10 Graphical User Interfaces

10.1 Introduction

Have you ever had difficulty in finding your way round a website, using the latest piece of software, or operating your mobile telephone? Why is so much computer-based technology difficult to use?

A vast array of computer-based systems surrounds us: stand-alone and networked personal computer systems, websites, safety-critical systems and embedded systems such as mobile telephones and microwave ovens. They influence every aspect of our lives, both at work and in our leisure time. Their effectiveness depends primarily on the user interface (UI), which allows the user to interact with the computer system built into them. Poor UI design can have serious consequences. How many of us would want to fly in an aeroplane whose pilot was unsure what all the knobs and dials did? For organisations, poorly designed UIs can increase training costs and reduce productivity. For the user, they can cause frustration and inefficiency. How often have you moved to another website because you cannot find the information you need?

10.2 General User Interface Design

User Interface Design is often associated with software interface and is frequently referred to as Human-Computer Interface or HCI. However, User Interface Design must be considered wherever users interact with controls or displays. The application of interface design is commonly found, these include products such as: a simple watch, a DVD player, an aircraft cockpit, a software program, etc. In many cases, good technology is not readily accepted because the product is not easy or efficient to use. A product's usability, acceptance, and marketability are often dependent on the user feeling that it is easy to learn and use. User Interface Design increases the intuitiveness, efficiency, and comfort level with a product, which translates into product acceptance and use. You need both good technology and usability for a successful product. Many technological innovations rely upon User Interface Design to elevate their technical complexity to a usable product. Technology alone may not win user acceptance and subsequent marketability. The User Experience, or how the user experiences the end product, is the key to acceptance. And that is where User Interface Design enters the design process.
When applied to computer software, User Interface Design is also known as Human-Computer Interaction or HCI. While people often think of Interface Design in terms of computers, it also refers to many products where the user interacts with controls or displays. Military aircraft, vehicles, airports, audio equipment, and computer peripherals, are a few products that extensively apply User Interface Design.

The importance of good User Interface Design can be the difference between product acceptance and rejection in the marketplace. If end-users feel it is not easy to learn, not easy to use, or too cumbersome, an otherwise excellent product could fail. Good User Interface Design can make a product easy to understand and use, which results in greater user acceptance.

10.2.1 User-Interface in Embedded Systems

Embedded systems involve close interaction with real-world physical components and mechanisms. This interaction takes the form of control, coordination, or monitoring of actual electronic or electromechanical devices. What stands out is both the importance of real-world objects and the sequence of events or transactions taking place between the embedded software, external physical objects, and their users.

An embedded system for a given application must ultimately exhibit the same effective behaviour and external characteristics regardless of how the internal program is realized. These features and capabilities can be embodied into which their performance must conform to requirements.

