



AS – Level ICT

IT05 14.06 Human / Computer Interaction

- **Describe the psychological factors that affect human / computer interaction: user friendly, give help to novices, provide short cuts for experts, make use of human long term memory to maximise efficiency.**

AQA Specification

NOTE:

(There are many factors in this module element that need to be considered when designing your forms for your IT06 Major Project)

What is human-computer interaction?

Human-computer interaction is an area concerned with the design, evaluation and implementation of interactive computer systems. Studies of such interaction look at the user interface and at how easy it is for the user to communicate with the computer and vice versa. Improving human-computer interaction is important because it allows the user to work more quickly and hence increases productivity. It can also increase job satisfaction and reduce user stress.

Psychological factors affecting human-computer interaction

Software designers and programmers are constantly developing useful and innovative packages which are commercial failures because they are unusable by those who are inexperienced. They have often been designed from the programmer's perspective and are sometimes simply a means for the programmer to display his or her skills.

Some programs require certain combinations of keystrokes for commands and although the combinations chosen may seem obvious to the programmer, they can be off-putting to a novice user. Sometimes the user may be led down a path in an application only to be left stranded when there is no indication provided as to what to do next. In other applications, when the user presses a key they are not given any indication of whether the key has been accepted or not. User-friendliness does not just apply to the software itself; it can also apply to the manuals that accompany the software. Many software manuals make too many assumptions as to the user's prior computer knowledge. Software should be designed from the point of view of the novice user.

An extract from a book on software development gave the following advice:

Software: You should ask yourself the following question:

'What kind of program can I write so that it will meet the needs of 90 per cent of the users rather than the 10 per cent or fewer experienced users who must have enormously varied features and capabilities'?

Manuals: You should ask yourself the following question:

'How can I inform a total novice about the most essential elements of any program with just a few pages of text'?

There is often a false assumption that the more complex a program appears, the more features it will have and the more useful it will be. For an application package to be commercially successful, it need not have complex graphics, colour options, windows, toolbars and icons. It needs instead to be user-orientated, so that the minimum number of keystrokes or mouse movements are needed to enable the user to achieve their objective. There must be a coherent logic in the interface which is repeated and reinforced as the user employs its various features.

Technophobia

Many people, when introduced to computing, are afraid to try things out. Some people never seem to grasp the operation of devices such as video recorders, washing machines, etc. Many of these people find it difficult to understand manuals and instruction books. The human/computer interface should therefore be designed for accessibility.

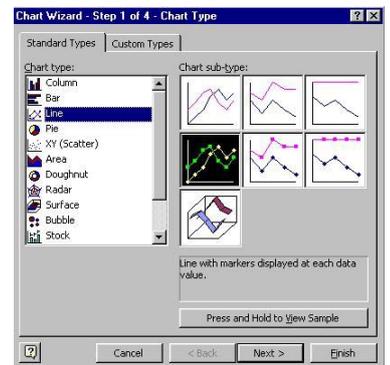
User Freindly

A user-friendly interface is less frustrating and less stressful to use, and the user can therefore work more quickly. Documentation accompanying the program should also be user-friendly. The user should not have to wade through a lot of confusing material to learn about basic features of the software, such as how to start the program or print out the results. One of the best methods of ensuring that software interfaces are user-friendly is to pay end users to test the completed version before release. Testers should be picked to include novice users.

Easy access to help

Some help screens can be very off-putting. They frequently use unfamiliar terms in their explanation. Help screens should explain things simply, giving the user examples.

Most users perform a variety of tasks using the software and are expert at some tasks but novice at others. Besides providing help when the user needs it, the software should recognise and anticipate the user's goals and offer assistance to make the task easier. If you have used the integrated package Microsoft Office, you will have come across Wizards that help you through some of the more complex tasks.



