A Comparison of Graphical User Interface Widgets for Various Tasks

Todd J. Johnsgard
Novell Incorporated
1555 N Technology Way, D231
Orem, UT 84057-2399
(801) 222-5223
TJohnsgard@Novell.com

Stanley R. Page
Novell Incorporated
1555 N Technology Way, D332
Orem, UT 84057-2399
(801) 222-7016
SRPage@Novell.com

Robert D. Wilson
Novell Incorporated
1555 N Technology Way, M132
Orem, UT 84057-2399
(801) 222-4633
BWilson@Novell.com

Ronald J. Zeno
Intuitive Design Eng.
(801) 568-1973
RZeno@aol.com

The purpose of the present study was to compare user performance, accuracy and preference while using standard user interface controls or "widgets" to complete specific types of tasks. Radio buttons were significantly faster, accurate, and preferred than any other widget for the mutually exclusive selection tasks. For the non-mutually exclusive selection tasks, check boxes were significantly faster and preferred. These widgets were superior due to the fact that all possible options were initially visible. As the number of options increased, the time to complete each task also increased. A practitioner's table for selecting effective widgets for specific types of tasks is provided. Further implications for user interface design and research are discussed.

INTRODUCTION

Standard GUI Widgets

As graphical user interfaces have evolved since the design of the Xerox Star several user interface controls or "widgets" have become standard. All of the leading GUI platforms have the following widgets: check boxes, scroll bars, radio buttons, pop-up menus, and sliders. Comparison studies of these widgets across the leading GUI platforms show just how similar they have become (Marcus, 1992; Myers, 1990).

Vendors provide user interface guidelines or style guides for each of the major GUI platforms to assist software developers in the proper use of the widgets (Apple Computer, 1992; Microsoft Corporation, 1991; Open Software Foundation, 1992; IBM Corporation, 1991). These guidelines explain the appearance and general use of the widgets. The IBM Common User Access guidelines (1991) provide a table showing which widgets to use under specific circumstances. However, there are often cases where more than one widget meets all of the requirements for accomplishing a particular task. The guidelines are not helpful in deciding which widget would be best to use.

Research on Widget Use

Several studies have compared the effectiveness of different styles of user interface. Temple, Barker and Sloane, (1990) and Tombaugh, Paynter and Dillon, (1989) found that subjects performed better and preferred graphical user interfaces over text-based systems. Benbasat and Todd (1993) found no advantages to iconic representations in the interface over text and only short-term advantages for direct manipulation over menus. Whiteside, Jones, Levy, and Wixon (1985) studied subjects using systems representing command, menu and iconic interface styles. These studies compared user interface styles but did not address the effectiveness of individual user interface widgets.

One widget that has been researched extensively is the menu. The studies concerning breadth versus depth in menu systems are of particular interest to the present study (Bishu and Zhan, 1992; Kiger, 1986; Lee and MacGregor, 1985; Paap, Kenneth and Roske-Hofstrand, 1986; Seppala and Salvendy, 1985; Sisson, Parkinson and Snowberry, 1986). These studies suggest that greater breadth (more options and fewer menus) leads to better performance than greater depth (fewer options and more nested menus).

Two studies compared the effectiveness of text entry and keyboard selection methods for entering dates (Gould, Boies, Meluson, Rasamny, and Vosburgh, 1989) and making airline reservations (Greene, Gould, Boies, Meluson, and Rasamny, 1988). Both studies found text entry methods faster and preferred over selection methods.

Two recent studies compared the effectiveness of seven different widget combinations for reordering fields in a table. In the first study (Tullis and Kodimer, 1992) radio buttons and single data entry fields were significantly faster than the other methods. The second study (Tullis, 1993) demonstrated that even experienced user interface developers could not pick the best set of widgets for a particular task just using common sense.

METHOD

Two studies were conducted. The first study included 69 participants using 14 widgets to perform various tasks. The second study involved 32 participants using the same widgets and tasks plus one additional widget (pop ups). (Table 1 contains pictures of the widgets used in these studies.)

Participants

101 participants took part in the two studies. Computer operating system experience (DOS, Windows, Unix, Macintosh, or Novice) was the only demographic data collected, although it was observed that the participants represented a wide range of ages and included both men and women.

Apparatus

The tool used to collect the data was developed with ToolBook 3.0 for Windows. Each widget was created to match the appearance and functionality of those found in the operating systems that used them. Keyboard input was allowed in entry fields. Selection using the mouse was required on all other widgets to retain consistent input methods across participants.

