On-line user-computer interface—The effects of interface flexibility, terminal type, and experience on performance

by GEORGE H. WALther

U.S. Air Force Academy
Colorado Springs, Colorado

and

HAROLD F. O'NEIL, JR.

The University of Texas
Austin, Texas

INTRODUCTION

In less than two decades the electronic digital computer has evolved from a high-speed replacement for the abacus to a full-fledged partner in dialogue with humans. By its very nature, on-line computing thrusts the user into an entirely different environment than does conventional batch processing. The problems inherent in a person's being made a system component—in a sense an extension of the computer hardware—were largely ignored until quite recently. Since the early users of on-line systems were highly skilled professionals who were both willing and able to communicate in terms most convenient for the machine, few programmers were aware of the user-computer communications gap, and even fewer were concerned about it. However, the lowered cost of computer access and the proliferation of on-line systems produced a new breed of users, people whose expertise was in some area other than computer technology. As their initial fascination with conversational computing wore off, users reported experiencing feelings of intense frustration and of being "manipulated" by a seemingly unyielding, rigid, intolerant dialogue partner, and these users began disconnecting from time-sharing services at a rate which was very alarming to the industry.

Perhaps one of the best and most recent statements of the recognition by the computer industry that considerable attention should be given to the user comes from Martin[1] of the IBM Systems Research Institute:

"Increasingly..., man must become the prime focus of system design. The computer is there to serve him, to obtain information for him and to help him do his job. The ease with which he communicates with it will determine the extent to which he uses it. Whether or not he uses it powerfully will depend upon the man-machine language available to him and how well he is able to understand it (p. 3)."

A review of the literature[2] clearly indicates that a shift of emphasis is currently in progress—from a deep concern for the elegance of algorithms to varying degrees of interest in satisfying individual users. But how does one discover the "best" method for designing a user interface? Can the user simply be asked what he would like to have happen when he sits down at a terminal device? Apparently not, according to several recent writers on the subject. Further, quite contrary to popular opinion, "armchair" intuitive design techniques have not proved to be a sufficient basis for even the most concerned systems designers to use. The study being reported on is a case in point. It was the intuitive feeling of the present writers that interface flexibility would be uniformly "good" for all users, but the data did not support this contention.

Much opinion has been offered as to what constitutes user-oriented design practices, but very few intuitions have been supported with data. Bennett[2] called for a transformation of the current art of system design into an engineering discipline. The United States Air Force is requiring human factors analyses of software design in new systems. Some writers have suggested that a common user interface should be the goal, presumably leading to a set of conventions which once discovered, would maximize ease-of-use perceptions and feelings of satisfaction on the part of users. Others point out that the differences which make individuals unique overshadow commonalities, so there is considerable doubt that any interface standards would please even most of the users most of the time.

One approach, and the one chosen for this study, is to allow the interface to be alterable by the user under operating conditions without the necessity for reprogramming. Therefore, the user interface is programmed with the capability for making differential responses to a variety of users under a wide range of conditions. This is what will be referred to as interface flexibility.
PROBLEM STATEMENT AND METHODOLOGY

This study investigates whether interface flexibility is a viable solution to the problem of giving the on-line interface the quality of adaptability. The question is asked whether interface flexibility, operationalized as options offered to the user of an applications program, is "good" for everyone's performance with their program, irrespective of their personal attributes. Where flexibility appears not to be universally the best approach for all users, an attempt is made to ascertain the kind of users to whom it should be offered.

The CDC 66/6400 on-line computer system of the University of Texas at Austin was used for this research. Four terminal devices were employed: two standard Model 33-KSR Teletypes, and two Datapoint 3300 cathode ray tube (CRT) keyboard terminals. Although the CRT's and the on-line system were capable of 300-baud display rates, these terminals were buffered down to teletypewriter speeds (110-baud) so that CRT users would have no terminal display performance advantage over teletypewriter users. Terminals accessed the on-line computer via acoustic data couplings to ordinary voice-grade dial-up commercial telephone lines.

