A METHOD OF ANALYZING PERSONAL COMPUTER USE IN AN APPLICATION ENVIRONMENT

Peter J. Kennedy and James R. Lewis

International Business Machines
Boca Raton, Florida

ABSTRACT

This paper describes a program for the IBM Personal Computer (PC) which runs in the background to record and time-stamp keystrokes as operators use real PC applications. The value of the program as a human factors research tool is discussed. In addition to this, an example of a study using the program to estimate the frequency of acquisition of the Backspace and Enter keys in PC applications and a comparison of two IBM keyboards are described.

OBJECTIVE

The purpose of this paper is to make a new research tool known to the human factors community. This tool was developed to collect data on control strategies adopted by IBM Personal Computer operators who are engaged in tasks using real, not simulated applications. Also, recent studies in which it has been used by the Boca Raton Human Factors department will be discussed.

PERSPECTIVE

Before trade-offs can be made in a personal computer's design, one should know how it will be used. For example, there are numerous popular application programs which permit a wide variety of operator control strategies. One is caught in the dilemma of not knowing the actual percentage of task time the operator apportions to each subtask. To maximize the validity of the data, one should measure operator performance at the most fundamental subtask level in the application environment.

METHOD

Operator performance with various hardware devices has been measured using a personal computer as the data collection device. Until now operator's tasks for such studies were simplified abstractions of typical real world tasks. We have taken the approach of using real applications running on a personal computer. This eliminates any validity questions which may be raised by resorting to simulated applications.

Most applications have "closed code" and thus cannot be modified to allow performance data to be collected automatically. We have developed a program to intercept keystrokes and run in the "background" on the IBM Personal Computer so that its execution does not interfere with the execution of an application running in the "foreground". The program time-stamps each key press from the keyboard and saves these data on a diskette for subsequent analysis. This program is usable with many different applications and is an unobtrusive way to collect operator performance data in a real task environment. The operator's interface and control strategies are identical to the application running with an unmodified operating system. The performance of the application program is not perceptibly modified by the execution of the data collection program.

RESULTS

In the past, special test programs were written to collect accurate performance data. The test programs would allow us to analyze speed and accuracy for each device tested. Thus, we could compare operator performance with different keyboards, but not in a realistic application environment.

The program has been used successfully in our department to estimate the frequency of acquisition of the Backspace and Enter keys by IBM Personal Computer users and to compare two IBM PC keyboards. As examples, descriptions of the studies are given.

EXAMPLE 1: Frequency of Acquisition of Backspace and Enter Keys

Perspective

When keyboard developers make design trade-offs with a potential effect on usability, they need two kinds of information. One type of information is speed and error data collected experimentally. The other type is information about the relative frequency with which keys are used by personal computer users in the field. Of particular interest are the frequencies of Backspace and Enter key acquisitions. On many current keyboards, the Backspace key may be a single-wide or double-wide key, and there are a variety of Enter key shapes. One may conduct experiments to assess the performance effects of these alternatives, but the frequency of acquisition is also required to make a rational trade-off if these keys are competing for area on the keyboard.

Method

Sixteen IBM employees participated in this study. Eight were using a word processor or editor, and eight were using a spread sheet program.

The PC system used by the participant was also used for data collection. Keystrokes were captured using an assembly language program which ran in the background of the applications, and was transparent to the user. The keystroke frequencies were tabulated using a BASIC program. A key acquisition was defined to be the pressing of a key which was not immediately preceded by the pressing of the same key.

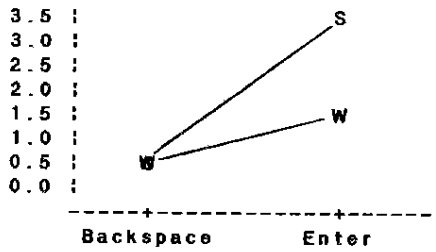
Data collectors visited different buildings at the Boca Raton IBM development site at different times of the work day. Employees observed using an IBM Personal Computer were approached in their offices and were asked for permission to record the work in which they were currently involved. They were told that their keystrokes would be recorded and the data used to help make trade-offs in future keyboard design. The type of system and application used by the subject were recorded.

Due to the possibility of interruptions during a data collection session, an additional requirement was imposed that data would be included only if the participant made a minimum of 250 keystrokes for word processing applications, or 75 keystrokes for spread sheet applications during the 10 to 15 minute session.

Results

The key frequency percentages were calculated for each participant, and are displayed in Figure 1.