Procedure

After a short verbal introduction the participants were directed to enter a participant id number into the system and select the operating system with which they were most familiar. For each trial an instruction screen (see Figure 1) appeared stating the problem. The participant read the instructions and then clicked on the START button. The instructions then disappeared and the widget appeared (see Figure 2). (Note Figures 1 and 2 have been cropped to save space.) Once the task was completed they clicked on an OK button which completed the trial and initiated the next trial.

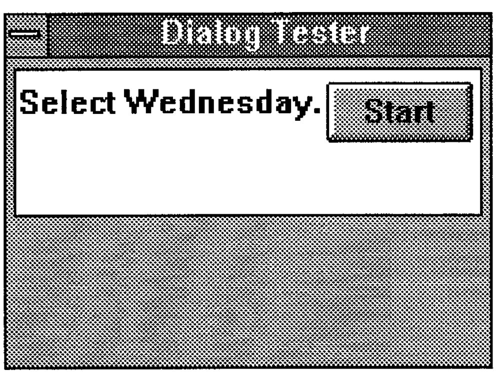

Figure 1. Sample instruction screen.

The participants completed eight practice exercises and were able to ask questions before beginning the first of three blocks of trials. At the end of each block (set of 87 trials), the participants were informed that they could pause and take a break if they chose. During the third block of the second study, the computer asked participants to rate each widget's ease-of-use.

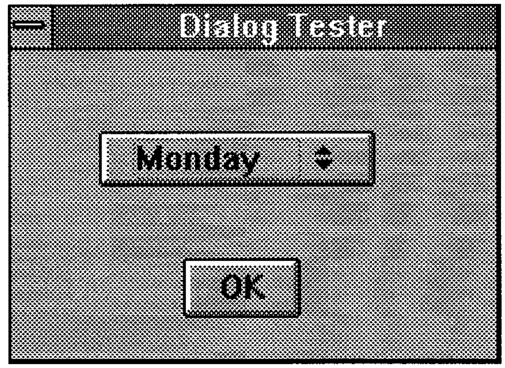

Figure 2. Sample task screen.

Design

The following data was collected:

1. Response Time: The elapsed time (milliseconds) between pressing START to begin, and pressing OK when finished, was recorded to indicate the time taken to complete the trial.
2. Errors: If an error was made, the trial was counted as incorrect regardless of the number of errors.
3. Preference: In the second study, during the third block, each participant indicated their opinion of each widget's ease of use by rating it immediately after using it. A seven point Likert scale (1 = easy to use; 7 = hard to use) was used to obtain these ratings.

The study tested the interaction between four independent variables.

1. Widget: Table 2 shows the various widgets that were used to complete the tasks.
   The following notes clarify the use of the widgets:
   - Sliders were only used in ordered cases.
   - Widgets that scroll displayed five items at a time.
   - Combo Boxes used were drop-down.
   - Multi-select lists did not require a modifier key.
2. Task: The task types used in this study are defined as follows:
   - Mutually Exclusive—participants selected one specific option.
   - Non-Mutually Exclusive—participants selected two items from the small sets, three from the medium, and eight from the large.
   - Select or Add—participants added an option when the desired option was not presented.
   - Set a Value—participants selected the time, percentage, or radio station.
3. Set Size: Each task (except for "set a value" tasks) was presented with a small set (5), a medium set (12), and a large set (30) of options to test each widget's performance in varying situations.
4. Set Order: Random and ordered sets were presented for all mutually and non-mutually exclusive tasks to determine

performance of widgets when the options were presented in a logical versus a random order.

Widget	Mutual	Non-Mutual	Select /Add	Set Value
Accumulator	·	♦	·	·
Check Box List	·	♦	·	·
Check Boxes	·	♦	·	·
Combo Box	♦	·	·	·
Edit. Combo Box	·	·	♦	·
Entry Field	♦	♦	·	♦
Multi-Select List	·	♦	·	·
Pop Up	♦	·	·	·
Radio button with other	·	·	♦	·
Radio Buttons	♦	·	·	·
Single-Select List	♦	·	·	·
Slider with Labels	♦	·	·	·
Slider with Values	♦	·	·	♦
Spin Button	♦	·	·	♦

♦ = task used; · = task not used

Table 2. Widgets used in this study according to type of task.

RESULTS AND DISCUSSION

Mutually Exclusive Widgets

The mean response times for mutually exclusive widgets appear in Figure 3.

Radio buttons were the only widget to be significantly (p<.05) faster than all other widgets. On set order, ordered sets were significantly (p<.05) faster than random sets. For set size, there were significant differences among all three set sizes (p<.05).