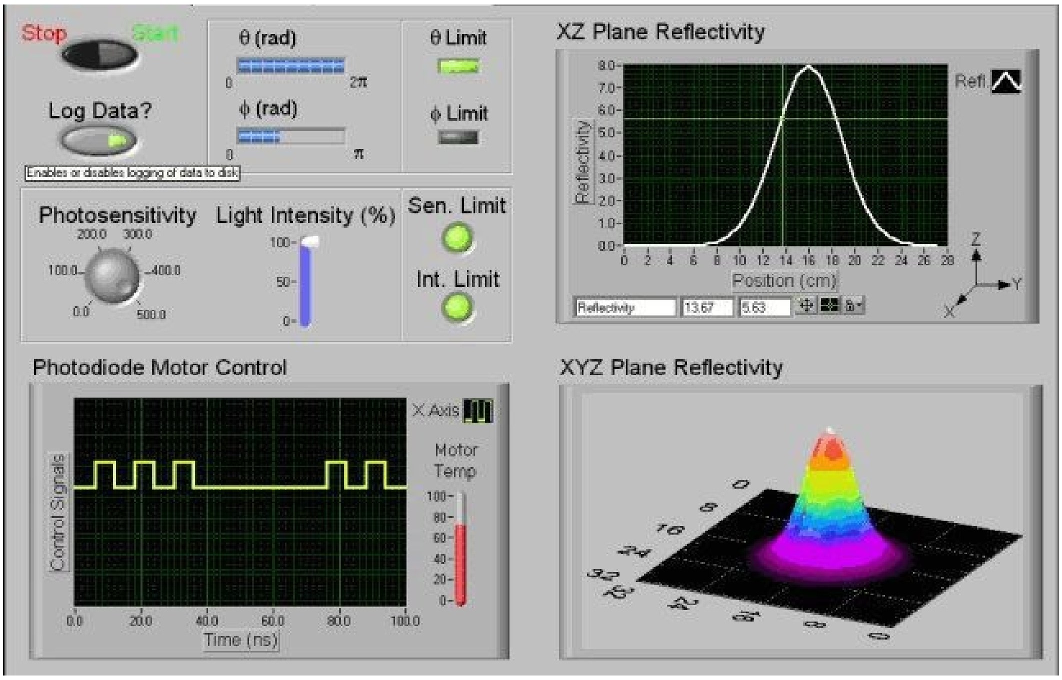


Fig 10.1 An example for the screen display of User Interface Design

10.3 Principles to consider for effective User Interface Design

This section describes some general factors which determine an effective user interface design. By this we mean how well the product interface helps the user accomplish a task. The ease of use of the interface plays a great part in the productivity of the user. After all the product is meant to help, not hinder the user! Principles should always be in the mind of the designer as they work on an interface. Guidelines will help the designer realise a principle.

10.3.1 Consistency

The importance of maintaining strict consistency varies according to the level of the task required. It is useful to be aware of when to apply them. Now, it is just important to be visually inconsistent when things must act differently as it is to be visually consistent when things act the same. We should avoid uniformity just for its own sake. Make objects consistent with their behavior. Make objects that act differently look different from the rest. The most important consistency is consistency with user expectations. The only way to find out user expectations is to do user testing, which no amount of study and debate will substitute.

10.3.2 User Efficiency

Look at the user's productivity, not the computer's. People cost a lot more money than machines, and while it might appear that increasing machine productivity must result in increasing human productivity, the opposite is often true. In judging the efficiency of a system, look beyond just the efficiency of the machine. As an example, which of the following takes less time? Heating water in a microwave for one minute ten seconds or one minute eleven seconds? Of course one minute ten seconds is obviously faster. But the user must press the one key twice, then visually locate the zero key, move the finger into place over it, and press it once. In the second case, the user just presses the same key–the one key–three times. It typically takes more than one second to locate the zero key.

Other factors beyond speed make the “111" solution more efficient. Seeking out a different key not only takes time, it requires a fairly high level of thinking and care. While the processing is underway, the main task the user was involved with–cooking their meal–must be set aside. The longer it is set aside, the longer it will take to reacquire it. In addition, the user who adopts the expedient of using repeating digits for microwave cooking faces fewer decisions.

A good example of this is Fitts' Law which says that “The time to acquire a target is a function of the distance to and size of the target”. In other words, an often used object or action should be large and easily accessible - by menu or mouse.

10.3.3 Using Human Interface Objects

Human Interface Objects are representations of objects that humans can relate to.

i) They can be seen, heard, touched, or otherwise perceived.
ii) Visual interface objects are quite familiar in graphic user interfaces, although those using other human senses are available.

iii) Since they relate to standard objects, they must have a standard way of interacting with it.

iv) They must also have standard resulting behaviours and not produce surprises.

Some familiar objects include folders, documents, and the trashcan. Thus, icons are typically used to represent them, pictorially.

Metaphors

As an extension to this concept, metaphors are representations of system objects that correspond to the familiar real world objects, so that handling them would not require extra training, as they operate like actual objects. For example, Windows has an object called a briefcase. Like a real-world briefcase, its purpose is to help make electronic documents more portable. Choosing metaphors well, will enable users to instantly grasp the finest details of the conceptual model. Bring metaphors alive by appealing to people's perceptions–sight, sound and touch as well as triggering their memories.

10.4 Ergonomic Guidelines for User-interface Design

Ergonomics is one major factor that made a big difference for an effective GUI design. The design concept will always have the end user in mind. Good ergonomics makes every effort to ensure a user friendly and comfortable user environment.

The following points are guidelines to good software interface design, which is to help implement the principles discussed. These guidelines apply to the content of screens. In addition to following these guidelines, effective software also necessitates using techniques, such as 'storyboarding', to ensure that the flow of information from screen to screen is logical, follows user expectations, and follows task requirements. These are presented in point form.