All measurements, instruction, data collection and tasks were administered by the on-line computer system of programs created expressly for this experiment. Programs were written either in SNOBOL4 or FORTRAN IV and were compiled into object code for the CDC 66/6400 system. Nine independent programs were called sequentially to administer the experiment to each user. Programs were modularized to provide re-start checkpoints in the event of system malfunctions from which full recovery was not possible and to reduce core storage requirements. A journal file was maintained on disk for each user. It contained such information as user personal data, timing statistics, verbatim transcripts of all messages issued by the user, and program instrumentation data. The applications program was instrumented to record user response time and system response time, syntax errors made by the user, counters of the frequency of use of flexibility options, and a copy of the target text file as it appeared following being edited by the user.

Sixty-nine undergraduate computer science students, all of whom had successfully completed at least the basic FORTRAN programming course and were nearing the end of either the intermediate programming course or a more advanced Survey of Programming Languages course, served as users in this study. This was the first encounter with an on-line system for about 41 percent of them. Another 38 percent claimed experience levels ranging from "once or twice before" to "several times before" (they will be referred to as users with "some experience" in this report), and 22 percent reported having "much experience."

An on-line text editor served as the experimental vehicle for the study. It was specifically designed and constructed for this experiment. Therefore, the syntax of the language of the interactive program was equally unfamiliar to all participants in the study. The syntax of the on-line editing language was a structured subset of English. Parsing was accomplished both positionally and by keyword recognition.

The first part of every command was a verb (RETRIEVE, DISPLAY, or CHANGE). The second parameter was a specifier indicating which and how many occurrences of the literal were to be considered, e.g., LAST 6, FIRST 1, EVERY, ALL. The literal came next and was identified by being enclosed between two system quotation marks (asterisks). The final command part was a file-range part indicating the portions of the file to be contextually searched or addressed, e.g., BETWEEN 10 AND 200 or IN 60. The two exceptions to this four-part syntax were (1) the DISPLAY command which required only the word LINES and a range of line numbers, and (2) the CHANGE command which of necessity included a search-target literal and a replacement literal.

An on-line tutorial program was embedded in the text-editor and had associated with it the EXPLAIN and HELP verbs.

Two different versions of the on-line text editor, an Inflexible version and a Flexible one, represented two levels of interface flexibility. In the Inflexible version, all command parts had to be spelled out fully. No mnemonics or abbreviations were permitted. Where a space was required as a delimiter, one and only one space was permitted. Every part of the syntax was essential for every command. Each statement had to be terminated with an end-of-statement character, predefined by the program to be a dollar sign.

Flexibility was operationally defined chiefly as the factors inherent in the interface program which would make it easier or more convenient for the user to express his commands, given that the commands had to follow a specific syntax. The design criteria specified that the user of the Flexible version should have as much freedom as possible in expressing his commands to the editor within the constraints of the language syntax, so long as any ambiguity as to user intent could be satisfactorily resolved by the program.

The syntax of the Flexible version was identical to that of the Inflexible with the following exceptions:

1. the ability to reset the quotation mark and end-of-statement characters. A RESET command served this function (e.g., RESET QUOTES SHORT or RESET EOS TO :).
2. the ability to abbreviate by truncation, so long as the remaining stem was unambiguous among other reserved words identified with that particular command part. The legal abbreviations for DISPLAY, for example, were: D D1 DISP DISPL DISPLA.
3. the ability to omit certain optional command parts, the meaning of which was assumed according to predefined default values. The decision logic behind selection of default values was selection of those values which, in the designer's estimation, were the values which would be most frequently intended, so long as inadvertent omission of a command part would not result in assumption of an option which would be costly in terms of excessively long displays or catastrophic inadvertent modification of the file.
4. the ability to declare equivalent synonyms for any reserved word in the language and subsequently to
abbreviate them according to the rules described in (2) above. The SYNONYM command was provided (e.g., SYNONYMS OF RETRIEVE ARE FETCH, FIND).

(5) the ability to use any number of spaces, commas, or periods as delimiters.

(6) the ability to use several equivalent forms of file-range parts. For example, these are all equivalent: BETWEEN 1 AND 10, BETWEEN 1-10, IN 1 ... 10, and just the digits 1 10.

(7) the ability to omit an end-of-statement character without the command being aborted. Instead, an advisory warning message, to the effect that one would be inserted and processing would continue, was issued.

