DESIGN OF USER FRIENDLY INTERACTIVE INTERFACES

by

JOHN FRANK YORK

B. S., ILLINOIS WESLEYAN UNIVERSITY, 1966

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1982

Approved by:

Major Professor

TABLE OF CONTENTS

LIST OF FIGURES

1.0 INTRODUCTION

The purpose of this report is to examine some of the principles and issues in designing a user-friendly interface for an interactive system. With the expanding use of computers, especially individually owned or used mirco-computers, more and more users are no longer highly trained computer personnel, but rather a cross-section of novice and experienced users. The user-friendly interface is that part of an interface that makes life easier for the user while trying to solve a problem. The design of a friendly interface allows both groups to use an interface, the first learns while using, the second performs the job efficiently.

The first part of this report will discuss some of the principles and issues that are important in the design of user-friendly interfaces. Although these interfaces include machine design of the user hardware, this report will be limited to the interface created by a portable applications program used on some existing hardware. The main design principle is that an interface allows anyone to use it regardless of their level of experience on either computers or the interface itself. Friendly interfaces should be designed or implemented so the novice can enter the interface and use it. At the same time, the experienced user should not be hindered by the aids needed by the novice.

The second part of this report will analyze an existing user interface for its degree of friendliness and offer

suggestions for improving its friendliness. The analysis gives examples on improving friendliness so that all levels of users can interact with the system in an efficient manner. The analysis will apply the principles and issues discussed in the first section to improve the friendliness of this system.

2.0 BACKGROUND

The user interface is that place where human and machine meet. The driver of a car interfaces with his car by using the steering wheel and various pedals. In the domain of computers, the interface for an interactive system consists of a key board, cathode ray tube (CRT), fingers, and eyes. The human gives the computer commands or instructions by typing them in and the machine returns the messages on the CRT. Just like he speedometer of the car tells the speed of the car, the terminal messages tell the user the exact state of the computer. As Schofield [33] stated the user interface consists of "all messages that can pass between the user and the machine and the conditions under which they can occur." Another definition of an interface is one provided by Moran [20], which is that the user interface consists of "those aspects of the system that the user comes in contact with physically, perceptually, or conceptually."

The attributes of user friendliness can not be defined as accurately as the interface. However, the opposite of friendliness, unfriendliness, of an interface can be described much better. As indicated by Gruenberger [9], "user-hostile" can be defined accurately and with a great depth of examples after six months experience in the computer field. Everyone can come up with examples of a user-hostile interface. How does the message "ERROR IN 24567" help a novice user? Even worse, how about just "ERROR"? For many, using a computer is difficult because of the lack of information on which to act. Typically, the

interface is designed for the experienced user and not the novice who is usually forced to fumble through the system using a weighty users' manual and to ask questions from whomever is around.

In summary, by having the machine supply all the necessary information for moving from one command to the next, the user can concentrate all energies toward problem solution. The friendliness of an interface cannot have an absolute value, since the interface is judged for friendliness by humans each having a different opinion of what exactly makes an interface friendly. These opinions are based upon a multitude of factors, all of which cannot be incorporated into the design of an interface. For this report, the friendliness of an interface will be the relative ease by which any person can use an interface without refering to a users' manual or some other non-machine aid. The interface should be fully capable of taking a user entirely through the system on its own without hindering the professional.

3.0 HARDWARE/ENVIRONMENT ISSUES

Hardware includes all physical parts of the computer. For an interactive interface this is basially a terminal producing softcopy used to pass messages back and forth between the user and the computer. Many aspects of the physical design of machines need to be studied. One example is the design of the keyboard, which presents many difficult problems. For example, the location of the characters with respect to each other, spacing between keys, glare of the key, method of activating keys (heat sense, physical pressure, light sensitive), shape of key, slope of keyboard, are all elements for consideration. Hardware design is a complex field in its own right and beyond the scope of this report. An even larger area, which encompasses hardware, is the users' working environment. This too is beyond the scope of this report.

Since interfaces are usually software packages used by many people on a wide variety of computer systems, the interface must be able to be implemented on a wide variety of machines. This implies that the software package will not have any control over the types of terminals used and and the environment in which the package is used.

Another hardware issue that has an effect on a user-friendly interface is the response time of the machine. As pointed out by Shniederman [35], response time of the interface has several effects. First, if the users are conditioned to an immediate response and do not get it they become concerned. The interface should provide some

messages if a particular operation is to take a long time. Second, variability in response time generates poorer performance and low user satisfaction. Consistent response times, up to a certain limit, are more important than just fast response times. Fast response times might even degrade user performance because the user may try to match machine speed and fail to check input leading to even more errors and frustration.

