50

Command/Shortcut Keys in WIMP User Interfaces: A Lost Cause?

Kee Yong LIM

School of Mechanical & Production Engineering
Nanyang Technological University
Nanyang Avenue
SINGAPORE 639798
mkylim@ntuvax.ntu.ac.sg

ABSTRACT In recent years, command/shortcut keys have proliferated exponentially in WIMP menus. However, it remains unclear whether the substantial resources spent on programming the functions of such keys, have resulted in real and/or significant benefits for user interaction. This paper reports a study to determine user interaction patterns for WIMP user interfaces and users' performance in exploiting command/shortcut keys. Subjects were served a questionnaire, a paper-based test and an on-line task to assess their preference for a particular style of interaction and their performance in respect of command/shortcut keys. The results of the study provide a new perspective for software development managers who are confronted with limited human/project resources. The work will support better decisions and resource planning in connection with value engineering, and with the effectiveness and productivity of user interface programmers.

KEYWORDS Command/Shortcut Keys, WIMP User Interfaces, Usability, Menu Design, Productivity and Effectiveness in Software Development.

1. BACKGROUND

For the past decade, graphical user interfaces have emerged as the predominant design for software applications. Today, WIMP (Windows, Icons, Menus and Pointer) user interfaces have effectively supplanted command-line user interfaces. This development is a direct endorsement of how the application of a simple human factors principle (recognition being superior to recall) could contribute significantly to the design of user friendly computer systems.

Paradoxically, there appears to be a shift in recent years, towards 'recall directed' interaction. Specifically, it may be observed that command/shortcut keys have been included rather pervasively in WIMP menus. It appears that the inclusion was made initially on the belief that the keys offer users a means of exploiting the best design features of both command line and WIMP user interfaces. This belief has gone unquestioned to the extent that command/shortcut keys have proliferated

Human–Computer Interaction: INTERACT'97 S. Howard, J. Hammond & G. Lindgaard (editors)
Published by Chapman & Hall

exponentially. Is the belief valid? Are the benefits of command/shortcut keys actually realised? Are there negative implications of increasing the number of command/shortcut keys? To address these questions, a study was conducted to determine users' interaction styles, preferences and performance with respect to their exploitation of command/shortcut keys.

2. THE STUDY

The study aims to determine:

- how well users remember different command/shortcut keys
- the extent (frequency and range) of use of particular command/shortcut keys
- the nature and frequency of errors for particular command/shortcut keys
- differences in interaction styles and performance between experts and novices in relation to command/shortcut keys
- users' preferences (if any) for particular devices to input command/shortcut keys.

The objective of the study is two-fold:

- to determine the actual effectiveness of command/shortcut keys
- to ascertain whether the (significant) resources invested by software companies to provide command/shortcut key functions have been worthwhile.

The study was conducted on two commonly used software applications. Subjects (university students) used in the test were generally computer literate ranging in age from 21 to 28 years old. 75 subjects were served a number of tests. First, a questionnaire was served to elicit their use of command/shortcut keys.

Second, the subjects were subjected to a paper-based multiple-choice answer test to determine their ability to recognise and recall the function of a set of command/shortcut keys. This test was structured into two sections as follows:

- Section 1: subjects were asked to match command names given in full (e.g. 'Save') to abbreviated representations of command/shortcut keys (e.g. 'Ctrl S').
- Section 2: subjects were asked to match abbreviated representations of command/shortcut keys (e.g. 'Ctrl S') against the correct command names which are typically indicated in full as menu items (e.g. 'Save').

The objective of the test is to ascertain the extent to which the abbreviated forms of command/shortcut keys is recognised by subjects. It is anticipated that subject performance would correlate with the appropriateness of the representations found in common software applications. In this context, 'appropriateness' was assessed according to established rules of abbreviations and semantics found in design handbooks (e.g. Helander, 1988). The complexity of the representation (e.g. length and syntax of the character string used to abbreviate the command/shortcut key) was also considered. Command/shortcut keys were thus classified into simple and complicated classes. In addition, representations associated with frequently used and/or generic command/shortcut keys were also noted. The latter classification refers to command/shortcut keys that are commonly found within one application and across different applications respectively. A third classification was also defined to uncover any optimisation by users (intentional or otherwise), of the physical effort required to use command/shortcut keys. It is conceivable that users may avoid using some command/shortcut keys not because they were unable to remember the keys, but because additional or awkward effort is required for their activation. In particular, some command/shortcut keys may require more than two-fingers and/or more than one hand to activate. It would be informative to ascertain the extent to which the economy of effort would affect users' interaction patterns. A final classification of command/shortcut keys addressed the consistency of the representations of generic commands found in different release versions of the same application, across different

applications, and across platforms. Worse, with a fast growing set of command/shortcut keys, some applications have even used the same key to represent different access functions found in different screen displays. As in the case of the economy of effort, it is conceivable that users might avoid using command/shortcut keys if a significant set of generic commands have different representations across these applications. The resulting cognitive load and frustration due to the negative transfer of learning effects, may deter a user from learning let alone use command/shortcut keys. It would be 'natural' for users to avoid a situation that engenders a sense of helplessness. It should be noted that applications which provide functions to enable users to define their preferred representations of command/shortcut keys (according to those they have already learnt), would alleviate but not solve the problem. Although enforcing consistency of representations is a solution, it is made very difficult by the large (and still growing) set of command/shortcut keys that are found in existing applications.

