CAR-TR-764 June 1995
CS-TR-3452

Buttons vs. menus: An exploratory study of pull-down menu selection as compared to button bars

Jason Ellis, Chi Tran, Jake Ryoo, Ben Shneiderman*

Human-Computer Interaction Laboratory
Department of Computer Science
University of Maryland, College Park, MD 20742-3255
* Address correspondence to Ben Shneiderman
ben@cs.umd.edu

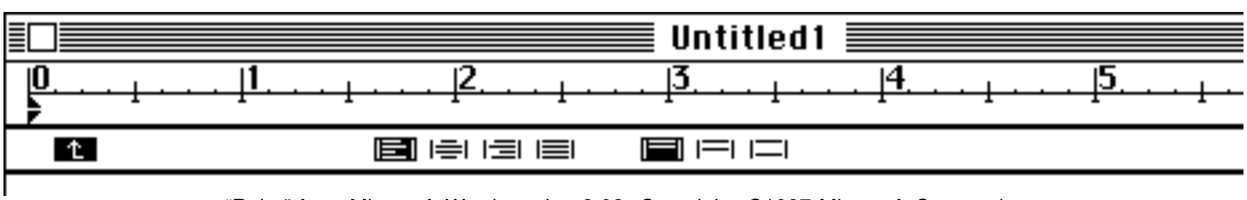
Abstract

Button bars are a relatively new interaction method intended to speed up application use as compared to pull-down menus. This exploratory study compares three command selection methods: pull-down menus, button bars, and user choice of pull-down menus or button bars. Effectiveness was measured in two ways: speed of selection and error rate. 15 participants performed 15 word processor related tasks. Results show that in frequently used functions, such as character attribute selection (bold, italic, underline, etc.), button bars are faster. There were no statistically significant differences in error rates between the three interaction methods

INTRODUCTION

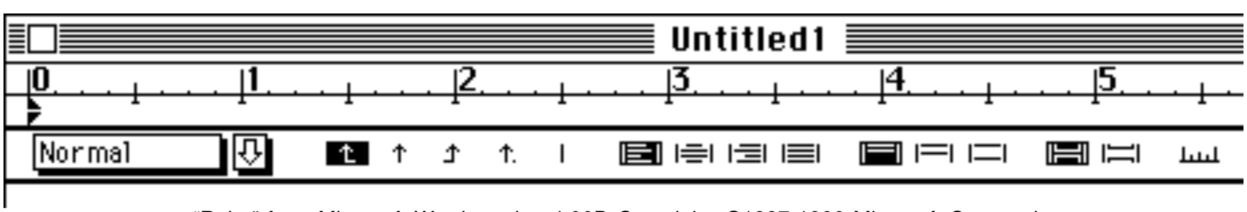
Throughout the history of computers, users have sought to get things done faster—especially repeated tasks. Shell scripts and macros were early entrants in the race to find ways to accelerate human-computer interaction. Later, character based programs began to implement crude buttons selectable via keyboard and/or mouse which were meant to be faster than accessing menus. One example can be found in early versions of the Quattro spreadsheet from Borland which allowed various commands to be activated via text-based buttons that were menu alternatives. It was the introduction of the graphical user interface, though, that really allowed button menu alternatives (usually implemented as a strip of buttons called a "button bar") to come into their own as potential interface accelerators.

Over the years, buttons bars have become a larger part of many popular programs. The Macintosh version of Microsoft Word, for example, has incorporated a button bar since version 3.02 which was released in 1987. This early version, though, only had a few formatting functions displayed as icons at the top of the window.

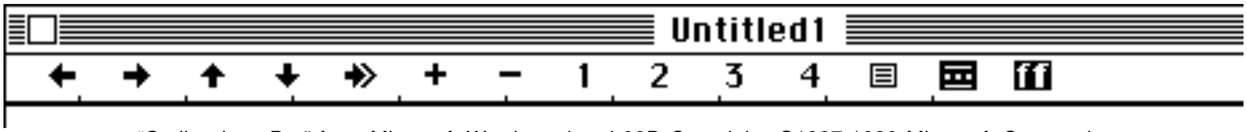


"Ruler" from Microsoft Word version 3.02 Copyright ©1987 Microsoft Corporation

Word 4.00D (1990) expanded this somewhat, adding outlining features and style sheets to the bar.

