THE 'KILLER ROBOT' INTERFACE

Dr. Horace Gritty
Department of Computer Science and Related Concerns
Silicon Valley University, Silicon Valley, USA

Abstract: The Robbie CX30 industrial robot was supposed to set a new standard for industrial robot intelligence. Unfortunately, one of the first Robbie CX30 robots killed an assembly line worker, leading to the indictment of one of the robot's software developers, Randy Samuels. This paper propounds the theory that it was the operator-robot interface designer who should be on trial in this case. The Robbie CX30 robot violates nearly every rule of interface design. This paper focuses on how the Robbie CX30 interface violated every one of Shneiderman's "Eight Golden Rules".

1. Introduction

On May 17, 1992 a Silicon Techtronics Robbie CX30 industrial robot killed its operator, Bart Matthews, at Cybernetics, Inc., in Silicon Heights, a suburb of Silicon Valley. An investigation into the cause of the accident led authorities to the conclusion that a software module, written and developed by Randy Samuels, a Silicon Techtronics programmer, was responsible for the erratic and violent robot behavior which in turn lead to the death by decapitation of Bart Matthews [FOOTNOTE: The media were misled to believe that Bart Matthews was crushed by the robot, but the photographic evidence given to this author shows otherwise. Perhaps authorities were attempting to protect public sensibilities.].

As an expert in the area of user interfaces (1,2,3), I was asked to help police reconstruct the accident. In order to accomplish this, Silicon Techtronics was asked to provide me with a Robbie CX30 simulator which included the complete robot operator console. This allowed me to investigate the robot's behavior without actually risking serious harm. Due to my extensive understanding of user interfaces and human factors I was able to reconstruct the accident with uncanny accuracy. On the basis of this reconstruction, I came to the conclusion that it was the interface design and not the admittedly flawed software which should be viewed as the culprit in this case.

Despite my finding, Prosecuting Attorney Jane McMurdock insisted on pursuing the case against Randy Samuels. I believe that any competent Computer Scientist, given an opportunity to interact with the Robbie CX30 simulator, would also conclude that the interface designer and not the programmer should be charged with negligence, if not manslaughter.

2. Shneiderman's 'Eight Golden Rules'

My evaluation of the Robbie CX30 user interface is based upon Shneiderman's 'eight golden rules' (4). I also used other techniques to evaluate the interface, but those will be published in separate papers. In this section, I offer a brief review of Shneiderman's eight golden rules, a subject which would be more familiar to computer interface experts such as myself as opposed to the robot hackers who read this obscure journal.

The eight golden rules are:

1. ***Strive for consistency***. As we shall see below, it is important for a user interface to be consistent on many levels. For example, screen layouts should be consistent from one screen to another. In an environment using a graphical user interface (GUI), this also implies consistency from one application to another.

2. ***Enable frequent users to use shortcuts***. Frequent users (or, power users) may be turned off by overly tedious procedures. Allow those users a less tedious procedure for accomplishing a given task.

3. ***Offer informative feedback***. Users need to see the consequences of their actions. If a user enters a command but the computer does not show that it is either processing or has processed that command, this can leave the user confused and disoriented.

4. ***Design dialogues to yield closure***. Interacting with a computer is somewhat like a dialogue or conversation. Every task should have a beginning, a middle and an end. It is important for the user to know when a task is at its end. The user needs to have the feeling that a task has reached closure.

5. ***Offer simple error handling***. User errors should be designed into the system. Another way of stating this is that no user action should be considered an error that is beyond the ability of the system to manage. If the user makes a mistake, the user should receive useful, concise and clear information about the nature of the mistake. It should be easy for the user to undo his or her mistake.

6. ***Permit easy reversal of actions***. More generally, users must be permitted to undo what they have done, whether it is in the nature of an error or not.

7. ***Support internal locus of control***. User satisfaction is high when the user feels that he or she is in control and user satisfaction is low when the user feels that the computer is in control. Design interfaces to reinforce the feeling that the user is the locus of control in the human-computer interaction.