10.4.1 Consistency

- certain aspects of an interface should behave in consistent ways at all times for all screens
- terminology should be consistent between screens
- icons should be consistent between screens colours should be consistent between screens of similar function

10.4.2 Simplicity

- break complex tasks into simpler tasks
- break long sequences into separate steps
- keep tasks easy by using icons, words etc.
- use icons/objects that are familiar to the user

 - delay (processing)	 - save	 - zoom

10.4.3 Human Memory Limitations

- organize information into a small number of "chunks"
- try to create short linear sequences of tasks
- don't flash important information onto the screen for brief time periods
- provide cues/navigation aids for the user to know where they are in the software or at what stage they are in an operation
- provide reminders, or warnings as appropriate
- provide ongoing feedback on what is and/or just has happened
- let users recognize information by making it available rather than try to remember it.

10.4.4 Cognitive Directness

This term refers to the mental processes that a user needs to use in order to get a device to perform a function. To make this process more "direct" is to reduce the amount of mental work needed a user uses less effort to accomplish a task. The shortest number of steps taken to perform an operation may not be the most intuitive!

- minimize mental transformations of information (e.g. using 'control-I' to indent a paragraph)
- use meaningful icons/letters
- use appropriate visual cues, such as direction arrows for movement of information
- use 'real-world' metaphors whenever possible (e.g. desktop metaphor, folder metaphor, trash can metaphor etc.)

10.4.5 Feedback

It is useful to identify the types of feedback that can be used. This makes us think more carefully about how to use them. Those discussed below are a typical selection of feedback types. This is especially true for the system is processing data.

For time related considerations:

- Acknowledge all button clicks by visual or aural feedback within 50 milliseconds.
- Display some sort of "waiting" indicator (e.g. hourglass) for any action that will take from 1/2 to 2 seconds.
- Animate the indicator so users won't think the system has "hung".
- Display messages indicating the potential length of the wait for any action that will take longer than 2 seconds.
- Communicate the actual length through an animated progress indicator.
- If possible, show text messages to users informed and entertained while they are waiting for long processes to complete.

- Have the system give a large signal upon return from lengthy (>10 seconds) processes, so that users know when to return to using the system.

For other types of feedback,

- give informative feedback at the appropriate points - in the form of messages
- provide appropriate sensory feedback
- confirm the physical operation you just did (e.g. typed ‘help’ and ‘help’ appears on the screen).
- this includes all forms of feedback, such as auditory feedback (e.g. system beeps, mouse click, key clicks etc.)

- provide appropriate semantic feedback – ‘semantic means ‘meaning’ - this type of feedback is meaningful to the action being performed. It is also time consuming to implement as it has to recognize the context of an action. (e.g. when adding up a series of numbers, a running total is shown, or when selecting an object, a set of icons showing possible actions are shown)

- provide appropriate status indicators to show the user the progress with a lengthy operation (e.g. the copy bar when copying files, an hour glass icon when a process is being executed etc.)

10.4.6 System messages

- provide user-centered wording in messages (e.g. “there was a problem in copying the file to your disk” rather than “execution error 159”)
- avoid ambiguous messages (e.g. hit ‘any’ key to continue – there is no ‘any’ key and there’s no need to hit a key, reword to say ‘press the return key to continue)
- avoid using threatening or alarming messages (e.g. fatal error, run aborted, kill job, catastrophic error)
- use specific, constructive words in error messages (e.g. avoid general messages such as ‘invalid entry’ and use specifics such as ‘please enter your name’)
- make the system ‘take the blame’ for errors (e.g. “illegal command” versus “unrecognized command”)

10.4.7 Anthropomorphization

- don't anthropomorphize (i.e. don't attribute human characteristics to objects)
- avoid the "Have a nice day" messages from your computer. It can come across as patronizing rather than cute.

