DESIGNING MENU DISPLAY FORMAT TO MATCH INPUT DEVICE FORMAT

GARY PERLMAN
LEO C. SHERWIN

Abstract

We report the results of an experiment designed to measure the effects of modeling menu format to match the format of input devices. Subjects were presented with menus in layouts of varying compatibility with two common input devices: IBM PC function keys in a matrix format and the digit keys at the top of standard keyboards. The results showed that the better the match between formats of menus and devices, the lower the selection times. Guidelines for the design of displays suggest that the best way to show items is in a vertical sorted list, which is incompatible with the format of IBM function keys. We conclude that software designers should model menu display formats after the selection hardware.

1. Introduction

Menus are used to simplify many software user interfaces. Menus have two uses. The first is to display a set of alternatives. The second is to allow selection of those alternatives using some selection mechanism. Many current systems use a pointing device such as a mouse to select items in a menu, so that the display is also an aid to selection. Such direct manipulation interfaces are thought to promote ease of use by being cognitively more intuitive and physically more response compatible (Hutchins, Hollan, & Norman, 1986). However, primarily for economic reasons, most systems do not use pointing devices, but instead use keyboards. For keyboards, selection can be done by typing words, letters, cursor commands to move to the desired item, numbers, or function keys. Our working hypothesis is that we can approximate a direct manipulation interface by modeling the format of menus after the physical layout of the keys that will be used to select from them.

The question addressed in this study is whether compatibility of formats of menus and selector keys has an effect on selection time and accuracy. In our experiment, we looked at two common selection devices: IBM PC function keys, and the digit keys at the top of standard keyboards. Perlman (1984) reported that when prediction of items in a menu in advance was not possible, numeric selectors were the best choice for textual selection. Perlman's experiment presented subjects with vertical menus and horizontal selection keys (standard keyboard digit keys). Would Perlman's subjects have selected menu items more efficiently if the menus had been presented in the same (horizontal) format as the selection key format?

The IBM PC keyboard presents the function keys arranged in a matrix format:

1	2
3	4
5	6
7	8
9	10

IBM Function Keypad

How is the user selection time and accuracy affected when this layout is used for selecting items from a menu arranged in similar or different formats?

During the experiment reported here, subjects were presented with eight menus of the numbers 1 to 10. Six of the menus were formatted differently than the input device format, and the remaining two were designed to match the input device format. Subjects entered the key corresponding to highlighted menu items and times and errors were recorded.

2. Method

2.1 Design

Two key types were used for item entry: Digit keys and IBM PC function keys. Each subject was tested with four menu types for each key type. Two menu formats were selected: linear and matrix. There were two orientations for each menu format: horizontal and linear. Linear menu formats were displayed both horizontally and vertically. Matrix menu formats were numbered horizontally (IBM PC Function Key Layout), and vertically. The vertical matrix format was:

1	6
2	7
3	8
4	9
5	10

Non-IBM Function Keypad

The crossing of all factors produce the following conditions:

Keytype	Format	Orientation	Expected Compatibility
function	matrix	horizontal	1 good
"	"	vertical	4 bad
"	linear	horizontal	2 fair
"	"	vertical	3 poor
digit	matrix	horizontal	2 fair
"	"	vertical	3 poor
"	linear	horizontal	1 good
"	"	vertical	4 bad

The expected compatibility column in the table describes how closely we think the menu format matches the input device format. As described in the table, the horizontally oriented, matrix menu format was expected to be closely compatible to the function key layout. The horizontally oriented, linear menu format was expected to be most compatible to the digit key inputs. A vertical linear format was expected to be a poor match for IBM PC function keys because the columns are odd and even numbers, not a simple sequence. Other expectations follow from similar reasoning.

Care was taken to prevent practice effects from affecting results. Half of the subjects were tested with the function keys first, and digit keys second. The remaining subjects were tested in the opposite order. Within each group of key type conditions, the different menus were presented in an order selected using a fourth-order Latin square design to control and counter-balance order-effects and carry-over effects.