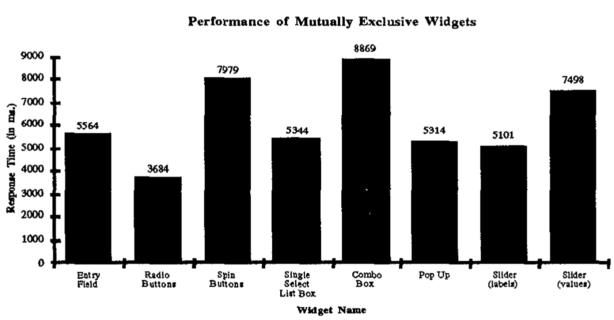

Figure 3. Mean response times for mutually exclusive widgets (in ms.).

The mutually exclusive widgets mean accuracy percentages and preferences appear in Table 3. Radio buttons were the most accurate and most preferred widget in this category. This finding is consistent with Tullis and Kodimer's (1992) experiment using various widgets to reorder table fields.

Widget	Accuracy	Preference
Entry Field	95.66%	2.69
Radio Buttons	99.34%	1.60
Spin Buttons	98.55%	3.50
Single Select List Box	95.52%	2.30
Combo Box	98.84%	2.75
Pop Up	98.85%	1.78
Slider (labels)	96.26%	2.52
Slider (values)	97.14%	2.70

Table 3. Mutually exclusive widgets mean accuracy percentages and preferences.

Non-Mutually Exclusive Widgets

The mean response times for non-mutually exclusive widgets appear in Figure 4.

Check boxes were the fastest type of widget. For set orders, the ordered sets were significantly (p<.05) faster than the random sets. The small sets were the fastest while large sets were the slowest.

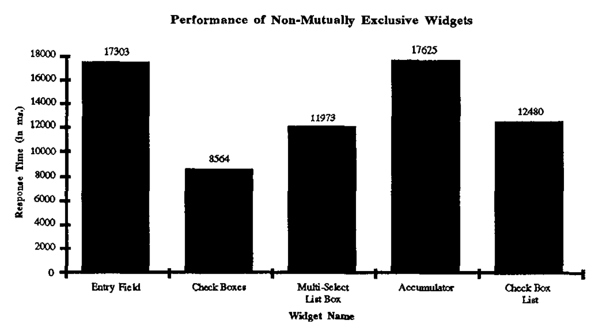

Figure 4. Mean response times for non-mutually exclusive widgets (in ms.).

The non-mutually exclusive widgets mean accuracy percentages and preferences appear in Table 3. Entry fields had lower accuracy due to typing and recall errors. This was attributed to entry fields having no set of values for users to select their response from. Check boxes were the most preferred widget in this class.

Widget	Accuracy	Preference
Entry Field	71.18%	3.85
Check Boxes	91.86%	1.54
Multi-Select List Box	93.35%	2.30
Accumulator	95.71%	3.06
Check Box List	93.73%	2.13

Table 4. Mean accuracy percentages and preferences for non-mutually exclusive widgets.

"Select From or Add Your Own Value" Widgets

Radio buttons with an entry field for other input were the fastest (9209 ms.), most accurate (94.72%), and most preferred (2.39) widget in this class. Editable Combo boxes had a mean response time of 18418 ms., accuracy of 90.98% and preference of 4.20.

"Setting a Value Within a Range" Widgets

Spin buttons were always the most accurate (98.24%) and had a mean response time (10803 ms.) plus a preference rating of 3.23. Entry fields were always the fastest (9300 ms.) and most preferred (2.29) widget in this category, but had a lower accuracy rating of 93.84%. Sliders with values were the worst overall widget in this class with a mean performance of 13335 ms., accuracy of 95.27%, and preference of 4.20.

Practitioner's Widget Table

Since the purpose of these experiments was to develop a table that user interface developers and programmers could use in software design we analyzed the data and produced Table 5 below.

Task	Best Widget	If space is tight
Mutually Exclusive	Radio Buttons	Pop-Up
Non-Mutually Exclusive	Check Boxes	Check Box List
Select or Add Own Value	Radio Buttons with other	Editable Combo Box
Setting a Value Within a Range	Spin Button	Entry Field

Table 5. Practitioner's widget table.