8. ***Reduce short-term memory load***. Human short-term memory is remarkably limited. Psychologists often quote Miller's law to the effect that short-term memory is limited to seven discrete pieces of information. Do everything possible to free the user's memory burden. For example, instead of asking the user to type in the name of a file which is going to be retrieved, present the user with a list of files currently available.

3. Robot console overview

The Robbie CX30 operator interface violated each and every one of Shneiderman's rules. Several of these violations were directly responsible for the accident which ended in the death of the robot operator.

The robot console was an IBM PS/2 model 55SX with a 80386 processor and an EGA color monitor with 640x480 resolution. The console had a keyboard, but no mouse. The console was embedded in a workstation which included shelves for manuals and an area for taking notes and for reading manuals. However, the reading/writing area was quite a distance from the computer screen, so that it was quite awkward and tiresome for the operator to manage any task which required looking something up in the manual and then acting quickly with respect to the console keyboard. The operator's chair was poorly designed and much too high relative to the console and the writing/reading area. This placed much strain on the operator's back and also caused excessive eye strain.

I cannot understand why a sophisticated system such as this would not include a better device for input. One can only conclude that Silicon Techtronics did not have much experience with user interface technology. The requirements document (5) specified a menu-driven system, which was a reasonable choice. However, in an application where speed was of the essence, especially when operator safety was at issue, the use of a keyboard for all menu selection tasks was an extremely poor choice, requiring many keystrokes to achieve the same effect which could be achieved almost instantaneously with a mouse. (See the paper by Foley et al. (6). Actually, I had most of these ideas before Foley published them, but he beat me to the punch.)

The robot operator could interact with the robot and thus impact upon its behavior by making choices in a menu system. The main menu consisted of twenty items, too many in my opinion, and each main menu item had a pull-down submenu associated with it. Some of the submenus contained as many as twenty items - again, too many. Furthermore, there seemed to be little rhyme or reason as to why the menu items were listed in the order in which they were listed. A functional or alphabetical organization would have been better.

Some items in the pull-down submenus had up to four pop-up menus associated with them. These would appear in sequence as submenu choices were made. Occasionally, a submenu choice would cause a dialogue box to appear at the screen. A dialogue box requires some kind of interaction between the operator and the system to resolve some issue, such as the diameter of the widgets being lowered into the acid bath.

The menu system presents a strict hierarchy of menu choices. The operator could backtrack up the hierarchy by pressing the escape key. The escape key could also terminate any dialogue. The use of color in the interface was very unprofessional. There were too many colors in too small a space. The contrasts were glaring and the result, for this reviewer, was severe eye strain in just fifteen minutes. There was excessive use of flashing and silly musical effects when erroneous choices or erroneous inputs were made.

One has to wonder why Silicon Techtronics did not attempt a more sophisticated approach to the interface design. After a careful study of the Robbie CX30 applications domain, I have come to the conclusion that a direct manipulation interface, which literally displayed the robot at the operator console, would have been ideal. The very visual domain that the robot operated within would lend itself naturally to the design of appropriate screen metaphors for that environment, metaphors which the operator could easily understand. This would allow the operator to manipulate the robot by manipulating the graphical representation of the robot in its environment at the computer console. I have asked one of my doctoral students, Susan Farnsworth, to give up her personal life for the better part of a decade in order to investigate this possibility a bit further.

4. How the Robbie CX30 interface violated the eight golden rules

The Robbie CX30 user interface violated each and every golden rule in multitudinous ways. I shall only discuss a few instances of rule violation in this paper, leaving a more detailed discussion of these violations for future articles and my forthcoming book [FOOTNOTE: CODEPENDENCY: How Computer Users Enable Poor User Interfaces, Angst Press, New York. This book presents a radically new theory concerning the relationship between people and their machines. Essentially, some people need a poor interface in order to avoid some unresolved psychological problems in their lives.]. I will emphasise those violations which were relevant to this particular accident.

4.1 Strive for consistency

There were many violations of consistency in the Robbie CX30 user interface. Error messages could appear in almost any color and could be accompanied by almost any kind of musical effect. Error messages could appear almost anywhere at the screen.

When Bart Matthews saw the error message for the exceptional condition which occurred, an exceptional condition which required operator intervention, it was probably the first time he saw that particular message. In addition, the error message appeared in a green box, without any audio effects. This is the only error message in the entire system which appears in green and without some kind of orchestral accompaniment.