No differences between the two versions were apparent to the user until after completion of the first editing task according to the rules of the Inflexible version. Then, prior to presentation of each of the second through fifth tasks, a new interface flexibility option was offered to the user, along with instructions and examples of how to use it. The options were introduced in the following order: (1) resetting of quotation marks or end-of-statement characters, (2) abbreviations, (3) omission of optional command parts, (4) declaration of synonyms for any reserved word.

All instructions were administered on-line by a Computer-Assisted Instruction (CAI) module which taught very basic text-editing and the use of this particular text-editor. This module was identical for users of both versions. No prior knowledge of text-editing was assumed.

The experiment involved only one session of approximately two hours' duration. Upon arrival at the CAI Laboratory at the University of Texas at Austin where testing was conducted a user was randomly assigned to one of four groups: Flexible/CRT, Flexible/Teletype, Inflexible/CRT, Inflexible/Teletype. All subsequent interactions were between the user and the computer. A proctor was available in case of system malfunction only. Events took place in this order: (1) introduction to the experiment and terminal familiarization, (2) demographic data collection, (3) typing practice and test, (4) survey of user attitudes toward the computer, using semantic differentials, (5) a five-item state anxiety scale from the State-Trait Anxiety Inventory,\(^5\) the results of which will be reported in a later publication, (7) the CAI module.

The experimental task consisted of 18 subtasks which each required that an error in a computer-based text file be corrected using the on-line text-editor. The error was described to the user, who then had to formulate the text-editing command to accomplish the correction.

After completion of the tasks, the same anxiety, attitudinal, and hostility measures were repeated. Analysis of their results appears in Walther.\(^5\)

RESULTS AND DISCUSSION

The principal data analysis employed a General Linear Models approach,\(^6\) sometimes referred to as a multiple linear regression technique. Bennett\(^5\) observed that controlled experimentation involving the user interface had produced results of questionable value because of the "unknown effect of uncontrolled variables on experimental subjects." This study capitalized on an important statistical capability of the General Linear Models approach—the ability to hold statistically constant, or fixed, all variables which were measured but which were not under experimental control, and whose effects were not specifically being investigated in a particular analysis. The net effect is the same as if experimental subjects had been selected such that they were all of the same sex, assigned to the same type of terminal device, had identical initial levels of anxiety and hostility, typed exactly at the same speed, shared in common the same amount of prior experience, and shared the same attitudes toward the computer.

In an experiment such as this, it would be obviously ideal if one could conclude that, irrespective of terminal type, sex, anxiety, attitude, typing ability, experience, or any other user attribute, it is better under all conditions to give the user a certain level of interface flexibility. However, as this study shows, things are not that simple. When the superiority of the Flexible version, for instance, depends on one or more factors other than flexibility itself, then it is said that the level of flexibility interacts with the one or more other variables, that they act in combination with each other to influence user performance.

Only those tests which were both statistically significant and relevant to this paper will be discussed. In all, there were fifty-five separate statistical tests made on the data.

Two measures of user performance are the criterion, or dependent, variables in this study: time-for-task and syntax-error-frequency.

TIME-FOR-TASK

User experience determines whether the flexible version is "better" with respect to time-for-task

Intuitively it was thought that users of the Flexible version would work faster during the editing tasks, irrespective of other factors, and that the more experienced users would be able to complete the tasks in proportionately less time.

Contrary to expectation, not all users of the Flexible version worked faster. Experience was a determinant of the level of interface flexibility which resulted in more rapid work. All users of the Flexible version worked faster than their similarly-experienced counterparts using the Inflexible version.

The more experience the user had with on-line systems, the better he was able to use the options to speed up his work. As expected, experience could not compensate for the non-availability of options in the Inflexible version, and inexperienced users worked at about the same rate with that version as those having considerable experience. Strangely enough, users of the Inflexible version and having some experience took longer than any other group to complete the editing tasks.
Flexibility options appear to help save time for all users except those having absolutely no prior experience with online systems. The Flexible version by design required less typing through the availability of abbreviations and the ability to omit optional parts. The more experienced users of the Flexible version were better equipped by virtue of their experience to take advantage of these short-cuts.

Users with no prior experience at all were probably somewhat overwhelmed with the options at a time when they had not yet become comfortable with the basic text-editing commands or with interactive computing in general. No explanation can be offered as to why the users of the Inflexible version and having some experience took so much longer than either of the other two Inflexible groups having none and much experience, respectively.