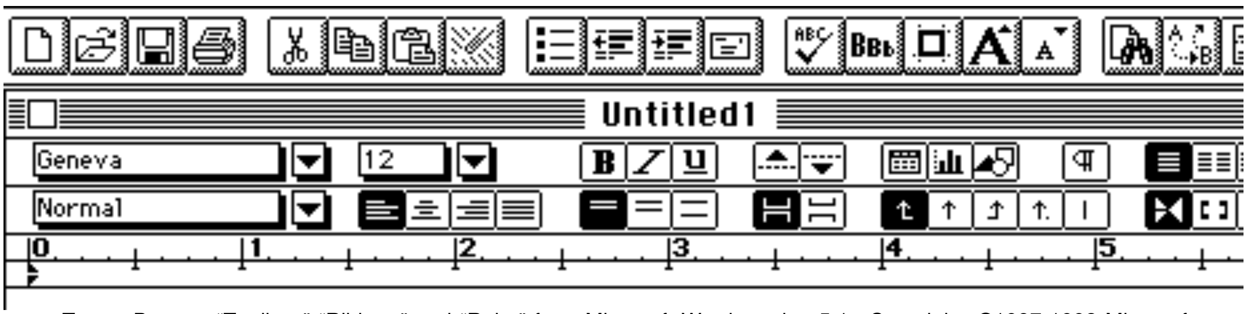


"Ruler" from Microsoft Word version 4.00D Copyright ©1987-1990 Microsoft Corporation



"Outline Icon Bar" from Microsoft Word version 4.00D Copyright ©1987-1990 Microsoft Corporation

The 5th version of MS-Word (1992) brought with it several new button bars in addition to the one attached to the ruler. These included something called a "Toolbar" which is a document-independent button bar that includes non-formatting commands.



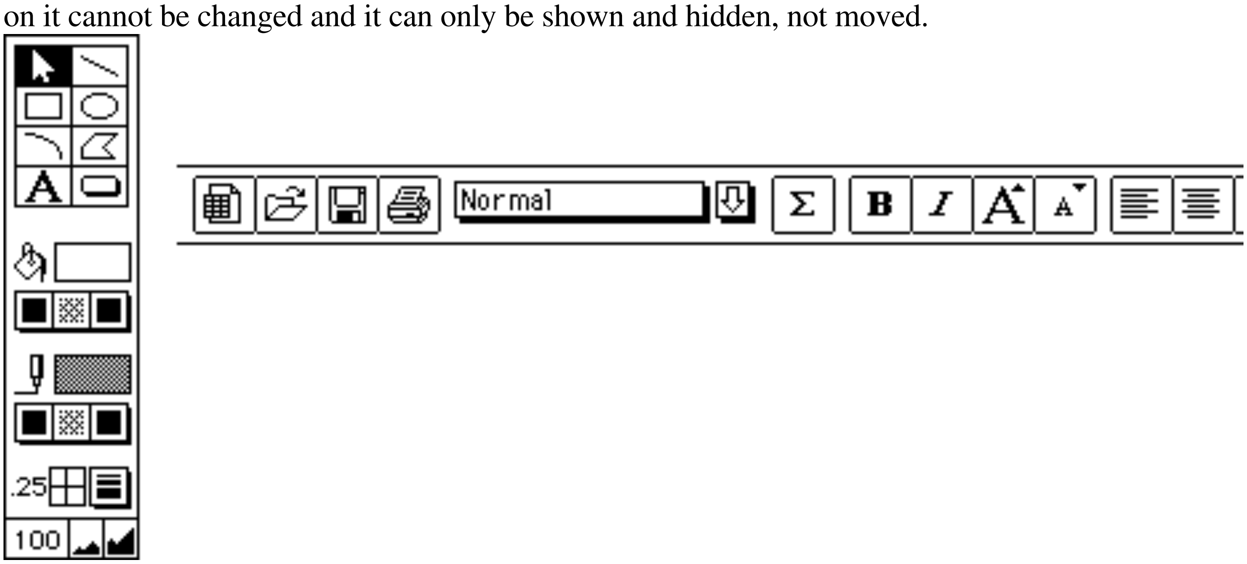
Top to Bottom: "Toolbar," "Ribbon," and "Ruler" from Microsoft Word version 5.1a Copyright ©1987-1992 Microsoft Corporation

Finally, the newest MS-Word (version 6) boasts a set of 10 different toolbars and several other button bar features. The growth of button bars as an integral part of MS-Word as well as Microsoft's other products seems to indicate that they are seen as a positive addition, at least by Microsoft's research and development staff. This theory was solidified by an e-mail response from Marshall McClintock, Usability Manager for Microsoft. "We have always held," he explains, "that toolbars and buttons are fundamentally GUI accelerators."

Many companies use buttons to draw attention to new features in their products. A new WordPerfect advertisement, for instance, is titled: "Our ruler bars give you a whole new way to look at formatting." The ad goes on to ask the reader to try out the new formatting commands available via buttons.

Beyond word processors, development products such as Microsoft's Visual C++ and BASIC, and Borland's Borland C++ allow any menu command to appear as an icon in their button bars. America Online (AOL) uses a button bar as it's primary user interface. All three major spreadsheet products (Quattro Pro, Lotus 1-2-3, and Excel) have user-configurable button bars. In short, button bars are widespread.