The guidelines used for its development were:

- Consistency — this was very important so users and developers would have a common set of widgets to use.
- No custom widgets — the table was only to include widgets that are found in today's GUI environments rather than suggesting adoption of new widgets that might confuse users.
- Brevity — a small, easy to understand table would truly be a valuable design tool.
- Tight space — that the table would consider cases when space is tight and offer a widget in these design scenarios.

Note that these guidelines apply to all set sizes and random or ordered tasks.

CONCLUSIONS

The results of this study show the importance of making options visible in a user interface. This was evident in the mutually exclusive and non-mutually exclusive tasks which included widgets with the same selection mechanism, but a different number of visible options. In these tasks, the widgets with the most visible options had the lowest performance times. These findings were similar to those of Bishu, Zhan, Sheeley, and Adams (1991). Their research showed that the best menu configuration was the one that initially showed all options.

When users had to take additional actions to make further options visible, their performance times increased. For the mutually exclusive selection task, no scrolling was required for the small set size. As a result, the performance time for radio buttons and lists had no significant difference, while the performance time for pop ups and combo boxes had no significant difference. Combo boxes and pop ups took more time because they required user activation before they displayed their options. As set size increased, performance times increased more for the widgets that required scrolling than for those that did not. For the large set, the performance time was best for the radio buttons, which always displayed all options, followed by pop ups, which displayed all options after activation, then lists, which required scrolling, and lastly combo boxes, which required scrolling after activation.

Similar results occurred in the non-mutually exclusive selection task. For the small set, there was no significant difference between performance times for check boxes, multi-selection lists, and check box lists, because no scrolling was required. When set size increased, check boxes performed better because they did not require scrolling, while there was no significant difference between performance times for multi-selection lists and check box lists.

The importance of visibility appeared in our test results in yet another case. For the non-mutually exclusive task, check boxes were the best overall, but their accuracy rate was not the best for the large set and they do not work for sets that are too large to display all options. For the large set, accumulators had the best accuracy rate but the worst performance time. We attribute the poor performance to its unique selection mechanism, and the high accuracy to the accumulator's display of selected items. We expect that in a future study, an accumulator composed of a check box list of options and a second list of selected items would be a better overall widget for large set sizes.

The results of this study demonstrate the need for further research in this area. We encourage others to perform similar studies with larger set sizes for the same tasks, and to explore other tasks. The results from the non-mutually exclusive task and the analog settings tasks showed that no single widgets tested was best in performance, accuracy, and preference. Further studies should be conducted in an effort to find better overall widgets for these tasks.

ACKNOWLEDGMENTS

The authors would like to thank Dennis Allen and Nolan Larsen for their support and encouragement throughout these studies. We would also like to thank Uhl Albert for his assistance in writing the Toolbook program. Additional thanks to Melissa Wallentine for running participants and LaRae Rahm for gathering participants. And finally, thanks to Jennifer Pugmire for her statistics assistance and to Mike Tingey for his assistance with this document.

REFERENCES

Apple Computer. (1992). *Macintosh Human Interface Guidelines*, Reading, MA: Addison-Wesley.

Benbasat, I., and Todd, P. (1993). An experimental investigation of interface design alternatives: Icon vs. text and direct manipulation vs. menus. *International Journal of Man-Machine Studies*, 38, 369-402.

Bishu, R. R., Zhan, P., Sheeley, G. A., and Adams, W. P. (1991). Depth/direction issues in menu design. *Proceeding of the Annual International Industrial Ergonomics and Safety Conference.*

Bishu, R. R., and Zhan, P. (1992). Increasing or decreasing? The menu direction effect on user performance. *Proceedings of the Human Factors Society 36th Annual Meeting*, 326-330.

Gould, J. D., Boies, S. J., Meluson, M., Rasamny, M., and Vosburgh, A. (1989). Entry and selection methods for specifying dates. *Human Factors*, 31(2), 199-214.

Greene, S., Gould, J. D., Boies, M. R., Rasamny, M., and Meluson, A. (1992). Entry and section based methods of human-computer interaction. *Human Factors*, 34(1), 97-113.

IBM Corporation. (1991). *Systems Application Architecture Common User Access Advanced Interface Design Reference* (IBM Report SC34-4290-00), Cary, NC: IBM Corporation.

Kiger, J. I. (1984). The depth/breadth trade-off in the design of menu-driven user interfaces. *International Journal of Man-Machine Studies*, 20, 201-213.

Lee, E. and MacGregor, J. (1985). Minimizing user search time in menu retrieval systems. *Human Factors*, 27(2), 157-162.

Marcus, A. (1992). *Graphic Design for Electronic Documents and User Interfaces*, Reading, MA: Addison-Wesley.