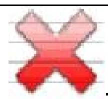
10.4.8 Modality

- a mode is an interface state where what the user does has different actions than in other states (e.g. changing the shape of the cursor can indicate whether the user is in an editing mode or a browsing mode)

- make user actions easily reversible – use ‘undo’ commands, but use these sparingly
- allow escape routes from operations

10.4.9 Attention

- Use attention grabbing techniques cautiously (e.g. avoid overusing ‘blinks’ on webpages, flashing messages, ‘you have mail’, bold colors etc.)
- don’t overuse audio or video
- use colors appropriately and make use of expectations (e.g. don’t have an OK button colored red! Use green for OK, yellow for ‘caution, and red for ‘danger’ or ‘stop’)
- don’t use blue for text (hard to read), blue is a good background color
- don’t put red text on a blue background
- use high contrast color combinations
- use colors consistently

 - correct or select	 - error or close	 - unexpected action

10.4.10 Display issues

- maintain display inertia – make sure the screen changes little from one screen to the next within a functional task situation
- organize screen complexity - eliminate unnecessary information
- use concise, unambiguous wording for instructions and messages
- use easy to recognize icons, be sensitive to cultural interpretation

10.4.11 Individual differences

- accommodate individual differences in user expertise (from the novice to the computer literate)
- accommodate user preferences by allowing some degree of customization of screen layout, appearance, icons etc.

- allow alternative forms for commands (e.g. key combinations through menu selections or mouse click selection

10.5 User Interface Design for Embedded Systems

The previous sections have discussed interface design in general. Embedded systems are part of a product and there is a close level of interaction between the user. In fact, they may be used on a daily basis by untrained users. Also, these systems may be controlling critical systems. Thus we have to recognize that the user interface is key and we must put users at the center of the design and development process.

If the interfaces are bad, complaints about the controls equipment will surface. Users have to refer back to manuals. To some extent, user may use only a fraction of the capabilities of systems.

When it comes to the user interface, embedded systems differ from "standard" applications development that use the entire computer screen as the primary interaction interface. The product containing the embedded system almost completely determines or constrains the appearance of the interface to the user.

Input devices are limited switches and keypads, outputs to LEDs, buzzers and screens of any sort are difficult to come by. However, there may be a great amount of freedom in the placement these devices subject to constraints of a product. For example, given a certain chassis size, there is quite a lot of freedom in where to place switches and LEDs. But it is very difficult to change these once the positions are fixed.

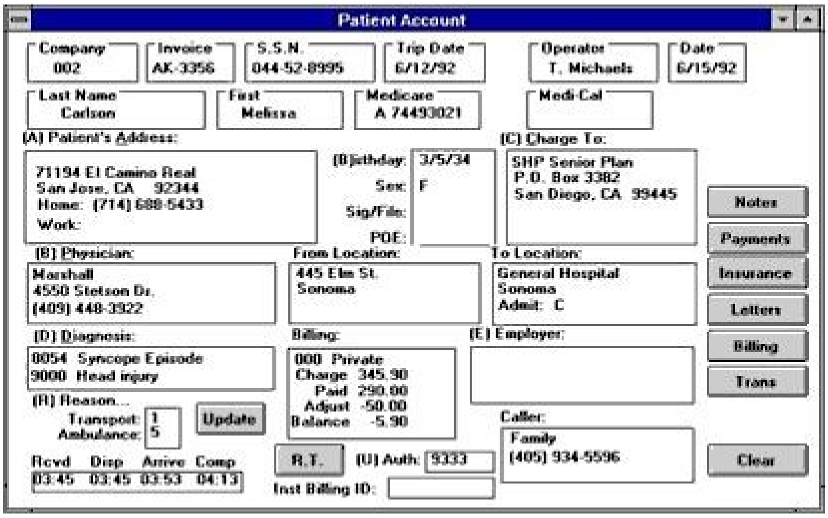
10.5.1 User-Centered Design

This is more than just being "user-oriented"; it makes the end user the focus that informs the entire design process. User-centered design is not about "user friendliness" either, but about making systems that are substantially easier for users to make use of — simply, quickly, and reliably— that make it easier to do things well.

Better user interfaces can be designed if you keep in mind some broad principles about human-machine interaction. Here we will mention a few of the more important ones as they apply to user interfaces for embedded system applications.

Keeping the user informed means that signals or messages to the user must be clear and clearly distinguishable. Sometimes the output devices for embedded systems applications are somewhat simplified. There might be only a set of LEDs or a small LCD panel, but even such simple devices can be used remarkably effectively to communicate a range of information.