4.2 Enable frequent users to use shortcuts

This principle does not appear in any way in the entire interface design. For example, it would have been a good idea to allow frequent users to enter the first letter of a submenu or menu choice in lieu of requiring the use of the cursor keys and the enter key to effect a menu choice. The menu selection mechanism in this system must have been quite a mental strain on the operator.

Furthermore, a form of type-ahead should have been supported, which would have allowed a frequent user to enter a sequence of menu choices without having to wait for the actual menus to appear.

4.3 Offer informative feedback

In many cases, the user has no idea whether a command that was entered is being processed. This problem is exaggerated by inconsistencies in the user interface design. In some cases the operator is given detailed feedback concerning what the robot is doing. In other cases the system is mysteriously silent. In general, the user is led to expect feedback and consequently becomes confused when no feedback is given. There is no visual representation of the robot and its environment at the screen and the operator's view of the robot is sometimes obstructed.

4.4 Design dialogues to yield closure

There are many cases in which a given sequence of keystrokes represents one holistic idea, one complete task, but the operator is left without the kind of feedback which would confirm that the task has been completed. For example, there is a fairly complicated dialogue which is necessary in order to remove a widget from the acid bath. However, upon completion of this dialogue, the user is led into a new, unrelated dialogue, without being informed that the widget removal dialogue has been completed.

4.5 Offer simple error handling

The system seems to be designed to make the user regret any erroneous input. Not only does the system allow numerous opportunities for error, but when an error actually occurs, it is something that is not likely to be repeated for some time. This is because the user interface makes recovery from an error a tedious, frustrating and at times infuriating ordeal. Some of the error messages were downright offensive and condescending.

4.6 Permit easy reversal of actions

As mentioned in the previous paragraph, the user interface makes it very difficult to recover from erroneous inputs. In general, the menu system does allow easy reversal of actions, but this philosophy is not carried through to the design of dialogue boxes and to the handling of exceptional conditions. From a practical (as opposed to theoretical) point of view, most actions are irreversible when the system is in an exceptional state, and this helped lead to the killer robot tragedy.

4.7 Support internal locus of control

Many of the deficiencies discussed in the previous paragraphs diminished the feeling of "internal locus of control". For example, not receiving feedback, not bringing interactions to closure, not allowing easy reversal of actions when exceptions arose, all of these things act to diminish the user's feeling of being in control of the robot. There were many features of this interface which make the operator feel that there is an enormous gap between the operator console and the robot itself, whereas a good interface design would have made the user interface transparent and would have given the robot operator a feeling of being in direct contact with the robot. In one case, I commanded the robot to move a widget from the acid bath to the drying chamber and it took 20 seconds before the robot seemed to respond. Thus, I did not feel like I was controlling the robot. The robot's delayed response along with the lack of informative feedback at the computer screen made me feel that the robot was an autonomous agent - an unsettling feeling to say the least.

4.8 Reduce short-term memory load

A menu driven system is generally good in terms of the memory burden it places upon users. However, there is a great variation among particular implementations of menu systems insofar as memory burden is concerned. The Robbie CX30 user interface had very large menus without any obvious internal organization. These place a great burden upon the operator in terms of memory and also in terms of scan time, the time it takes the operator to locate a particular menu choice.

Many dialogue boxes required the user to enter part numbers, file names, and other information from the keyboard. The system could easily have been designed to present the user with these part numbers and so forth without requiring the user to recall these things from his or her own memory. This greatly increased to memory burden upon the user.

Finally, and this is really unforgivable, incredible as it may seem, there was no on-line, context-sensitive help facility! Although I was taken through the training course offered by Silicon Techtronics, I often found myself leafing through the reference manuals in order to find the answer to even the most basic questions, such as: "What does this menu choice mean? What will happen if I make this choice?"

5. A reconstruction of the 'killer robot' tragedy

Police photographs of the accident scene are not a pleasant sight. The operator console was splattered with a considerable amount of blood. However, the photographs are of exceptional quality and using blow-up techniques, I was able to ascertain the following important facts about the moment when Bart Matthews was decapitated:

1. ***The NUM LOCK light was on.***

The IBM keyboard contains a calculator pad which can operate in two modes. When the NUM LOCK light is on, it behaves like a calculator. Otherwise, the keys can be used to move the cursor at the screen.