The final design consideration was fatigue and boredom. For each menu layout, each subject went through all 10 keys twice, for a total of 160 trials per subject. Counting instructions and debriefing, the experiment took 15 to 20 minutes.

2.2 Apparatus

An IBM PC with standard keyboard and hard disk was used for instruction and menu display and for collection of selection times. Menus were displayed on the screen so that distances between menu items were similar to distances between the keys that best matched their formats.

So, the matrix format had the same dimensions as the function keypad and the linear format had the same dimensions as the digit keys.

2.3 Procedure

Each subject was presented with introductory instructions to start the experiment. Subjects were instructed to make their selections as quickly and accurately as possible. Subjects were directed to use their left index finger for all selections even though most preferred their right hand. The left hand was used because the IBM PC function keys are at the extreme left of the keyboard, and it would be awkward for subjects to use the right hand. If subjects used the right hand for digit keys, and the left for function keys, then we could not make direct comparisons of digit and function keys within subjects. Subjects were instructed to use the digit key 0 for digit responses to item 10.

Subject were presented with eight blocks of trials, one for each menu layout. Each block was preceded by instructions about which keys to use. Each subject was allowed as much time as needed to read the description and to study the menu before proceeding to the trials.

Each block of trials had one menu constantly displayed on the screen. There were 20 trials per block, consisting of two permutations of the 10 keys. Each trial began with a menu item being highlighted in inverse video, and ended with a keypress. The amount of time between trials was randomly generated uniformly between two and three seconds to prevent anticipatory responses.

2.4 Subjects

Eight members of the Wang Institute community participated in the experiment. There were seven right-handed subjects. All had experience with PC's, but none were regular PC users.

3. Analyses

The 1280 data were put through a six factor analysis of variance (subjects x keytype x format x orientation x block x trial). Because of the design, in which each key was presented once per block per menu layout, we could not analyze key position number and trial number at the same time. We adopted the .01 significance level for all effects reported. There were main effects of the blocking factor and trial factor (showing that subjects got faster with practice), and there were no higher order interactions with these control factors, so they were dropped from further analyses. We then put the data through a five factor ANOVA (subjects x keytype x format x orientation x position).

4. Results

The main results of keytype, format, and orientation can be seen in Figure 1 and in the following table.

Selection Times by Conditions (msec)

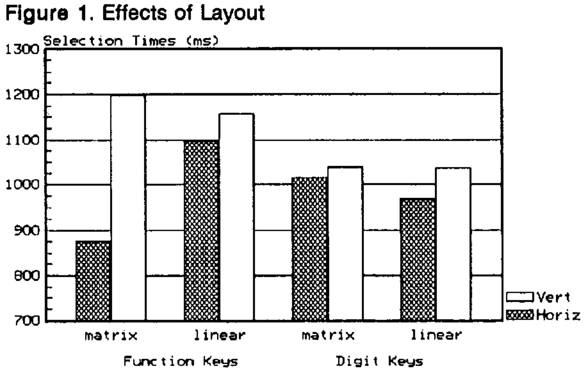
Function Keys:

	horiz	vert	
matrix	875	1198	1037
linear	1096	1158	1127
	986	1178	1082

Digit Keys:

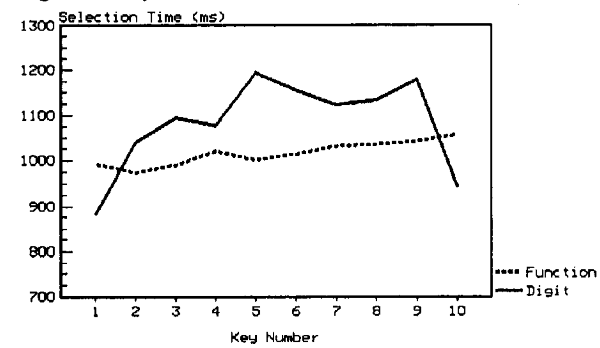
	horiz	vert	
matrix	1016	1038	1027
linear	968	1036	1002
	992	1037	1015