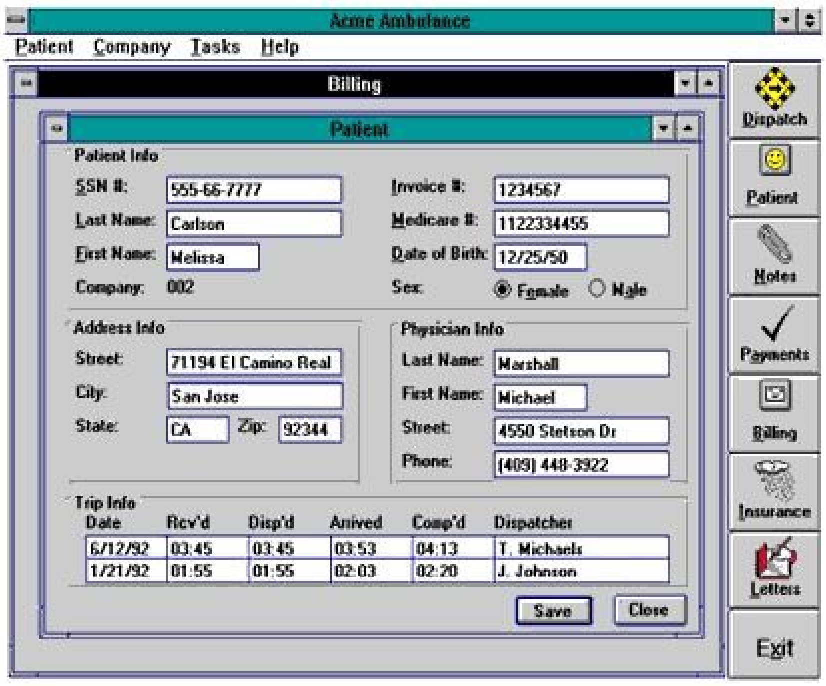
An effective way to get better user interfaces is to take a critical, investigative approach, studying the user interface, studying users, and studying users using the user interface. The following is a GUI design in need of redesign. See if you can identify its purpose. Here is what is wrong with this screen:



In respect to the design guidelines mentioned earlier, some of the problems are:

Consistency - Groupings of data have no pattern, not properly arranged, showing lack of symmetry.
Cognitive directness - Too many buttons, not explained well, no icons. Difficult to know what to do!
Simplicity - there is no indication of the overall tasks and subtasks, don't know where to start.

Here is an improved version of the GUI. It shows a much improved interface for this same application:



By reordering input fields and using icons in the controls, using the interface has become a more pleasant and productive task.

10.5.2 Critical Test

The critical test is to give someone a prototype, or simulation, and tell them what the system is supposed to do. Then let try to use the product without a manual or further instructions. To get the most from these studies, you should video the session, then review the session with the user. Keeping in mind that any problem the user has, however minor, is not a symptom of their ability or inattentiveness, but a problem in the user interface that ought to be examined more closely. Every mistake or misstep they make indicates a probable design flaw.

10.5.3 Usability for GUI in Embedded Devices

Adding a graphical display to your product may allow you to add more features in a smaller space, but it also raises usability issues.

With a non-graphical display, one layout of buttons and displays has to be designed and evaluated. With a graphical user interface (GUI) there is no limit to the number of possible layouts. Making each one user friendly, while remaining consistent with the others, is a big challenge.

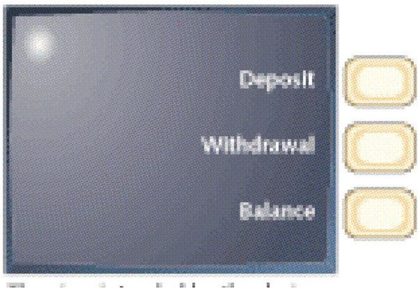


Fig 10.2 Example of GUI for ATM machine

While a GUI has many advantages, it is important to note a couple of the disadvantages.

i) Though a GUI allows a number of different controls on the screen, they all have the same tactile feel when making an input, needing more visual attention as compared to say, a push button as compared to a switch.

ii) Another disadvantage of the GUI is that space does not generally permit the important controls to be permanently visible. A related problem is that if only a GUI is used, it will not be possible to have all of the controls visible at all times. This means that the user may have to explore the interface to find some of the functions.