The motivation for such a comprehensive classification of command/shortcut keys was two-fold:

- to support richer interpretation of the data to be collected. For instance, by serving the test to subjects with different levels of experience (of the software application), the extent to which user performance may improve with learning may be determined for each class of command/shortcut keys.
- to ensure a balanced set of test questions. Thus, the test questions used will address the complete spectrum of command/shortcut keys; i.e. covering simple keys such as Ctrl S, Ctrl C, Alt E to more complicated Ctrl-Shift, Alt-Shift combinations (see Windows Interface Guidelines, 1995). Command/shortcut keys associated with common/generic functions were similarly selected to make up half the set used in the test.

To avoid potential interaction effects due to the clustering of a particular class of command/shortcut keys, test questions were also randomised. A 'don't know' response was also included as one of the multiple-choice selections to minimise reckless guessing by subjects.

To complement the questionnaire and paper-based tests, a final objective assessment was also conducted. A subset of the 75 subjects were selected at random to participate in an on-line test involving a wordprocessing task. The task entailed typing and editing of a text passage and involved a set of command functions that may be activated by mouse-based menu selections, and/or keyboard inputs. No time limit was enforced for task completion in this test.

Subject inputs were recorded and analysed to ascertain their interaction patterns and the extent of their exploitation of command/shortcut keys. The subjects were also observed to determine the extent of their (actual) preferences for a particular style of interaction, i.e. mouse-based interaction with menu items or keyboard-based interaction with command/shortcut keys.

For both paper-based and on-line tests, the nature of errors committed by subjects with respect to the recognition and use of command/shortcut keys, were also analysed.

3. RESULTS

The study yielded rather rich information. A detailed discussion of the results will be reported at a later date. The present paper will only provide a summary of some of the results.

Generally, the study indicated the following:

(1) command/shortcut keys were remembered rather poorly. The performance of subjects was unsatisfactory even for common functions such as 'Find'. In this instance, only 11% of the subjects knew the appropriate command/shortcut key. With more complicated command/shortcut keys (e.g. those involving a modifier key, see Windows Interface Guidelines, 1995), correct identification of the key functions dropped drastically to about 4%.

(2) limited use (frequency & range) of command/shortcut

keys. The results indicated that subjects tend to use only command/shortcut keys that are simple and common/generic across software applications. The results seem to suggest that combination type command/shortcut keys were used so infrequently that subjects were unable to achieve a satisfactory level of competence. In particular, the subjects knew the function of only about 10% of the keys. Later observations of their performance of a wordprocessing task revealed a limited use of access keys (e.g. keying 'O' for Open). It was observed that subjects (mostly right-handed) were unwilling to release the mouse to key in access keys and combination type command/shortcut keys. They were also uncomfortable with using the left-hand to key in commands. It appeared that input using the mouse was more 'intuitive' and economic in effort (see also (5) below). Subjects also found it more disruptive to have to shift their visual focus from the screen to the keyboard, when they wanted to locate the keys to activate the more complicated command/shortcut keys. It would be interesting for future research to examine whether touch typists would also be similarly affected. Another observation made was that subjects were more inclined (consciously or otherwise) to move from the keyboard to the mouse when they needed to activate menu items. This inclination was observed even for subjects who knew the command/shortcut keys required to activate the desired command functions. However, the converse did not apply. Instead, when the subject's hand was already resting on the mouse, they were less inclined to release the mouse to use the keyboard, even if they know the command/shortcut key. For instance, observations during the on-line test revealed that some subjects would even use the mouse to 'Save' a file rather than key in 'Ctrl S'. This inclination of subjects was observed to be more pronounced when more complicated command/shortcut keys were involved. One explanation for this observation may be that moving the mouse would incur little or no additional mental resources (see also (4) below). Conversely, for more complex tasks, it is possible that competition for the same mental resources may arise if users need to recall complicated command/shortcut keys. Thus, users would tend to choose the easier style of interaction rather than suffer a higher mental load or lower task performance.

(3) more errors for meaningless or semantically weak command/shortcut keys. Subjects encountered more problems with combination type command/shortcut keys and function keys which were complex and semantically obscure, e.g. Ctrl-Shift P, Alt E P and F5. Such command/shortcut keys may be compared with semantically 'acceptable' keys such as 'Ctrl S' for 'Save'. Observations of subject performance of the on-line task revealed slower recognition of access keys with inappropriate abbreviations (e.g. keying 'T' for 'Help Topics', see Windows Interface Guidelines, 1995). It is indeed surprising to find that such poor designs of command/shortcut keys persist despite extensive recommendations for command 'naming' (see Barnard et al., 1982; Hirsh-Pasek et al., 1982; Ehrenreich, 1985; etc.).