Microsoft Corporation. (1991). *The Windows Interface, an Application Design Guide*, Redmond, WA: Microsoft Press.

Myers, B. (ed.). (1990). All the widgets. *CHI '90 Special Issue: Video Review* (Video 57), Association for Computing Machinery.

Open Software Foundation. (1992). *IXI Desktop: Motif 1.2 Style Guide*, Cambridge, England: IXI Limited.

Paap, K. R. and Roske-Hofstrand, R. J. (1986). The optimal number of menu options per panel. *Human Factors*, 28 (4), 377-385.

Seppala, P. and Salvendy, G. (1985). Impact of depth of menu hierarch on performance effectiveness in a supervisory task: Computerized flexible manufacturing system. *Human Factors*, 27, 713-722.

Sisson, N., Parkinson, S., and Snowberry, K. (1986). Considerations of menu structure and communication rate for the design of computer menu displays. *International Journal of Man-Machine Studies*, 25, 479-489.

Temple, Barker and Sloane, Inc. (1990). The benefits of the graphical user interface. *Microsoft Report*, Spring 1990.

Tombaugh, J. W., Paynter, B. and Dillon, R. F. (1989) Command and graphic interfaces: User performance and satisfaction. *Designing and Using Human-Computer Interfaces and Knowledge-Based Systems*, Edited by G. Salvendy and J. J. Smith, Amsterdam: Elsevier Science Publishers, 369-375.

Tullis, T. (1993). Is user interface design just common sense? *Proceedings of the Fifth International Conference on Human-Computer Interaction*, 9-14.

Tullis, T. and Kodimer, M. (1992). A comparison of direct-manipulation, selection, and data-entry techniques for reordering fields in a table. *Proceedings of the Human Factors Society 36th Annual Meeting*, 298-302.

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

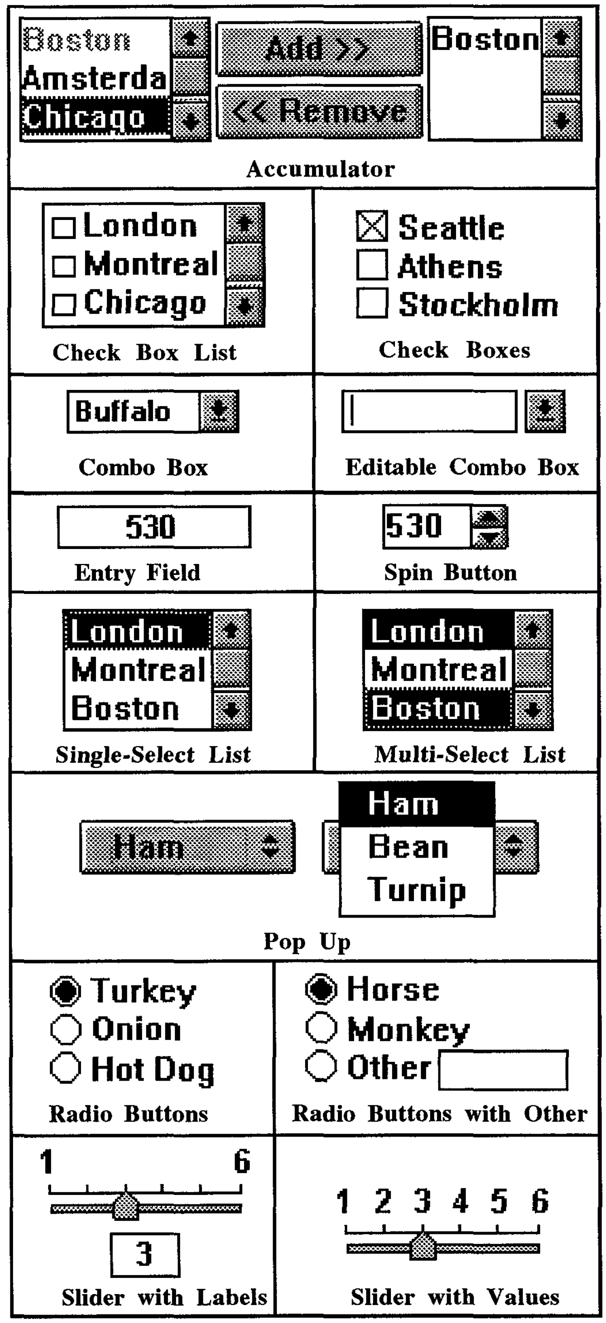

Table 1. Pictures of the widgets used in these studies.