USER INTERFACE – KEYBOARD AND KEYSHORTCUTS

Grzegorz Sowa, Zbigniew Filutowicz

IT Institute, Academy of Management, Lodz, Poland
(gsowa, zfilutowicz)@gmail.com

Abstract

This article presents interface-related problems arising from a traditional keyboard. Some alternative suggestions for keyboard layout are discussed. Several ways of analyzing interfaces are introduced: the keystroke level model, the human processor model and cognitive walkthroughs.

Key words: user interface, keyboard, keyshortcut, layout.

1 Introduction

The ordinary keyboard has always been, and remains, the main interface between a human and a computer. For this reason, the analysis of optimal keyboard usage, which has been carried out for many years, is still important.

Despite firmly established the QWERTY standard, new layouts are constantly appearing and being tested, and it is possible that new generations of users may adopt some of these designs.

Graphic interfaces for operating systems and applications have evolved gradually from symbol systems and are still closely connected to them. The use of keyboard shortcuts to call up functions is still widespread and related to the layout of the keyboard.

2 The history of the keyboard

2.1 Discovery

The modern computer keyboard is a direct descendent of the mechanical typewriter. One of the first the typewriter makers, C.L. Sholes, also designed the keyboard layout which is still in use today. At first, he used an alphabetic layout, but it transpired that letter combinations commonly encountered in English, such as ‘th’ caused the thumbs to block the levers in the model of typewriter then in use. The study of common letter pairs in English helped to minimalise this fault. Thus, the design of the keyboard layout was more carefully thought out than is often believed. It was also patented in 1878.

2.2 Dvorak's Keyboard

The most significant competitor to the standard QWERTY layout appeared a long time ago in 1932 [12]. The essence of Dvorak's design is the positioning of the most commonly used letters in the home row, vowels on the left and consonants on the right (AOEUI-DHTNS); this enables the fingers of both hands to be used in quick succession when writing. Similarly, the next most commonly used letters are in the upper row – which is easier to reach than the lower row.

Careful research [2] has confirmed that transferring the writing process from one hand to another is much faster than striking the keys with one hand. This is backed up by the MHP (Motor Human Processor) theory. Salthouse [13] expressed it concisely: 'A character on the same hand cannot be initiated with the cognitive operator until the motor processor execution of the previous character is complete'.

Despite these theoretical advantages, Dvorak's layout has not caught on, a textbook example of an invention that was made too late [3]. The difficulty of remaking mechanical typewriters is no longer relevant, but the QWERTY standard is further entrenched by keyboard shortcuts which use the function buttons F1=F12, and the special keys Ins, Del, Home, End, PageUp, PageDown, PrtSc, ScrLk, Pause, Break, often in combination with each other or with letters.

For practical reasons (the limited size of computers and finger span), a keyboard cannot have too many keys. This limitation applies particularly to laptops (not to mention mobile devices). The most important solution to this are numbers (keys UIOJKLM) and arrows (WASD) positioned on letter keys – conveniently placed on the QWERTY keyboard and entirely randomly in the Dvorak layout. Similarly, the most popular editing shortcuts: Undo, Cut, Copy, Paste are opened through the neighbouring keys Ctrl+Z, X, C, V – but only in the QWERTY layout.

2.3 Variations on the Dvorak keyboard

Despite the lack of market success, the Dvorak keyboard is being developed. One of the projects involved adapting it for people who write in a suitable way: only with their right or left hands. Apart from disabled users, this makes sense for those whose other hand has to be occupied with something else (the mouse, for example). This idea, however, does not appear to have a future – users have become used to the alphabetic layout on mobile phones.

However, an interesting idea is R. Kaufman's suggestion [6] – a keyboard adapted for programmers working in popular languages (Pascal, C, Java, XML). The layout of the letters (in Dvorak's standard) is not important here overall, as what is more important is the positioning of all the brackets and special signs in a logical order in the upper row: (([{}(=*)+]!#), as well as the unique number layout – separating odd and even numbers (7531902468). This kind of modification of the standard keyboard appears to be realistic and useful. Whilst it is true that entering the source code is not the most difficult aspect of programming, it appears in copious amounts, so efficiency is significant here.

2.4 Programming language instructions on the keyboard

The simplicity of Basic inspired an idea which involved putting all Basic instructions on the keyboard, and this was put into practice on the Sinclair ZX Spectrum [14] computer.

Some examples: letter R: RUN, Y: RETURN, U: IF, D: DIM, F: FOR, G: GOTO, N: NEXT and so on.

Unfortunately, this was the only way for instructions to appear in the code – beginner users found this very problematic, and programming in Basic on this computer is not remembered fondly.

The obvious and most significant drawback was the practical impossibility of developing the programming language based on this concept.

2.5 Research into typing

The comparison of typing speeds has a long history. Extensive research has allowed several useful conclusions to be drawn [2]. Alongside the aforementioned advantage of alternate usage of hands, it has been established that we type words rather than letters: reading the word or at least the syllable before typing increases the typing speed considerably. The mental processes are carried out simultaneously here, and the time is radically reduced. The results of this research inspired similar treatment of the computer interface.