Figure 1. Effects of Layout



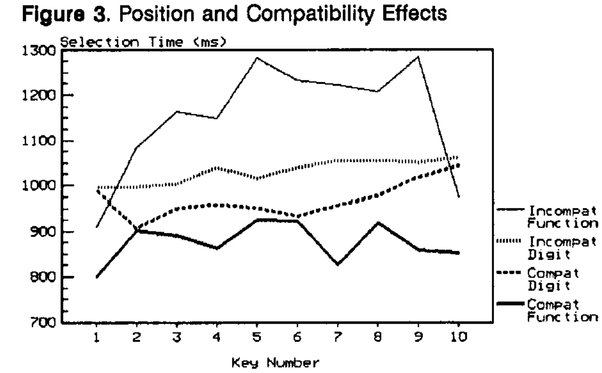
Major Effects: Selections from horizontal menus were an average of 118 msec faster than vertical menus ($F(1,7) = 48.8$, $p < .001$), and this difference was evident in all four pairing of menu layouts. Overall, digit keys were 67 msec faster than function keys ($F(1,7) = 25.6$, $p < .01$), but this is misleading because the best layout and the three worst were with function keys, with the layout most compatible with function keys 323 msec faster than the least. Figure 1 shows the keytype x format x orientation interaction. In particular, the compatible layouts (both horizontal) function-matrix (875 ms) and digit-linear (968 ms) were significantly faster than the other conditions. A Scheffe confidence interval was computed for the three-way interaction in Figure 1, and differences greater that 52 msec can be considered reliable at the 95 percent confidence level. This showed most of the means to be significantly different from all others.

Figure 2. Key Number Effects



Key Position-Number Effects: There was a significant effect of key position ($F(9,63) = 11.5$, $p < .001$), which should only be looked at with the keytype x position interaction ($F(9,63) = 7.7$, $p < .001$), because the function keys and digit keys are physically different. Figure 2 shows the selection times for the individual function and digit keys. Even this graph is over-simplified because there was a significant four-way keytype x format x orientation x position interaction ($F(9,63) = 3.3$, $p < .01$). Figure 3 shows the data for compatible

Figure 3. Position and Compatibility Effects



and incompatible function and digit keys plotted for each key position number. Recall that for each input device, there was one compatible condition and three incompatible.

Error Analysis: There were only 14 errors in 1280 trials (about one percent). With such small numbers, statistical analysis is not appropriate. There were no clear trends in the distribution of errors over conditions. These errors had almost the identical mean selection time as the correct selections, so a speed accuracy tradeoff can be ruled out. Most errors could be explained as subjects hitting the key next to the one intended.

5. Discussion

Figure 1 shows that our expectations about the effects of compatibility were largely correct. The advantage of the highly compatible function keys is not just because function keys are a better input device, because all digit key conditions were better than three of four function key conditions. For the most common menu layouts, digit keys were superior.

Figure 2 shows a small gradual increase in selection times with digit keys (less than 7 msec per digit key from left to right). This may be due to our instruction to use the left hand for all responses, and that subjects may have begun trials with their left hands near the left end of the keys. The trend for function key position is different; subjects were best with keys 1 and 10, so it seems that subjects were only able to use their knowledge of the positions of those two keys, and had to search for labels of others.

Figure 3 shows the interaction of key position number with menu layout. The superior performance for function keys 1 and 10 was not due to those keys being in the same location in both matrix format menus, because the superiority is also seen in the linear format menus. In Figure 3, the data for all menus not matching the input device are collapsed because the trends for those menus are similar. Figure 3 highlights the difference of the compatible and incompatible formats for function keys. For the compatible function key format, the differences between all keys were small, especially compared to the incompatible layouts. Some subjects reported being able to touch-type compatible function-key selections because the menu indicated the position on the keypad as well as the function key number. Such high compatibility approaches that of direct manipulation interfaces (Hutchins, Hollan, Norman, 1986).