Figure 1. Ave. Key Frequency %s



An analysis of variance was conducted on the independent variables of application (word processing or spread sheet) and key acquisition (Backspace or Enter) using the dependent variable of key acquisition percentage. The table below summarizes the results:

Table 1. Key Frequency ANOVA

Effect	df	F	p
Application	1,14	6.86	.02
Key Group	1,14	20.82	.0004
App X Key Grp	1,14	5.16	.04

Analysis of the interaction showed that the application groups did not differ in the percentage of Backspaces, but did differ in the percentage of Enters ($\underline{W}_n$ = 46, $\underline{p}$<.05). The ratio of Enters/Backspaces was 2.2 for word processing and 5.3 for spread sheets.

Discussion

These results indicate that the type of application used has an effect on the percentage of Enter key acquisitions, but not on the percentage of Backspace key acquisitions. If the Backspace and Enter key were competing for keyboard area for a word processing application, then compromises to the size of the Backspace key would have to be about twice as degrading to performance as compromises to the size of the Enter key to justify compromising the Enter key rather than the Backspace

key. For spread sheet applications, the Backspace compromise would have to be about five times as damaging as the Enter key compromise to justify the same decision.

EXAMPLE 2: Comparing the IBM PCAT and IBM 7531 Keyboards

Objective

The purpose of this study was to compare performance between keyboards on two IBM systems: the IBM PCAT and the IBM 7531 Industrial Computer System.

Method

Twenty-four non-IBM participants were hired to take part in this study. They represented a range of age, sex and computer experience.

Twelve of the twenty-four participants were randomly assigned to use a 7531 keyboard on their first and second days of testing. The other twelve used a PCAT keyboard on their first two days. After each person had worked for two days, they were asked to switch to the other machine for two additional days of work. Thus, each person served as his own control and order of presentation effects were counterbalanced.

Half of the 7531 participants were assigned to use a spread sheet (Lotus-123) and the other half were assigned to use a word processor (DisplayWrite-2). The same assignments were given to the PCAT participants. Sample numerical data and prose text of graded levels of difficulty were given as source material.

At the end of each day, two short timed typing tests were run. One was a 5-minute data entry task and the other was a 5-minute data editing task. The keystrokes were captured and time-stamped by an assembly language program running in the background of the application.

Results

The keying rates (mean keys/sec) are shown in Table 2, along with the results of *t*-tests. No significant differences were found between the keyboards in terms of this productivity measure.

Table 2. Keyboard Keying Rates

Application/ (Task)	7531	PCAT	t	p
Spread Sheet (Data Entry)	1.14	1.14	-0.04	.97
Spread Sheet (Data Edit)	0.86	1.07	-1.43	.19
Word Process. (Data Entry)	2.31	2.27	0.37	.72
Word Process. (Data Edit)	2.22	1.99	0.52	.62

Discussion

Since no performance differences were found for the keying rate variable, no discrimination was made about the keyboards on this point. Preference data was also collected, but a discussion of this is beyond the scope of this paper.

OVERALL DISCUSSION

Testing with simulated applications has a valid place in the spectrum of test conditions. With a simulated application, the experimenter can have a great deal of control over the user's repertoire of actions. However, real applications may allow an operator to correctly do a task in more than one way. This freedom is a characteristic of the real world, but may make the analysis of operator performance more difficult due to the experimenter's lack of control. The trade-off between control and generality is not new to the philosophy of experimental design.

The method described and demonstrated in this session may allow us to tease out some aspects of personal computer-operator interaction which were previously inaccessible. The technique is generalizable and can be useful over a wide range of tests. Using the technique, it will not usually be necessary to develop special simulated applications or to re-write data collection programs.

Although the technique was developed for keyboard testing, it can also be used to evaluate applications programs. By studying operator performance with specific tasks in different applications, we can objectively evaluate the usability characteristics of each of the applications.

ACKNOWLEDGEMENTS

The authors wish to express their appreciation to Manuel Aparacio IV and Brian C. Stanton for their efforts and invaluable contributions to the development and implementation of this research tool.

ADDITIONAL READINGS

Bradley, David J. (1984). Assembly Language Programming for the IBM Personal Computer. Englewood Cliffs, N. J.: Prentice-Hall, Inc.

Glasco, David B. and Murray Sargent III. (1983). Using IBM's marvelous keyboard. Byte, 402-415.

IBM. (1983). Personal Computer Technical Reference Manual. Boca Raton, Fl.

Lafore, Robert. (1984). Assembly Language Primer for the IBM PC & XT. New York: New American Library.

Morgan Christopher L. (1984). Bluebook of Assembly Routines for the IBM PC & XT. New York: New American Library.

Schneider, A. (1984). Fundamentals of IBM PC Assembly Language. Blue Ridge Summit, Pa: Tab Books Inc.