4.0 PSYCHOLOGICAL ISSUES

An examination of the psychological issues is necessary in the design of an interface for friendliness, especially if the interface is used by people with various levels of computer experience. Many psychological issues are involved in the interaction of people with a computer. This report will examine those that are important in the design of an interactive interface for use by both novice and experienced users. The user's experience with computers in general and with a specific interface. Just as hardware issues could not be included in this report because they were beyond the control of the software package, there are certain psychological issues that cannot be included in this report. For example the software package creating the interface cannot have any control of the working environment existing in the office where the interface is used. It cannot provide positive or negative reinforcement for the users other than that displayed upon the CRT.

There are six psychological issues that have an impact on the design of an interactive interface. They are memory of users, level of experience, closure, attitude, anxiety, and control. Each of these subjects will be discussed in the ordered listed. Other areas concerning the psychological design of an interface are included under these six.

4.1 MEMORY

The memory capacity of the user has the greatest impact on the design of a friendly system. Since the interface provides all the information about the software package, the only way the user can be given information about the system is on the CRT. The CRT cannot hold the commands or instructions on the screen while the user goes through the system. Only that information which is held in the user's mind and displayed on the CRT will be available for determining the next instruction. Therefore, the interface has to be designed so that it does not require more memory capacity than the user can apply, or it must have a way to page back through the commands like a user's manual.

The first source of information is in sensory information storage. Sensory information storage has a very short span of retention and is constantly being bombarded with input from the six senses. These raw sensations are held only for the time necessary for short-term memory to examine them and either hold or ignore them. Short-term memory processes the information in the sensory information storage. Short-term memory can retain information for a longer period of time but is limited in the number of items that can be retained. An item of information can be retained for a longer period of time by "rehearsing" (Lachman [13]) it. Rehearsing is the process of repeating the piece of information over and over again. An example is repeating someone's name after being introduced to them the first time.

Long-term memory is where the user's permanent information is kept and can be retrieved and used by the short-term memory. The process by which the human mind places information into permanent storage is not well understood. However, it does appear that the placement of information into long-term memory requires time and effort. The actual organization and continuing reorganization of information in long-term memory is even less understood (Sheiderman [35]). The method for moving information into long-term memory appears to be a process aimilar to that of moving information from the sensory information memory to short-term memory - rehearsal or repetition.

The rehearsing or repetition of information for retention is important in designing an interface. George Miller's 1956 paper, "The Magic Number Seven - Plus or Minus Two", indicates that the short-term memory of a person can only handle seven "chunks" or units of information. Each of these chunks can become larger as the person gains more and more experience with the information. Short-term memory can transfer these chunks to long-term memeory and then retrieve them, combine several together, and transfer them back to long-term memory over and over again. Short-term memory, with its capacity to handle only seven chunks or units of information at one time is what Lachman [13] terms the "bottleneck" in the human information processing system. This bottleneck limits learning by the user. Short-term memory cannot be overloaded and this is a serious limitation in designing the user-friendly interface. The user can only handle approximately seven units of information at one time;

but time and repetition can compress several commands into one chunk. Information from both sensory and long-term storage can be retrieved by short-term memory. Short-term memory with its ability to work with information from both of these memories is sometimes referred to as working memory. With the information from these two sources, short-term memory forms a conceptual model of the interface while learning it by creating a single chunk from several other chunks. These compressed chunks represent a conceptual or semantic meaning to the user. It is only through short-term memory that the user learns to use the system. Moran [20] found that a conceptual model was formed by users of his Command Language Grammar (CLG). He also found that the conceptual model formed by the users was not in all cases the model that the designer had in mind. This problem, the user forming a different model than that desired by the designer, was also noted by Gaines [7].

Another limitation stemming from the limited capability of short-term memory is that people tend to get confused using an interface. While studying users of ZOG, a menu driven network selection system, Robertsen [30] found that people failed to read all of the menu when displayed on the CRT because they could not hold all of the information presented on the menu in short-term memory. Since the information for moving through the system was lost, the user soon became lost within the system.