This does not mean, however, that each manufacturer implements button bars in the same way. Button bars are almost always used as an accelerator for menu selection, that is, the commands which appear on button bars are a subset of those on the menus. Another commonality is the use of icons to represent commands. Other factors such as the placement of the bars, the number and kind of commands which can appear as buttons, user configurability and the size of the bar itself are more variable. Microsoft Excel, for example, has a button bar which appears across the top of the screen below the menu bar. It is approximately 1 cm high and spans the entire width of the screen. Claris Resolve, on the other hand, has a button bar which appears to the left of each document. This bar's size is the same as the height of the window while it's width is a constant 2 cm. Excel's bar is more customizable as many commands on it's menus can be added and the bar will grow or shrink accordingly. The bar can also be made to appear as a vertical strip to the right or left of the screen. Resolve's button bar is less flexible. The commands on it cannot be changed and it can only be shown and hidden, not moved.



"Palette" from Claris Resolve version 1.0v2 (left) Copyright ©1991 Claris Corporation
"Toolbar" from Microsoft Excel version 4.0 (right) Copyright ©1985-1992 Microsoft Corporation

Each of these products approaches button bars with a different attitude and each has it's interesting points. Thus, with software developers including so many varieties of button bars in their new products, users are left to wonder how useful they really are.

Button bars certainly offer a new way to access commands, many times moving commands once obscured by many layers of menus to a button at the top of the screen. New users may be drawn to a company's product by advertisements which tout new features through colorful buttons. The fact that button bars can be found in almost every product today is testimony to the approval of users. However, this perceived advantage does have a cost.

Some screen space, for example, must be used by the button bar. The more buttons on the screen, the less space users have for their work. As button bars in Microsoft Word have grown from taking .75 cm of screen space to almost 2.5 cm, the workspace available to users has decreased by almost 1.75 cm. This might not seem like that much but it is enough space for two or three lines of text. If the text a user needs in order to finish a sentence at the very bottom of the screen is obscured by a bar at the top, button bars can become a hindrance.

As button bars grow over time, so does on-screen clutter. With the introduction of Microsoft Word 5, for instance, a once relatively sparse window is filled with two new rows of buttons. These new additions might prove distracting to users of the previous version who had gotten used to the more empty window it provided.

Additionally, the icons on buttons are sometimes difficult to relate to a specific command. In fact, even Marshall McClintock admits "we discovered early on that trying to develop 'intuitive' icons (that is, icons that 95% of users recognize on the first try) for abstract commands was a losing proposition." Thus, users could conceivably spend more time trying to figure out which button activates the command they want than it would take them to find the same command on a deeply nested menu.

Another problem with button bars is that many times the user's hands must leave the keyboard in order to use them. This might interrupt the user's thought process and slow them down. Even if the keyboard can be used to select buttons, there is no easy way to jump to a particular button. That is, while menus can be selected using the first letter of commands, there is no analogous method of selection for button bars.

While not much work has been done with button bars, icons (the pictures which appear on buttons) have inspired much study. Theoretically, there are three different types of icons: ones that depict the objects to be operated on, the operations or operators themselves, and the operators acting on objects (Norman, 1991). The effectiveness of tasks performed by users dealing with icons is dependent on the their ability to recognize what is depicted by the icon. Also important is their ability to recall from memory the link between the icon and the command that will be invoked or form a new link (Norman, 1991). Norman (1991) suggests that highly conventional, concrete symbols that directly depict objects and actions are more effective than those that resort to analogies or abstractions. A study by Arend, Muthig, and Wandmacher (1987) showed that selection time for textual menus and menus that use icons with abstract representations increases proportionally with menu size. Selection times for distinctive concrete icon menus, though, remained relatively constant. If designed properly an icon can be more visually distinct that words, especially in the case where there is much text cluttering (Norman, 1991).

Studies indicate that users are more likely to remember visual (icons) than verbal (text) representations (Standing et. al., 1970). A major problem with visual representations is that users must learn what the different icon symbols represent (Shneiderman, 1992). However, with continuous usage, remembering the different icons' meanings will become easier. Also, most icons directly correspond to an object and action to be performed (such as a file opening), making them all the more easy to remember (Rogers, 1989; Hemenway, 1982). Clusters of icons must be organized so that buttons with common command sets are grouped together while still being distinct within that group (Rogers, 1989). In other findings, most users preferred icons although error rates were higher for them than in other systems (Moran, 1983; Whiteside et. al., 1985).

Guastello and Traut (1989) did an experiment involving verbal (text) representations of commands as compared to pictorial (icon) representations. They found that "mixed modality metaphors" (text and picture together) were more readily recognized by the subjects than pictorial or verbal representations alone. Subjects also reported higher satisfaction when using this interaction method. Pictorial metaphors were second in recognition, followed closely by verbal representations. Additionally, they discuss the importance of icon design and command names as factors in the usability of a particular design. Specifically, they claim "correct identification of an ambiguous figure is dependent on two parameters of stimulus composition: the total amount of detail present, and the degree to which the distinguishing characteristics of closely similar figures favor one interpretation over another." The experimenters believe that mixed modality metaphors are one step towards realizing this ideal.