(4) no significant difference between expert and novice performance in the use of command/shortcut keys. Before reporting the results, it is pertinent to acknowledge that there is no absolute or universally accepted definition of an expert or novice user. For the present study, an expert was defined generally as a user who:

- is generally knowledgeable about computers
- is generally knowledgeable about software applications
- has spent more than 6 hours per week on the software application under test
- has used the software application under test for more than a year

In this context, it was observed that the recognition and recall of command/shortcut keys by expert and novice users were equally poor. Subjects identified correctly only 10% of common command/shortcut keys; did not know the functions of 80% of the keys and gave wrong answers for the rest. The results suggested that the extent of use of command/shortcut keys did not increase with higher competency in using the software. These results contradicted observations by d'Ydewalle et al. (1995) that

expert subjects used command/shortcut keys more frequently than menu selections via a mouse. A likely explanation of the difference is that subjects in the present study accumulated their expertise largely from using WIMP user interfaces only. As a result, they may be predisposed towards mouse-based interaction. In the case of d'Ydewalle et al., a significant number of the subjects 'graduated' from command-based user interfaces to WIMP user interfaces. Some of their subjects were also programmers. Thus, it may be postulated that users' past interaction experience can determine their interaction pattern/behaviour with WIMP user interfaces. For users who were exposed to WIMP user interfaces only, it appears that a form of 'transfer of learning' effect may manifest itself as a habituation or predisposition towards a particular style of interaction, i.e. mouse-based or keyboard-based interaction. The results of the study suggest that such users may go through a sequence of menu selections using the mouse, without even noticing the representations of command/shortcut keys displayed next to the menu items selected. Thus, these users may not learn the representations at all, or would only learn them after much longer exposure to the software application.

(5) overwhelming preference for mouse-based command input. The survey of subject responses indicated a negative attitude towards command/shortcut keys. Subjects reported that they used such keys rarely as they could not remember them. They also felt that inadequate guidance was given on the use of the keys. Later observations of their performance of a wordprocessing task revealed an overwhelming preference for mouse-based command input via menu selections. The ratio of mouse to keyboard input was determined to be about 88:12. This result compares well with the 90:10 ratio reported by d'Ydewalle et al. (1995). Thus, it may be concluded that mouse-based command input is by far the most dominant style of interaction for WIMP user interfaces.

4. DESIGN RECOMMENDATIONS

The results of the study indicated that for common software applications, the effectiveness of command/shortcut keys is very limited. This observation is aggravated further by the strong predisposition of users towards mouse-based interaction. This predisposition was particularly pronounced for users who have used only WIMP user interfaces since starting on computers. With the pervasive implementation of WIMP user interfaces, new and recent generations of users would fall into this category. Consequently, the use of command/shortcut keys by users would not only be limited presently, but may be expected to diminish in the near future. It may thus be concluded that the current emphasis on providing a comprehensive set of command/shortcut keys, is misguided and should be curtailed. In view of the reported worldwide shortage of IT staff, continued deployment of human resources to programme an extensive set of command/shortcut keys, is not only unproductive but highly irrational. Such a software development practice should be reviewed.

Existing user interface design guidelines for command/shortcut keys should also be revised. It is recommended that the set of command/shortcut keys provided should be small, and be confined preferably to simple and commonly used keys that are generic across software applications (see also Rosenberg and Moran, 1984). Design effort should thus be directed at defining an optimal set to balance efficiency trade-offs between mouse-based menu selection and keyboard-based input of commands (see Paap, 1988). Such a set of command/shortcut keys could be specified based on an understanding of their classifications and associated implications as described in this study. By abandoning the emphasis on providing an all-inclusive set of command/shortcut keys, it would also be easier to define more appropriate and consistent representations of command/shortcut keys. The problems reported earlier in this paper may thus be avoided.

5. REFERENCES

Barnard, P. J., Hammond, N. V., MacLean, A. and Morton, J., 1982, Learning and Remembering Interactive Commands in a Text Editing Task, Behaviour & Information Technology, 1(4), 347-358, Taylor & Francis.

d'Ydewalle, G., Leemans, J. and Rensbergen, J. V., 1995, Graphical vs Character-Based Wordprocessors: An Analysis of User Performance, Behaviour & Information Technology, 14(4), 208-214, Taylor & Francis.

Ehrenreich, L., 1985, Computer Abbreviations: Evidence and Synthesis, Human Factors, 27(2), 143-155.

Helander, M, 1988, Handbook of Human-Computer Interaction, Elsevier Science (North Holland).

Hirsh-Pasek, K., Nudelman, S. and Schneider, M. L., 1982, An Experimental Evaluation of Abbreviation Schemes in Limited Lexicons, Behaviour & Information Technology, 1(4), 359-369, Taylor & Francis.

Microsoft Windows, 1995, The Windows Interface Guidelines, A Guide for Designing Software, Microsoft Corp.

Paap, K. R., 1988, Design of Menus, In: Helander, M. (ed.), Handbook of Human Computer Interaction, 205-235, Elsevier Science.

Rosenberg, J. and Moran, T., 1984, Generic Commands, In: Shackel, B. (ed.), Human Computer Interaction - INTERACT'84, 245-249, Elsevier Science.