4.2 LEVEL OF EXPERIENCE

The level of experience of the user is based upon the user's prior training or familiarity with computer systems in general, and the specific interface being examined. Undoubtly the novice needs to have available more detailed instructions than the experienced user. However, even the experienced user, in some respects, acts just like an inexperienced user when exposed to a new interface. Ledgard [16] found support for this issue when he was looking at natural languages for interfaces. This happens because users move back and forth between the novice and experienced levels as they learn the old system and start on another. For this reason users experienced with computers will not be disturbed by learning on an interface designed for the novice. The experienced user will, however, get frustrated quickly if it is necessary to follow the novice's repetitious dialogue, especially at low terminal speed. Tagg [39] pointed this out in his work. Conversely the novice cannot be expected to learn efficiently on an experienced user's interface. The novice must be given the opportunity start at a simpler level.

Anything that can be done to reduce the amount of new information required to operate the system will aid the user. Using a natural language in the command structure for the interface automatically gives the novice an advantage for there is no need to learn new words or symbols. However, long commands in a natural language slows down the experienced user unless there is a way to shorten the

commands such as only using the first letter of a command. In planning any design, the needs of both the inexperienced and experienced user must be taken into account or the interface may not serve its function.

4.3 CLOSURE

Closure is the relief a person feels when information in short-term memory no longer needs to be retained. Closure creates a strong drive to complete a task, free short-term memory, and gain relief. Every time you sign off from a computer or finish typing in a command, closure is experienced. Novice computer users will experience a greater feeling of relief than an experienced user in entering the same command. The experienced user will be able to use larger and longer commands because as experience is gained, more powerful commands are needed to produce the same amount of closure.

The pressure of closure means that users, especially the novice, may prefer several small commands rather than one large one (Shneiderman [35]). The size of the command or entry will, of course, vary with the feelings or preceptions of the user. The desire of the novice to use several small commands is twofold. First it allows frequent checks at each step of the operation to gain assurance that all is going well. Second, it permits the user to forget about the earlier portions of the operation, thus alleviating short-term memory requirements.

4.4 ATTITUDE AND ANXIETY

Attitude and anxiety are closely related and affect each other. Attitude is the state of a person's willingness to learn. Anxiety is a feeling people have when placed under stress. As outlined by Lindsay [15] stress has three causes:

a. Internal models are inadequate to explain the present situation.

b. Internal models lead to an undesirable result that a person feels powerless to prevent.

c. Stress itself.

People's attitude toward a computer is partly explained by the fact that they want to avoid situations that might produce anxiety or undesirable outcomes. If the user has a pleasant experience with a computer, then there will be a favorable attitude toward the computer. If there is an unpleasant experience, a negative attitude will result. Reducing the users' anxieties will improve their ability to use the computer. Smith [36] stated that one reason the computer created anxiety was that its workings were invisible. He further stated that a person just introduced to a computer exhibits emotions based on fear, awe, and general uncertainty.

Users with negative attitudes toward computers make more errors and learn more slowly than those with positive or neutral attitudes. Lucas [19] found that the use of his interactive information storage and retrieval system for medical research was significantly associated with favorable

"user attitudes". Individuals who rated the system highly tended to use it most frequently. Overcoming a negative attitude is partly the job of a user-friendly interface. The design of the responses can improve the attitude of the user toward the computer. A negative attitude displayed by the user is based upon the anxiety generated from working with the computer. To overcome anxiety and any attitude problems, every effort must be made to make the user feel at ease when confronting the terminal and using the interface. Smith [36] stated that communicating with a computer is typically a very "unsatisfactory experience...with rather clumsy devices". Communicating with a computer is, according to Hayes [10] a "time consuming and frustrating experience". On the other hand, the user will not be appreciative if the interface is patronizing.

The user will feel more comfortable with the interface if the instructions and messages to and from the machine are written in clear, concise English and are easy to understand. Simple tasks should be the norm until the user gains confidence from their successful application. Diagnostic messages should be understandable, non-threatening, and low-key. Constructive messages and positive reinforcement produce faster learning and increase user acceptance more than short, terse, cryptic messages that force the user to go to a thick manual for meaning. Robertsen [30] found that by allowing the user to remain at the terminal was good in that it eliminated the jump from barely acquired knowledge to actual use and any search for relevant information by having it available on the

terminal.