An experiment comparing icons and text interacted with via direct manipulation and menus was conducted by Benbasat and Todd (1993). Subjects used one of four interfaces: Icon-Direct, Icon-Menu, Text-Direct, and Text-Menu. Similar to the previous study, the experimenters here also report "little or no performance advantage for icons" as compared with text. Direct manipulation using icons or text, however, proved faster than their menu counterparts. The experimenters feel, though, that good color coding might lower response time and decrease errors for actions using icons as opposed to text.

In an experiment conducted by Kacmar and Carey (1991), the usability of icons in user interfaces was assessed. Three types of menu selection were tested: text, icon, and text and icon. Thirty subjects were randomly assigned a menu type. Then, the subjects were given a task and had to choose which menu item best represented it. For example, they might be asked to debug a program and would have to choose from the following: a text item labeled "DEBUG," an icon with a picture of a crossed out bug, and the same icon including the word "DEBUG." Although there were no significant differences between the three menu types with regard to selection times, there was evidence that text and icon menus produce the fewest errors.

Since different types of menu systems produce different levels of performance, there are apparently many cognitive processes are at work. Tulving's model (Tulving, 1971) of long term memory encoding specifically states that cues are an important factor in memory retrieval. For us, a cue would be an icon or the text in a menu. Cues are effective if they are encoded with the event, in this case the menu item is the cue and it's action is the event. Button bars have an advantage over text menu items because of their visual availability to the user. The user has easy access to the button bar whereas the text menu items are embedded under the menu headings, absent from view. Button bars facilitate recognition while text menus necessitate user recall of the menu heading under which each command is located. Thus, button bars are potentially easier and faster to use, since they rely on visual stimulants and not solely on memory recall. The reason recognition is easier than recall is that generation is not required since the original item itself is present.

Dual code theory states that memory has two forms: verbal codes and imaginal codes (Paivio, 1971). Objects that are described are stored in verbal codes, and objects that are visualized are stored in imaginal codes. According to this theory, visual images such as those in button bars produce superior results because much more detail can be retained over the verbal code.

Design guidelines that should be considered when implementing icons include (Shneiderman, 1993): (1) The object or action should be represented in a form that is familiar to the user and easy to recognize. (2) The number of icons should be limited, an icon should not be created for everything. (3) Keep in mind three dimensional objects; they are catchy but can also promote cluttering. (4) Make a single selected icon clearly visible when surrounded by the ones that are unselected. (5) Make the icon stand out from the background. (6) Make each icon distinctly different from every other icon. (6) Pay careful attention to make the icons that make up a group fit in with other members of the group. (7) Consider adding details such as color, shading and animation.

Cognitive styles of users vary greatly. Some users are comfortable using textual interfaces and others are more satisfied with iconic ones. Designers can provide the best of both worlds and present the opportunity for users to cross from one style to the other. Icons can be appropriate for certain tasks oriented in visualization, for example, using icons in paint programs immerses users in a visual environment so that they do not have to switch from visual cognition to textual cognition.

EXPLORATORY STUDY

2.1 Introduction and Hypotheses

The objective of this exploratory study was to test the effectiveness of using pull-down menus, button bar, and button bar and menus together. The measures we used to determine the effectiveness of these methods were speed of command selection and number of errors during selection. An additional measure we used was subjective user satisfaction.

To asses the effectiveness of these methods of interaction, WordPerfect 6.0 (a graphical word-processing program) for Microsoft Windows was used. The menu item choices were standard word-processing commands, such as cut, paste, open, save, spell check, etc. Button bar users selected among twenty icons, two of which were two levels deep. The text menu subjects selected among eight text menus, two to three levels deep. Finally the "both" subjects had both button bars and text menus available to choose from. The following hypotheses were formed:

Hypothesis 1: Allowing the users a choice of button bars or menus will be the fastest interaction method. Users will select whichever method will be quickest for them.

Hypothesis 2: The group which uses menus only will have the lowest error overall rate. Being textual, the menu commands will be more easily recognized than the icons on buttons.

The 15 subjects were put into three groups of five. Then one group was assigned to each interface.

2.2 Subjects

The subjects that were tested were 15 college students with the average age of 21. Three of the subjects were female and 12 were male. All subjects were familiar with word processors and graphical user interfaces but one. Some of the word processors that the subjects had used were Microsoft Word and Ami Pro.

2.3 Materials

WordPerfect 6.0a for Windows was used as the testing ground for this exploratory study. For menus only group, the layout consisted of ten text headings listed horizontally under the title of the window.

