnie Young for his assistance in running participants in this study.

## REFERENCES

- Card, S., Moran, T. P., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Erlbaum.
- Cockburn, A., Gutwin, C., & Greenberg, S. (2007). A predictive model of menu performance. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2007*, 627-636.
- Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47, 381-391.
- Hick, W. E. (1952). On the rate of gain in information. *Quarterly Journal of Experimental Psychology*, 4, 11-26.
- Hornof, A. J., & Halverson, T. (2003). Modeling user behavior: Cognitive strategies and eye movements for searching hierarchical computer displays. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems CHI '03*, 249-256.
- Hyman, R. (1953). Stimulus information as a determinant of reaction time. *Journal of Experimental Psychology*, 45, 423-432.
- John, B. E., & Kieras, D. E. (1996). The GOMS family of user interface analysis techniques: Comparison and contrast. *ACM Transactions on Computer-Human Interactions*, 3, 320-351.
- Jorgensen, A. H., Garde, A. H., Laursen, B., & Jensen, B. R. (2002). Using mouse and keyboard under time pressure: preference, strategies, and learning. *Behavioral and Brain Sciences*, 21, 317-319.
- 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, 73-88.
- Keppel, G. (1973). *Design and analysis: A researcher's handbook*. Englewood Cliffs, NJ: Prentice-Hall.
- Lane, D. M., Napier, H. A., Peres, S. C., & Sandor, A. (2005). Hidden costs of graphical user interfaces: Failure to make the transition from menus and icon toolbars to keyboard shortcuts. *International Journal of Human-Computer Interaction*, 18, 133-144.
- Medin, D. L., & Smith, E. E. (1981). Strategies and classification learning. *Journal of Experimental Psychology: Human Learning and Memory*, 7, 241-253.
- Newell, A. & Rosenbloom, P.S. (1980). Mechanisms of skill acquisition and the Law of Practice. In J. R. Anderson, (Ed.), *Cognitive Skills and their Acquisition*, 1-51, Hillsdale, NJ: Lawrence Erlbaum.
- Pirolli, P., & Card, S. (1999). Information foraging. *Psychological Review*, 106, 643-67.