4.5 CONTROL

Control is a psychological issue in the design of a user-friendly interface because too often designers have attempted to design the interface so that the computer is an actual entity or intelligent being. Control is the sense of being in charge or command of an object such as a tool. This tool responses to the wishes of the user and not the other way around. Control of the machine cannot be taken from the person using the interface. The interface demanding answers or commands from the user may satisfy the novice user. Once the user becomes familiar with the interface, the user will want to control it and not vice versa. In Dzida's [6] survey of user perceived qualities of a friendly interface, control was one of the factors repeatedly mentioned.

As users gain knowledge and maturity, they begin to resent any attempt on the part of the computer to demand commands from them. Experienced users regard the computer as a tool. They resent messages that even suggest the idea that the computer is demanding responses from them. Control also means that there must be no doubt in the user's mind as to who made the error. The computer is just a tool which reacts to the user's commands. Both of these points were brought up by Gaines [7] in his discussion of his interactive dialogue programming rules. He also stresses the point that users want to dominate the computer. Every

activity of the system must be a clear consequence of the user's actions. Popularity of micro computers is partly due to the fact that the user/owner feels in control. He can see, feel, and handle the file kept on a floppy disk totally unlike a file on a mainframe.

4.6 ERROR HANDLING

Creating the most concern with the users, especially the novice, is error handling. Error messages in computer systems have always consisted of short cryptic messages that can be totally confusing. Ling [18] suggests that messages have little value in an interactive system if the diagnostic message contains little chance for easy recovery or change of tasks. The interface must be tolerant of the user no matter what level of experience. One example of this is the novice who enters the system with great reluctance, and as the next command is being entered the system signs off with a terse, abrupt message stating that it had waited long enough for the command. It is doubtful that this person will ever attempt to use the computer again.

Error handling can also bring about negative reinforcement. Sounds or noises associated with a error message have always proved to be ineffective because the sound draws the attention of others to the user's error. No one likes to have attention drawn to their errors. Any error message that forces the user to leave the system to find the meaning of an error message will decrease its effectiveness. Doherty [4] states that the user's time is

valuable and the computer should be managed so that the user's ability to work is enhanced with the "least amount of inconvenience."

Error handling impacts on all the other psychological issues discussed earlier. The anxiety and attitude of the user is greatly affected by error handling routines. The naturalness of the error message impacts on the ease with which the message is understood. In describing SITAR, which is an interactive text processing system for small computers, Schneider [32] stated that it tried to prompt users and lead them to successful completion of their session time rather than just tell them that an error had occurred.

5.0 DESIGN ISSUES

Design of an interactive interfaces will, in the future, incorporate all aspects of computer programming, hardware capabilities, access control, data base design, and user capabilities. Rayner [25] states that data bases of the future will have to consider open systems with a wide variety of users. The design issues covered in this section can be used by a software package without considering any hardware restrictions or capabilities.

The most important thing in designing any user-friendly interface is that there be a deep-rooted concern by the designer for the convenience of the user. The degree of friendliness of the interface must be defined early in the design stage. A philosophy for designing business programming languages as given by Zloof [41] is that the user should be required to know very little about the system to get started. Lucas [19] supported this philosophy with the comment that "too often technical issues become control focus" while the reactions of the users are ignored.

Compounding the problem of designing what the system should give the user in the way of friendliness is the differences in levels of experience of the users. In other words, the designer must either know the level of experience of each group of users and build specific interfaces for each, or design an interface that will meet the needs of many users. Unfortunately, in the past, the design of the interface served only one level of users mainly because of time and money constraints. One other reason for the poor

design of user-friendly interfaces is due to the fact that the computer-man dialogue is poorly understood; as pointed out by Bernard [1].

A well designed interactive interface can relieve the user of the administrative details of learning the interface. Some of the administrative work could even be incorporated in a programming language as proposed by Negus [21] in his DIALOG language. The user interface that can save a person time will be extremely valuable in the future and will expand the use of computers.

One of the main problems in designing a user-friendly interface is that there are few hard and fast principles. Everyone has a specific example of what the interface should not do, but very few have examples of what it should do. What is friendly for one person might be extremely hostile for the next. Unfortunately data for the design of a user-friendly interface requires observations of people which are not easily quantifiable. The design of a user-friendly interface is still very much an art. It will take design and testing to find out the "best design". In several articles the authors (Moran [20] and Barnard [1]) have stated that designing a user interface is still a "purely intuitive endeavor" and not something that can be put into laws. In addition sometimes the needs of the users change through the design. Several studies have been done by Reisner [29], a psychologist, which explain how several psychological principles can be applied when designing a friendly-user interface.

Design issues are broken down into three areas: User