**Terminal-type and “evaluative” attitude both determine whether the flexible version is “better” with respect to time-for-task**

It was predicted that users of the same version and having the same prior experience level would work faster if they were using a CRT, even though it operated at the same speed as the teletypewriter. Although terminal type had an effect on time-for-task, the nature of its effect with respect to a particular level of flexibility was a function of the user’s initial “evaluative” attitude (how much he liked the computer and how “good” he rated it). As attitude became less positive, times-for-task increased for the Flexible/CRT group. As initial attitudes became less favorable, times decreased in the Inflexible/CRT group. There was no relationship between time and attitudes among teletypewriter users of either version.

Why should the times-for-task of only the CRT users be related to their “evaluative” attitudes? Anecdotal information provided by the users upon conclusion of the experiment indicated that CRT users felt more positive about the computer than users of the teletypewriter. For Flexible/CRT users who initially felt extremely positive about the computer, the features of the Flexible version matched their expectations of computers and probably provided sufficient incentive for them to work against the possible obstacles of greater memory demands of the CRT terminal due to a lack of any hardcopy which can be referred back to, and the unfamiliar feel of the keyboard. Those whose attitudes were somewhat neutral may have been indicating that they just were not sure about computers and did not know what to expect. For them, the combination of this uncertainty with the presentation of the options at a time when they may have just begun to feel comfortable about using the system and the text-editor, and the greater memory requirements and keyboard physical features of the CRT, may have led them into more error conditions. These users also may have had to take longer between commands to figure out what to type next. Either of these conditions would have increased their times-for-task.

Those users of the Inflexible version at a CRT and who initially were most positively inclined toward computers probably experienced a mismatch between their expectations of the system and their observation of it, to the extent that they found working with the Inflexible version to be boring, highly repetitive, and consequently they had to take greater care to avoid errors. The silent characteristics of the CRT terminal did nothing to break the monotony for them. So it is possible that their need for more deliberate, careful typing is what increased their times. Those Inflexible/CRT users who expected less from the computer may not have been disappointed with the Inflexible version and may have been sufficiently fascinated with the CRT itself that the lack of flexibility in the interface was not particularly bothersome. It is somewhat surprising, however, that the attitudes had no apparent effect on time-for-task among teletypewriter users of either version.

**SYNTAX ERRORS**

Experience and “evaluative” attitude both determine whether the flexible version is “better” with respect to syntax-error-frequency

Prior to the experiment it was thought that increasing levels of prior on-line experience would result in progressively fewer syntax errors among users of the Flexible version. The reverse effect was anticipated among users of the Inflexible version. However, the data showed that experience, alone, was an insufficient determinant of the effects of interface flexibility on syntax errors. Initial attitudes also had a strong effect on the syntax error frequency of each of the three experience levels within the Flexible group. The user’s predisposition or attitude seemed to have very little effect on the Inflexible group’s syntax errors. Less favorable attitudes seemed to predispose all but very experienced users to greater error frequencies. Where attitudes were extremely favorable, error rates were quite low, with a slightly greater error rate being noted among users of the Flexible version. However, error frequencies rose sharply among these users as their attitudes were determined to be less favorable. The exceptions were high experience users of the Flexible version whose error rate, though very restricted in range by comparison with other users, tended to be higher where attitudes were the most favorable or positive. Those users who, on the basis of their brief encounter with the experimental on-line system, had developed highly favorable attitudes concerning the computer apparently liked the options subsequently given them in the Flexible version and were able to take advantage of these options without undue syntax errors. Those users having neutral or negative feelings about the computer and having none or only minimal prior experience seemed to be predisposed to making more syntax errors. Possibly, these users’ lower expectations concerning the computer were consonant with their earlier discovery that the flexibility options could get them into trouble or would require greater mental effort until they achieved mastery in the use of the options.
It is possible that those in the Much Experience group, who also liked the computer very much, are the users who were encouraged to experiment and to “toy around” with the flexibility options. The journals of some of these people indicate this to be the case. They probably wanted to test the system, were possibly Computer Science majors at the university, and saw this as an opportunity for creativity. This kind of trial-and-error behavior would certainly result in a high incidence of syntax errors initially.

Irrespective of their attitude toward the computer, those totally inexperienced users in the Inflexible group made virtually no syntax errors at all. They had a version of the editor that was straightforward, easy to use, and simple to learn.