Menu bar (top) and “Power Bar” (bottom) from WordPerfect for Windows version 6.0a Copyright ©1992-1994 WordPerfect Corporation

For the button bar only configuration, a button bar with twenty buttons was added horizontally under the text menu items. The “both” setup was the same as that of the button bar only setup.

An IBM ValuePoint computer with a 15" color monitor was used. The computer consisted of an Intel 80486 CPU operating at 33MHz with 24 megabytes of RAM. This machine accessed WordPerfect from a network drive. The resolution of the screen was set at 1024x768 pixels with 256 colors. A mouse was used as the only input device for this exploratory study. In addition, a stop watch was used to record the time it took to make selections.

We used 4 forms for each subject. Each user filled out two forms: a subject selection questionnaire before the study and subject debriefing questionnaire afterwards. Each subject was asked to sign a consent form prior to the study as well.

2.4 Procedures:

2.4.1 Administration

One group of subjects was tested at a time. Subjects in a particular group were not tested in the vicinity of the other groups. Subjects were assigned to groups randomly. Before the onset of the study, each subject was given an overview of the study and instructions. A 5 minute instructional period consisted of an introduction to the menu commands to familiarize the user with the functions that they would be performing.

When the subject felt they were ready to begin the study the administrators took their places, sitting beside the participant. The first administrator held the task list and administered the tasks while the second administrator recorded the task times and errors. The first administrator was also responsible for coordinating the initiation and timing of tasks.

Each subject was given 15 tasks to complete. Subjects were required to move the mouse cursor to the lower left of the screen and then remove their hand from the mouse between tasks.

2.4.2 Grading

Before timing began, the first administrator made sure the subject and recorder were ready, and said “go” to let the participant know they could begin. The timer was started at this point and timing stopped after the correct selection was made. If the participant made an error, they were told to try again and timing did not terminate until a correct item was selected or the maximum time of 60 seconds was reached. If an error occurred which made it impossible for the participant to go on, a maximum value of 60 seconds was assigned to that particular task. After all tasks were completed, subjects were given a debriefing questionnaire to assess their views regarding the study and the interface they used.

<u>Accuracy</u>: An error was determined to be the selection of the wrong button, or menu item. It was possible to have more than one error per task since each incorrect selection counted as an error.

<u>Speed of Selection</u>: Speed of menu selection was obtained by using a stop watch with an accuracy of 1/100th of a second. Timing for each task was recorded after each successful selection.

RESULTS

The distribution of task times shows that the majority of the fastest performance times belong to the button group (Figure 1 and Table 1).

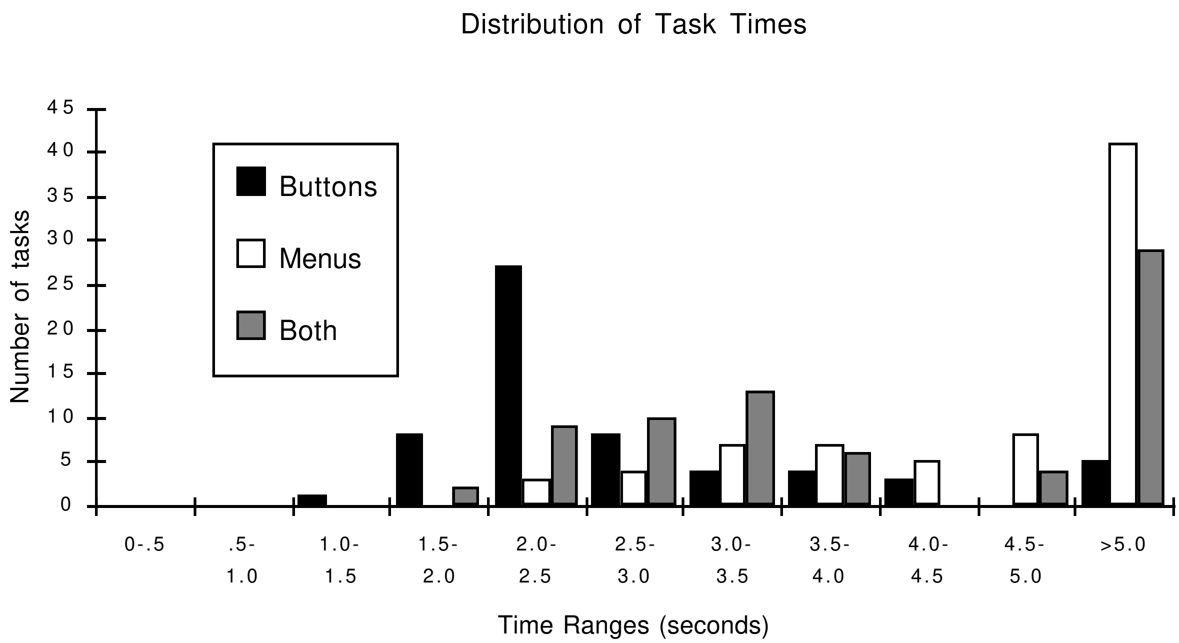


Figure 1. This chart represents the total distribution of times for all tasks in the exploratory study.