2. ***Blood was smeared on the calculator pad.***

Bloody fingerprints indicate that Bart Matthews was using the calculator pad when he was struck and killed.

3. ***A green error message was flashing.***

This tells us the error situation in force when the tragedy occurred. The error message said, "ROBOT DYNAMICS INTEGRITY ERROR - 45 ".

4. ***A reference manual was open and was laid flat in the workstation reading/writing area.***

One volume of the four volume reference manual was open to the index page which contained the entry 'ERRORS / MESSAGES'.

5. ***A message giving operator instructions was also showing on the screen.***

This message was displayed in yellow at the bottom of the screen. This message read "PLEASE ENTER DYNAMICAL ERROR ROBOT ABORT COMMAND SEQUENCE PROMPTLY!!!"

On the basis of this physical evidence, plus other evidence contained in the system log, and based upon the nature of the error which occurred (robot dynamics integrity error - 45, the error which was caused by Randy Samuels' program), I have concluded that the following sequence of events occurred on the fateful morning of the killer robot tragedy:

10:22.30. "ROBOT DYNAMICS INTEGRITY ERROR - 45" appears on the screen. Bart Matthews does not notice this because there is no beep or audio effect such as occurs with every other error situation. Also, the error message appears in green, which in all other contexts means that some process is proceeding normally.

10:24.00. Robot enters state violent enough for Bart Matthews to notice.

10:24.05. Bart Matthews notices error message, does not know what it means. Does not know what to do. He tries "emergency abort" submenu, a general purpose submenu for turning off the robot. This involves SIX separate menu choices, but Mr. Matthews does not notice that the NUM LOCK light is lit. Thus, the menu choices aren't registering because the cursor keys are operating as calculator keys.

10:24.45. Robot turns from acid bath and begins sweep towards operator console, its jagged robot arms flailing wildly. No one anticipated that the operator might have to flee a runaway robot, so Bart Matthews is cornered in his work area by the advancing robot. At about this time, Bart Matthews retrieves the reference manual and starts looking for a reference to ROBOT DYNAMICS INTEGRITY ERROR - 45 in the index. He successfully locates a reference to error messages in the index.

10:25.00. Robot enters the operator area. Bart Matthews gives up on trying to find the operator procedure for the robot dynamics integrity error. Instead, he tries once again to enter the "emergency abort" sequence from the calculator keypad, when he is struck.

6. Summary and conclusions

While the software module written by Randy Samuels did cause the Robbie CX30 robot to oscillate out of control and attack its human operator, a good interface design would have allowed the operator to terminate the erratic robot behaviour. Based upon an analysis of the robot user interface using Shneiderman's eight golden rules, this interface design expert has come to the conclusion that the interface designer and not the programmer was the more guilty party in this unfortunate fiasco.

References

1. Gritty, Horace (1990). The Only User Interface Book You'll Ever Need. Vanity Press, Oshkosh, WI, 212 pp.

2. Gritty, Horace (1992). "What We Can Learn from the Killer Robot", invited talk given at the Silicon Valley University International Symposium on Robot Safety and User Interfaces, March 1991. Also to appear in Silicon Valley University Alumni Notes.

3. Gritty, Horace (expected 1993). CODEPENDENCY: How Computer Users Enable Poor User Interfaces, Angst Press, New York.

4. Shneiderman, Ben (1987). Designing the User Interface, Addison-Wesley, Reading MA, 448 pp.

5. Robbie CX30 INTELLIGENT INDUSTRIAL ROBOT REQUIREMENTS DOCUMENT : Cybernetics Inc. Version, Technical Document Number 91-0023XA, Silicon Techntronics Corporation, Silicon Valley, USA, 1245 pp.

6. Foley, J. P., Wallace, V. L., and Chan, P. (1984). "The Human Factors of Computer Graphics Interaction Techniques". IEEE COMPUTER GRAPHICS AND APPLICATIONS, 4(11), pp. 13-48.