Their complete lack of experience and the fact that this was a “one-shot” encounter with the system gave them no basis for becoming either annoyed or frustrated, even in the face of the rigid constraints imposed on them by the Inflexible interface program.

**Terminal-type and “evaluative” attitude both determine whether the flexible version is “better” with respect to syntax-error-frequency**

Within each version of the editor, it was thought that users of the CRT terminals would make somewhat fewer errors than users of the mechanical teletypewriter terminals. However, within each version, CRT users made more syntax errors than the others, with a linear, positive relationship between syntax-error-frequency and “evaluative” attitude.

As noted earlier, there was a very sharp increase in syntax-error-frequency among users of the Flexible version as these users’ attitudes were observed to become less favorable. This was found to be true of users of both types of terminals. Therefore, attitude is a good predictor of syntax-error-frequency among users of the Flexible version, due to its high correlation with error-rate, but it is less helpful in predicting syntax errors of users without access to flexibility.

The chief differences between the two types of terminal devices are (1) the greater memory demand placed on CRT users with information at the top of the screen is erased as a new line is written at the bottom, (2) completely silent operation of the CRT compared to relatively noisy teletypewriters, (3) more modernistic-looking CRT consoles which may be perceived as being less mechanical in appearance, and (4) easy-to-press electronic action keyboards on the CRT and the mechanical linkage action of a teletypewriter. It is unclear which of these differences gave the CRT an inferior position with respect to user-performance.

The fact that the teletypewriter gave the users access to all previous dialogue transcripts, the fact that there was a one-to-one correspondence between a key-press and an audible click as the character was printed (keeping the user from having to look up at the display to see if something happened), and the possibility that the action of the teletypewriter keyboard was more similar to the “feel” of manual typewriters with which undergraduates may have been more experienced, all probably helped the teletypewriter users to work faster and more accurately.

**CONCLUSION**

The results clearly indicate that interface flexibility is not uniformly effective with all users in optimizing performance. In a single encounter with the on-line system, users are more prone to make syntax errors if offered short-cut flexibility options. Nevertheless, most all users of the Flexible version worked significantly faster than those not having the options. The exceptions were the novices who worked more rapidly without the options than with them.

A user’s prior on-line experience is a sufficient basis for deciding which interface to offer him if it is important to minimize length of an on-line session. If he has no prior on-line experience, he will work faster in his first session without options being offered. If he has any prior experience at all, he will need less time if offered flexibility options.

Neither a CRT nor a teletypewriter always works better with any one level of flexibility in minimizing time-for-task. If only teletypewriters are available, there is no need to determine the user’s “evaluative” attitude toward the computer in order to keep sessions as brief as possible. Using a teletypewriter, neither the presence nor absence of flexibility options will result in significantly different times. However, where a CRT is involved—even one that operates at typewriter speeds, the user’s attitudes toward the computer become a critical factor in predicting time-for-task.

The experience factor alone is also an insufficient basis for determining whether flexibility will minimize syntax errors in a single session with the computer. In general, users having access to flexibility options made many times more syntax errors. However, the less experienced users who had favorable attitudes made very few errors.

If duration of a session is critical, flexibility can be very instrumental in facilitating more rapid work for almost all users. But, if the lack of syntax errors at a first session is the more important criterion, flexibility should be offered only to the less experienced users with very favorable attitudes, and to experienced users with a neutral or negative attitude toward the computer. The more experienced user with positive attitudes would be expected to make progressively fewer syntax errors in subsequent sessions as his temptation to experiment declined.

The data suggest that silent-keyboard terminal devices without hardcopy capability increase the likelihood of errors. The one initial measure which never seemed to make any difference was typing ability, confirming Morrill’s findings that professional typing skills are not necessary for effective system use if the inputs are short and direct.

This study also confirmed the findings of Carlisle that more errors are committed by users of a CRT than by teletypewriter users.

Future research in this area should involve multiple sessions by the same users to see how performance is affected over time.
Interface flexibility indeed appears to be a viable solution to the problem of giving the on-line interface the quality of adaptability. Options offered the user of an applications program in the manner in which he expresses his commands to the computer are not uniformly "good" for everyone's performance, and we now have a beginning basis for predicting those kinds of users for whom it is the best approach.

REFERENCES