2.5 The significance of the keyboard

In today's world, efficient typing on a keyboard is, paradoxically, becoming less relevant as an issue. Since a huge number of people use a keyboard and type their own texts, with better or poorer skills, the efficiency of professional typists is not very significant in the larger scheme of things. Keyboards for professionals should certainly be ergonomic, but the average user values above all the standard layout and the quality of the keys. The clarity of the interface is much more important than productivity.

3 Models of use of the key interface

3.1 Modeling aims

The most obvious task when analyzing the interface is to compare existing interfaces. A precise evaluation of the applications currently in operation requires a methodical approach, general impressions are not sufficient. This is significant for the process of choosing between many applications on offer. Precise data from the analysis can be placed alongside the description in words.

Another case is to design the interface for a new application. There is no sense waiting to complete it, one should evaluate a model. Other tools are important – one should simulate something non existing yet [5, 10].

3.2 The KMS model

One of the first and best known tools for analyzing screen interfaces is the KLM-GOMS model [1]. The development of the Keystroke Level Model and Goals, Operators, Methods, Selection rules shortcuts sums up its significance clearly.

Basic interface operations have been defined as [7]:

- Keying – pressing and releasing a keyboard key (0.2 seconds)
- Pointing – pointing to an object on the screen with the mouse (1.1 s)
- Homing – moving the hand from the mouse to the keyboard (0.4 s)
- Mental preparing - thinking (1,2 s)
- Responding – waiting for the system to respond

Drawing time (or moving an object) using the mouse (the theory behind this has existed since 1954! - [4]) was then added to these basic operations, but the essence of this model has remained the same. The exact operation times are of course approximate, but can serve as initial estimates in the model evaluation of an interface. It is extremely important to note that an interface does not need to exist for such an analysis to take place – a sufficiently detailed design is enough.

9 standard steps are suggested when evaluating interface [7]:

1. Choose one or more representative task scenarios.
2. Have the design specified to the point that keystroke-level actions can be listed for the specific task scenarios.
3. For each task scenario, figure out the best way to do the task, or the way that you assume users will do it.

4. List the keystroke-level actions and the corresponding physical operators involved in doing the task.
5. If necessary, include operators in case the user must wait for the system to respond
6. Insert mental operators in case the user has to stop and think.
7. Look up the standard execution time for each operator.
8. Add up the execution times for the operators.
9. The total of the operator times is the estimated time to complete the task.

The analysis of contemporary interfaces yields one basic conclusion. The time taken to press a keyboard key is considerably shorter than the time taken to move the cursor and press a screen button using the mouse. However, beginner users in particular cannot be expected to remember numerous commands and keyboard shortcuts. The theoretical advantage of speed when using only the keyboard is obscured by the time needed for thought – for remembering or reading the instructions in order to retrieve the right command or shortcut.

For the time being, the best solution is a balance between entering commands and shortcuts on the keyboard and the graphic interface based on a scroll-down menu. Unfortunately, a large amount of shortcuts makes the use of the keyboard more difficult, as unwanted options are called up accidentally remarkably often.

The old standards are being replaced by new ones, which are developing in proportions that are hard to grasp. The Word 2007 word processor currently in use suggests a few hundred shortcuts to actions, which are of course also available on the menu. There is no standardization here, different applications use their own solutions, and there is little noticeable continuous development. We remember the Pascal word processor from our own work experience days, where the method of selecting and moving a block of text was in no way reminiscent of the current popular method Ctrl C, Ctrl-V.

For the experienced user, progress is doubtful. In the Lotus 123 spreadsheet, entering STD was enough to call up the function of calculating standard deviation, while currently in the Polish version of the Excel spreadsheet, this function is called ODCH.STAND.POPUL, and the only reasonable way of calling it up is to choose this name from the scroll-down list.

3.3 Human processor model

Even everyday experience demonstrates that we can carry out many mental and manual tasks simultaneously: watching the road in front of the car and carrying on a conversation, typing on a laptop keyboard and watching the screen, etc. Therefore models based on precise identification of a sequence of tasks and adding the times taken together are imperfect. Multimedia and the multifunctionality of many contemporary tools complicates these matters even more.

One of the solutions (to this) is Model Human Processor methodology [9] – the development of GOMS based on the assumption that perceptual, cognitive and motor tasks can be carried out simultaneously. Critical path diagrams, known from operational research, are used here to represent the relationships between tasks.

3.4 Cognitive Walkthrough

The methods mentioned above focus on measuring the time taken to carry out a task by the user. This misses the main problem. The interfaces of professional applications are practiced on for a long time and users become adept at operating even flawed ones after some time: overall the functionality of an application is more important than the quality of its interface.

In practice, there is often a different problem: will the user who sees the interface for the first time understand what he or she needs to do without becoming discouraged before the task has been completed? It is estimated that even the majority of internet shop transactions are abandoned because potential clients do not understand the procedure. Therefore, research into situations where the client does not know the structure of the task at the beginning (the number of tasks or their hierarchy) is very significant. The structure is discovered by the user during the use.