Task List

Tasks on lines without numbers were not timed.

1. Load "testfile.wpd" to edit *
   Select the line starting with "7.6.2"
2. Bold the line starting with "7.6.2" **
3. Underline the line starting with "7.6.2" **
4. Center the line starting with "7.6.2"
   Select the first sentence
5. Change the font of the first sentence of the first paragraph to Arial **
6. Italicize the first sentence of the first paragraph
7. Change the font size of the second paragraph to 6 point **
8. Undo the last task
   Select the entire document
9. Change the spacing of the entire document to double space *
10. Cut the first paragraph
11. Paste after the second paragraph
12. Spell check the document
13. Save the document (not "save as")
14. Close the document
15. Create a new document

For menus: *=stop time once dialog box appears
**=wait until selection in dialog box is made to stop time

ANOVA

				dfbet= 2			dfw= 12		**F(.05) =3.88**
	Buttons		Menus		Both		Total		
Task	Mean	Stnd Dv	Mean	Stnd Dv	Mean	Stnd Dv	Mean	Stnd Dv	***F-ratio***
1	4.19	3.92	5.60	2.13	5.46	2.92	5.09	2.92	0.32
2	**2.24**	**0.21**	**39.28**	**19.99**	**10.74**	**16.14**	**17.42**	**21.39**	**8.55**
3	**2.39**	**0.77**	**12.15**	**5.24**	**3.99**	**3.21**	**6.18**	**5.53**	**10.71**
4	8.77	6.65	14.72	4.55	11.85	10.06	11.78	7.33	0.80
5	6.80	9.26	11.34	4.80	6.79	4.46	8.31	6.46	0.80
6	**2.15**	**0.59**	**5.68**	**0.88**	**2.72**	**0.58**	**3.52**	**1.73**	**36.78**
7	16.54	22.49	6.13	0.90	8.25	6.85	10.31	13.40	0.82
8	3.24	1.17	3.44	1.25	10.72	9.36	5.80	6.23	3.01
9	**3.59**	**1.32**	**11.04**	**5.82**	**7.44**	**4.03**	**7.36**	**4.97**	**4.01**
10	2.52	0.91	6.58	6.12	3.28	1.67	4.13	3.88	1.70
11	2.76	0.82	5.78	4.08	4.16	1.55	4.23	2.70	1.73
12	3.03	0.98	3.42	1.35	5.80	6.71	4.08	3.91	0.71
13	2.90	1.94	4.16	0.66	9.07	12.01	5.38	7.07	1.07
14	4.08	3.06	4.27	0.59	4.17	1.85	4.18	1.94	0.01
15	2.77	0.78	3.37	0.56	4.05	1.96	3.40	1.29	1.29

Table 1. A listing of tasks and statistical data used to compare the three groups. The mean time and standard deviation of the task times are included for each group per task. Overall mean and standard deviations are displayed in the "Total" column.

An analysis of variance showed that four out of the fifteen tasks times had a statistically significant difference at the .05 level (Table 1). The tasks were: #2) Selecting the "Bold" function, #3) Selecting the "Underline" function, #6) Selecting the "Italicize" function, and #9) Selecting the "Change Spacing" function.

T-Test

	df=8	t(.05)=2.31	
Task	Button & Menu	Button & Both	Menu & Both
2	3.64	1.99	2.59
3	3.12	1.23	2.49
6	2.43	0.68	1.97
9	2.47	1.80	1.03

Table 2. This table displays the t-ratio for the tasks that were determined to be statistically different by the ANOVA.

T-tests showed that Buttons were statistically significantly faster than Menus at the .05 level in the four tasks which were significant (2, 3, 6, and 9). There was no statistical difference between Buttons and Both at the .05 level. The Both group, though, did perform statistically faster than the Menus group for tasks 2 and 3. The remaining tasks were insignificant among the Menu group and the Both group.

Error rates were not analyzed because they were very sparse and not interestingly high for any particular task. However, it was observed that the highest number of errors occurred in the Button bar group, having a grand total of twelve errors. The second highest errors observed fell in the Menu group with a total of five errors. Finally, the "Both" group scored three total errors. The highest error rate per task recorded was only two. In fact, for the majority of the exploratory study, subjects made no errors at all.

DISCUSSION

4.1 Interpretation of Results

Results show that subjects where much quicker at using the button bar for tasks that involved changing character styles. The bold, italicize, and underline tasks were available from buttons and required only one click to invoke the function. On the other hand, performing these tasks using a menu requires the user to travel two levels deep into the menu structure and manipulate various check boxes and radio buttons in a dialog box. It is apparent that if a user has to travel further down the menu structure, their performance time will be longer than if they had invoked the function by using the one click process facilitated by buttons.