The largest effects are in the area of 300 msec, which seem barely large enough to be perceivable, and not enough to make a great difference in productivity. Even with users typing hundreds of function keys per day, the expected gains would be only a few minutes, and on an individual basis, this does not seem important, even when multiplied by millions of users of IBM PC keyboards. But even though the effects are small, we think they are still important for system designers because many such small effects can add up to improve systems. We think that designers need this sort of information to tune their systems, and we think that archives for this information, such as the guidelines by Smith & Mosier (1986) can help system designers make use of it. Use of such reliable but small effects is related to the programming practice of code tuning (Bentley, 1982) in which, after the major algorithms have been chosen, many small improvements can be made to increase system performance by factors of two to ten.

Perhaps more important than the size of the effects are the perceptions of users. Several subjects volunteered their opinions that the compatible function key layout was their favorite and that it allowed touch-typing, or selection without reading the keys. Subjects also reported their difficulty with selections in the worst condition, function keys selecting from a vertically oriented matrix menu. However, the average difference between these extremes was less than 300 msec, and there were almost no errors for either. If users are aware of such low-level design decisions, and are irritated by bad ones, then design decisions may be made for marketability.

Smith & Mosier (1986) suggest that lists should be displayed in single columns, and that when a list is displayed in multiple columns, the items should be ordered vertically within each column because people can search such displays more efficiently. This was substantiated in the area of menu search by Backs, Walrath, and Hancock (1987). In contrast, our subjects did not have to search through menus, but only had to execute the selection demanded of them. The design of the IBM PC function keys is therefore unfortunate in several ways. Suppose you have a menu of commands, such as those from the DOS operating system for the IBM PC. If the command names are arranged in two columns, with items ordered within each column, then the function key numbers are paired as below.

1 cd	2 mkdir
3 copy	4 print
5 delete	6 rename
7 edlin	8 sort
9 find	0 type

Some PC function keys are arranged horizontally (e.g., the new IBM PC keyboard), not in a matrix, so the numbers are necessary to type the correct key. Perhaps systems should be designed to adapt their displays to match the input device on individual machines. It is important to make user interface designers understand that the constraints imposed by hardware may require reinterpretation of even the best guidelines.

Further research is needed to see if these results generalize to other (larger) keypad layouts. For example, the numeric keypad might be used instead of function keys. The work of Buxton (1986) on touch pad input is highly relevant.

6. Conclusions

For horizontal selection keys, horizontal menus should be used if search does not dominate performance, but vertical menus would perform almost as well, and are easier to search. For matrix function keypads, designers should consider using menu layouts that match the keypad layout, where the advantage for most keys is about 300 msec, and where subjective preferences also lie.

Usually, user interface software developers do not have much choice in the hardware they use. The question we have addressed with this study is how the interaction of the design of software and hardware can affect the efficiency of the user.

7. References

Backs, R. W., Walrath, L. C., & Hancock, G. A. (1987) Comparison of Horizontal and Vertical Menu Formats. Proceedings of the 31st Annual Meeting of the Human Factors Society. 715-717.

Bentley, J. L. (1982) Writing Efficient Programs. Englewood Cliffs, NJ: Prentice-Hall.

Buxton, W. (1986) There's More to Interaction than Meets the Eye: Some Issues in Manual Input. In D. A. Norman, & S. W. Draper (Eds.), User Centered System Design: New Perspectives in Human-Computer Interaction. New York: Ablex Publishing.

Hutchins, E. L., Hollan, J. D., & Norman, D. A. (1986) Direct Manipulation Interfaces. In D. A. Norman, & S. W. Draper (Eds.), User Centered System Design: New Perspectives in Human-Computer Interaction. New York: Ablex Publishing.

Perlman, G. (1985) Making the Right Choices with Menus. In B. Shackel (Ed.), Human-Computer Interaction: INTERACT '84. Amsterdam, The Netherlands: North-Holland. pp. 317-321.

Smith, S. L., & Mosier, J. N. (1986) Guidelines for Designing User Interface Software. Bedford, MA: Mitre Corporation.