One of the research methodologies examining these kinds of users is Poulson's Cognitive Walkthrough [11].

The evaluation of the proposed interface consists of four elements:

1 Representation of the active aim of the user
2 Description on the button
3 Localisation and the picture of the button
4 Action taken after pressing

A very wide range of imperfections can interfere with the action the user takes. The label may be ambiguous, there may be several similar buttons active on the screen, etc.

4 The interface and the law

The interface cannot be hidden. The best companies' achievements are available for all to see and imitate.

An important precedent here is a well-known conflict in the 1990's between Lotus and Borland about the menu in their competing spreadsheets [8].

The beginning of the court summons explains the problem: "This appeal requires us to decide whether a computer menu command hierarchy is copyrightable subject matter. (…) Borland included in its Quattro and Quattro Pro version 1.0 programs "a *virtually identical* copy of the entire 1-2-3 menu tree."(…). In so doing, Borland did not copy any of Lotus's underlying computer code; it copied only the words and structure of Lotus's menu command hierarchy'.

In the end, the judge ruled that the structure of a menu cannot be the subject of a patent.

5 Conclusions

An application's interface is often the deciding factor for its level of success. For the user, the interface is the application.

The interface evolves very slowly, in contrast to the software it serves. In contemporary interfaces, very old solutions exist alongside new ones. The traditional keyboard layout, keyboard shortcuts, the concept of calling up menu functions through selected letters – all of this was devised a long time ago. Therefore, the design and evaluation tools which were used with old interfaces are still useful now. Currently, the analysis of interfaces focuses on simultaneous operations. In practice, however, how easy an interface is to understand is more important than how fast it is to use.

References

1. Bonnie E. J., Kieras D. E., *Using GOMS for User Interface Design and Evaluation: Which Technique?*, 1996, the Association for Computing Machinery, Inc, 10 June 1996, ftp://ftp.eecs.umich.edu/people/kieras/GOMS/Which-GOMS.pdf
2. Bonnie E. J., 1996, TYPIST: *A theory of performance in skilled typing*, Human–Computer Interaction, 11, 321-355. http://cogtool.hcii.cs.cmu.edu/trac/export/3275/docs/papers/TYPIST.pdf
3. David P. A., May, 1985, *Clio and the Economics of QWERTY* The American Economic Review, Vol. 75, No. 2, Papers and Proceedings of the Ninety-Seventh Annual Meeting of the American Economic Association., pp. 332-337. http://www.econ.ucsb.edu/~tedb/Courses/Ec100C/DavidQwerty.pdf
4. Fitts P. M., 1954, 6 (June), *The information capacity of the human motor system in controlling the amplitude of movement,* Journal of Experimental Psychology 47, pp. 381-391. Reprinted in Journal of Experimental Psychology: General, 1992, 121, 3, pp. 262–269,. http://eidetic.ai.ru.nl/egon/education/IntroCE/ literature/classics/Fitts54 Information_capacity_motor_system.pdf
5. Halverson, T., Hornof A. J., 2007, *A Minimal Model for Predicting Visual Search in Human-Computer Interaction*, CHI 2007 Proceedings, Gaze & Eye

Tracking April 28-May 3, 2007, San Jose, CA, USA www.citeserx.ist.psu.edu/ viewdoc/ download?doi=10.1.1.109.5822.pdf

6. Kaufman R., *Programmer Dvorak Keyboard Layout*, http://www.kaufmann.no/ roland/dvorak/
7. Kieras, D., 2001, *Using the Keystroke-Level Model to Estimate Execution Times*, University of Michigan, www.pitt.edu/~cmlewis/KSM.pdf
8. *Lotus Development Corp. v. Borland International Inc.* U.S. Court of Appeals, First Circuit March 9, 1995 http://digital-law-online.info/cases/ 34PQ2D1014.htm
9. Lui, Yili; Feyen, Robert; and Tsimhoni, Omer., 2006, *Queueing Network-Model Human Processor(QN-MHP): A Computational Architecture for Multitask Performance in Human-Machine Systems.* ACM Transactions on Computer-Human Interaction. Volume 13, Number 1, March 2006, pages 37-70, www.citeseerx.ist.psu.edu/viewdoc/download?doi=10.pdf
10. Miller C. S., March 14, 2003, *Relating Theory to Actual Results*, Computer Science and Human-Computer Interaction.
11. Poulson, P.G., Lewis, C., Rieman, J., and Wharton, C. (1992). *Cognitive walkthroughs: A method for theory- based evaluation of user interfaces*, International Journal of Man-Machine Studies 36, 741-773.
12. Rehr, D. *The QWERTY Connection*, http://home.earthlink.net/~dcrehr/ whyqwert.html
13. Salthouse T.A., 1986, Perceptual, *Cognitive and Motoric Aspects of Transcription Typing*, Psychological Bulletin Vol. 99 No. 3, http://portnov.com/keyboard/ 303.pdf
14. *Spectrum Emulator - Basic Usage*, www.guybrush.demon.co.uk /spectrum/docs/Basic.htm