Additionally, since the performance times were so much better, it is possible to draw some conclusions about certain button layouts. For example, character styling buttons were developed well enough so that users could identify and operate them much faster than the corresponding menu option. These buttons are properly clustered together since they all perform forms of character styling. Also, the icons include verbal cues such as B for bold, I for italicize, and U for underline. The style of the characters on the icons provide additional cues; the B is in bold, the I is italicized, and the U is underlined (see figure 2).

Figure 2.

Subjects in the group that was given a choice between buttons and menus did not perform as well as we had expected. One explanation could be that users spent extra time determining whether it was to their advantage to use buttons or menus for a particular task. Since they had a choice, they could have chosen to use buttons only and achieved approximately the same results as the button group. However, the data showed that the buttons group performed statistically faster than the menus group for four tasks (2, 3, 6 and 9) while the "both" group performed statistically faster for only two (2 and 3).

Errors were relatively uncommon. Subjects were very careful in searching for appropriate functions to accomplish a given task. There seems to be a higher rate of errors in the Button group, but the error rates were not statistically significant.

4.2 Anomalies

Some subjects had difficulty in finding functions for certain tasks because they were unsure of what they were looking for. For example, some users did not know that justification refers to the aligning of text. Therefore, it took them an extraordinary long time to find the function for centering text. Also, some subjects did not know that the "Font" menu selection deals with style attributes of characters. These problems could explain why certain tasks took longer for certain subjects. In addition, the target pool of subjects was not as uniform or as large as we had hoped.

4.3 User Satisfaction

Most subjects that used buttons said that they liked the ability to perform a frequent task quickly, but icon representations were not always easy to understand. Some subjects said that buttons made them feel they were more in control of the application because they did not have to search for a function in long, deeply nested menus. For the functions that were not easily recognized, subjects felt that it might have been easier to use the menu, since each function has a verbal representation. In order to find out what a button does, a user must look on a different part of the screen to get an explanation. Subjects also had a difficult time remembering the link between each icon and its corresponding function.

Subjects that used menus liked their organization, but they did not like traveling from menu to menu searching for a function. Some subjects said that they liked the menus because they were familiar with them. Many menu items are standard in most applications. However, others complained that some menu items were under unexpected headings.

CONCLUSIONS

5.1 Suggestions for future researchers

Future researchers should test a larger subject pool. While our subjects had roughly equivalent experience with button bar and menu interaction, there was still a large variation in performance. A larger pool would reduce the effect these variations had. Additionally, more stringent screening would allow designers to target their study for the intended users and reduce the variation further.

Our exploratory study measured the amount of time it takes new users to learn a system. Providing more extensive training and task sets might allow researchers to examine this along with the performance of more experienced users.

Researchers might want to look at the effect performing actions that do not involve button bars. For example, one possible sequence of actions is: executing a bold command, selecting a block of text, executing a justification command, and then scrolling up ten lines. In this case, we would have measured only the command selection time (i.e. time to find bold and justification commands). Other researchers might look at the other times (times for scrolling, etc.) and see if they are different between interaction methods.

In this study, the group which used buttons only was told that all the commands they would need were on the button bar. Another interesting study might be done wherein all the commands are not present on the button bar. One might test one group who could use either interaction method and another that was forced to use the menus. Since some commands would not be on the button bar, the group that had a choice would have to use menus for some commands.

Further studies should be done regarding the design of icons. No computer industry standard is yet available. Adopting standards from non-computer industries seems problematic as research has shown that icon standards in other industries are inadequate (Guastello and Traut, 1989).

Other pointing devices might be used in future studies. Testing several different pointing devices with button bars, menus, and both would add a new dimension to the study. Testing verbal icons, icon-menu combinations, and hot-menus with frequently used commands duplicated on a separate text menu are additional possibilities. Other menu shortcut methods such as keyboard commands and context-sensitive menus might be introduced as well.

Future researchers might also want to program their own interface for testing and have timings done in software. This would eliminate the error factor inherent in stopwatch timing done by humans. Errors for each task could also be kept track of this way.

5.2 Impact for practitioners

Our study found that button bars can indeed be useful under certain circumstances. Their ability to improve performance is most apparent when the meaning of each icon is clear and the menu option equivalents are deeply nested. We have looked at many button bar designs and have closely observed user interaction with one of them. Through this experience, we have arrived at the following guidelines for more effective design:

- *Consistent icons across applications:* Company-wide standardization of button icons has become fairly widespread. In the future, however, an effort should be made to make industrywide button standards. All products, regardless of manufacturer, should use the same button bar icons for similar commands.