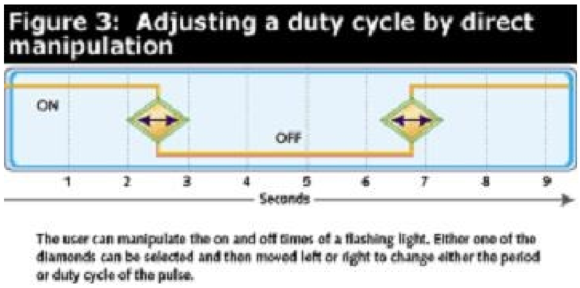
<u>GUI used with hardware</u>

Many embedded products get the best of both worlds by adding a graphics screen to support peripheral information, while the most important user dialog still takes place using custom controls. The next level of GUI is to provide interactions where the input and output is graphical by nature, not just a set of controls that could have been implemented with mechanical switches, dials, and sliders. For example, instead of outputting a numerical value, the value could be graphed over time giving the user a better sense of the changes within the control process.

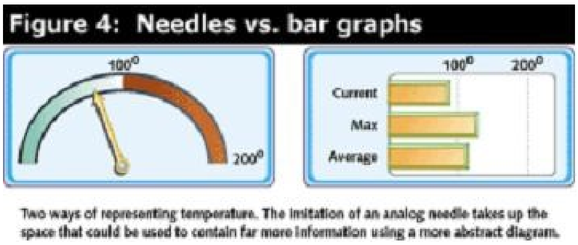
10.5.4 Different types of Interaction in GUI

More precisely, a GUI can be made up of several types of *controls* which the user manipulates, or to display information. In some systems, these controls are also known as *widgets*.

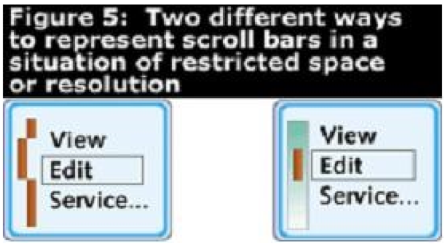
Tick-box Control



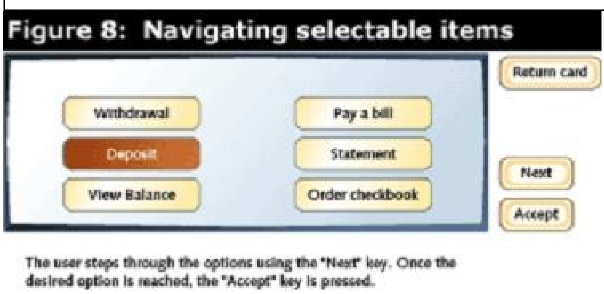
Direct Manipulation



Needles and Bar Graphs



Scroll Bars



Navigational Select Buttons

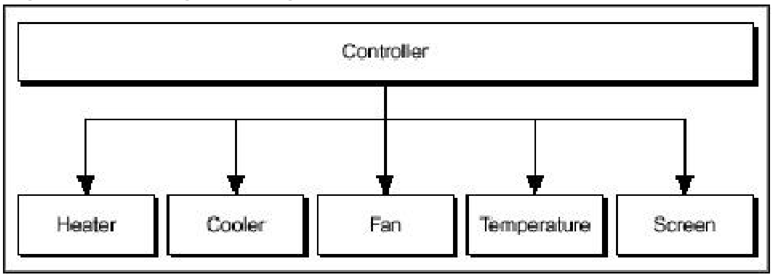
Appendix

CASE STUDY

Description

The embedded system for an air conditioning system. The control system consists of the following subsystems:

Data acquisition board with integrated analog and digital I/O
A 9 in. LCD screen
Heater
Cooler
Fan Temperature Sensor



The heater and cooler are both turned on and off with digital signals. Analog signals are used to inform the heater or the cooler how many degrees the air needs to be heated or cooled, respectively. The fan has five different speeds controlled by digital signals. From the temperature gauge the current temperature is sent by an analog signal.

The relationship between the classes is represented in Universal Modeling Language (UML), a notation system used in object-oriented analysis and design.

Subsystem

The Climate System component is comprised of five classes:

Climate Controller
Fan
Heater
Cooler
Temp Sensor

Design of Graphical User Interface

The picture beside represents the GUI. A knob in the upper left corner controls the desired temperature. On the right side, current system information is displayed.