- *Consistent button interaction:* Buttons should involve one click, not a click and drag. Users are confused by buttons which invoke menus or have other unexpected results.
- *Group together buttons which summon related commands:* Similar commands such as “cut,” “copy,” and “paste” or “new document,” “open,” “save,” and “print” should appear together.
- *Order buttons in a meaningful manner:* Users have trouble if commands appear in an unexpected sequence. Placing “print” before “open,” for instance, is disruptive.
- *Place buttons for speed:* If possible, place the most used buttons where the user looks first. For left-to-right reading cultures, for example, the most frequently used buttons should be placed in the upper left.
- *Allow for keyboard selection:* One possibility is to allow the use of number keys to select one of the first 10 icons and arrow keys to move from one to the next.
- *Allow for user configurability:* Users may want to add buttons for commands they use frequently, and remove those they don’t. They should also be able to move and hide the bar as well.
- *Avoid overly cluttering the display:* Include only the most frequently used commands as buttons. Allow for no more than one line of buttons, no more than 1 cm high. This provides for a line of buttons without using up much valuable work space.
- *Icons should be visually distinct:* Icons which look alike confuse users and slow them down.
- *Use color but be careful:* Color can be useful for grouping commands. Too much color causes distraction, though. Use it sparingly.
- *Buttons should give visual cues to their activation:* Consider changing button color or even animating icons when their button is selected.

5.3 Theory refinement

Our original hypothesis was that the use of buttons and menus together would be faster overall as users could pick which method was easiest for them. Menus, we decided, would be inherently slower as they require traversing more “depth” but we thought their textual command representation would reduce errors. Buttons, we hypothesized, would fall somewhere in-between. We now believe that buttons are the best interaction method for our tasks although they would be less useful for real world tasks where all commands are not found on the button bar.

The “user choice” group seemed to be confused by having to make a choice between using buttons or menus. Buttons were reliably faster than the menu or “user choice” groups for some tasks. Additionally, the error rates were not significantly different. Thus, we have found that buttons are the most effective means of interaction for our limited task set.

This does not mean, however, that button bars will be the fastest method of interaction in all cases. In particular, more obscure, less frequently used commands should not be placed on button bars. Only the most frequently used commands should appear on the bar or it would easily consume the entire screen. Even if screen space were not an issue, users cannot be expected to remember the thousands of commands that are available in a given program in terms of icons. Thus, we suggest that the middle ground of providing both menus and button bars would serve the user best. While our subjects seemed to have trouble deciding which interaction method to use from time to time, with experience this effect will fade.

We believe that giving the user a choice is generally the best strategy. Button bars give inexperienced users a way to get work done quickly while experienced users can delve into menus and/or customize the button bar to their style of working. For new users, a button bar only interface might be easier to understand but the memory demands inherent in this strategy would soon frustrate many of them. It is at this point, we propose, that allowing the user to choose their own method of interaction would begin to outshine the alternatives.

ACKNOWLEDGMENTS

We would like to thank the following:
- Marshall McClintock of Microsoft for his quite gracious and useful remarks.
- The Computer Science Center for allowing us to use their facilities.
- Gary Perlman and Peter Foltz for setting up HCIBIB, an extremely useful resource.
- All our subjects for being so cooperative and a special thanks to our pilot study subjects for their patience, understanding, and helpful comments.
- Anna Winkler for the original idea.

REFERENCES

Arend, U., K.P. Muthig and J. Wandmacher. “Evidence for global feature superiority in menu selection by icons.” Behaviour and Information Technology 6 (1987): 411-426.

Benbasat I. “An Experimental Investigation of Interface Design Alternatives: Icon vs. Text and Direct Manipulation vs. Menus.” International Journal of Man-Machine Studies 38 (1993): 369-375.

Ellis, Henry C. and R. Reed Hunt. Fundamentals of Cognitive Psychology. Dubuque, I. A.: Brown & Benchmark, 1993.

Guastello, Stephen J., Mary Traut and Gene Korienek. “Verbal versus Pictorial Representations of Objects in a Human-Computer Interface.” International Journal of Man-Machine Studies 31 (1989): 99-120.

Kacmar, Charles J., Jane M. Carey. “HCI Myth 1 -- ‘A Picture is Worth One Thousand Words’.” Behaviour and Information Technology 10 (1991): 443-457.

Norman, Kent L. The Psychology of Menu Selection: Designing Cognitive Control at the Human/Computer Interface. Norwood, N.J.: Ablex Publishing Corporation, 1991.

Paivio, A. Imagery and verbal processes. New York: Holt, 1971.

Rogers, Yvonne. “Icons at the Interface: Their Usefulness.” Interacting with Computers 1 (1989): 105-117.

Shneiderman, Ben. Designing the User Interface. Reading, Mass.: Addison-Wesley Publishing Company, 1993.

Tulving, E. and D. M. Thomson. “Retrieval processes in recognition memory: Effects of associative context.” Journal of Experimental Psychology 87 (1971): 116-124.