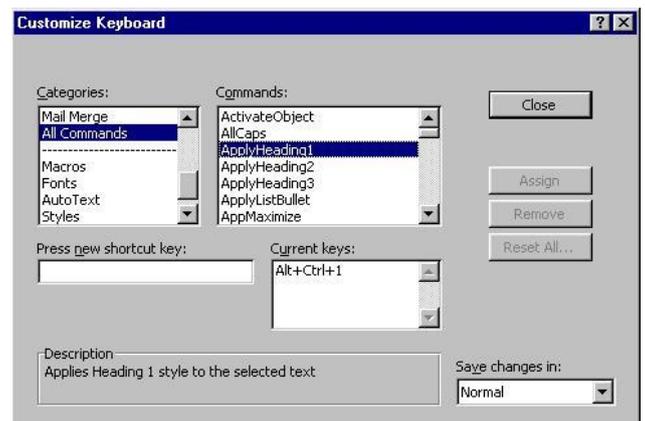
The screen shot opposite shows the first step of the Chart Wizard in Microsoft Excel. Clearly this wizard gives help to the novice user but also allows the expert to control every aspect of creating graphs.

Software should allow novice users to gain confidence in exploring the system. They should know that it is possible to try an action, view the result, and undo the action if the result is not what they wanted.

Providing short-cuts for experts

Expert users, many of whom type quickly, are able to memorise commands composed of combinations of keys and this saves them time compared with using the mouse and clicking on icons and pull-down menus.

Novice users are more likely to want to use the mouse, so many packages provide alternative interfaces.



Making use of human long-term memory to maximise efficiency

To develop software which is easy to learn, developers must understand how learning occurs.

Human thinking, in many ways, is like a computer with different memory areas. There are three main areas: the sensory register, short term storage and long-term storage.

The sensory register reacts almost immediately to stimuli to our senses. For example, we quickly move our hand away from a source of heat.

Short-term storage is where data is held temporarily. An example of this is when we ask someone for a phone number without writing it down and then remember it for the short period it takes us to dial the number.

Long-term storage is for those things we need to remember over a longer period. The more often anything committed to long-term memory is rehearsed, the longer it will usually be remembered for.

The human/computer interface (HCI)

An interface is a point of contact between two systems, and in the human/computer interface, this is the link between the computer and the user.

Human/computer interfaces are as different as their users. Some operators are passive and will only react in accordance with the computer's instructions whilst others will make enquiries, issue commands and get the computer to do things for them.

The response to an error message depends upon the experience of the user. Untrained or inexperienced users will often be scared of error messages and unable to know what to do without help; experienced users are usually able to do something about the problem themselves.

Improving the human/computer interface

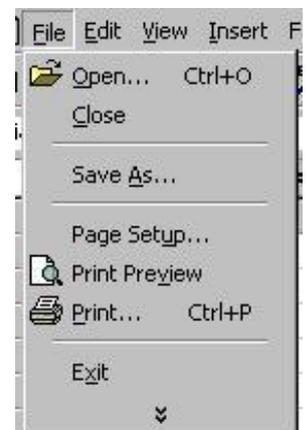
We have already looked at what is involved in human-computer interaction. Now we will look in more detail at the things that can be done when designing an interface to improve it for the eventual user.

Command structures and menus

Command interfaces make use of commands which the user enters via the keyboard to accomplish certain tasks. The main advantage of this approach is that the user can achieve a lot with one simple command. For example, in the command-driven operating system MS-DOS, the user issues commands such as `FORMAT A:' to format a floppy disk in the A drive. Commands are quick if you are an experienced user but prone to typing errors for the novice user.

When designing a menu system there are a number of things to bear in mind:

- There should be an opening menu, or general menu, from which other menus can be accessed, but you should never go more than three or four levels deep.
- The grouping of items in each menu should be natural and comprehensible.
- There should be either letters or numbers that allow the user to gain quick access to a particular item in the menu.



Menus are ideal if operators are likely to receive very little training or are infrequent users of the system, or are possibly unfamiliar with the terminology of the system.

Screen design

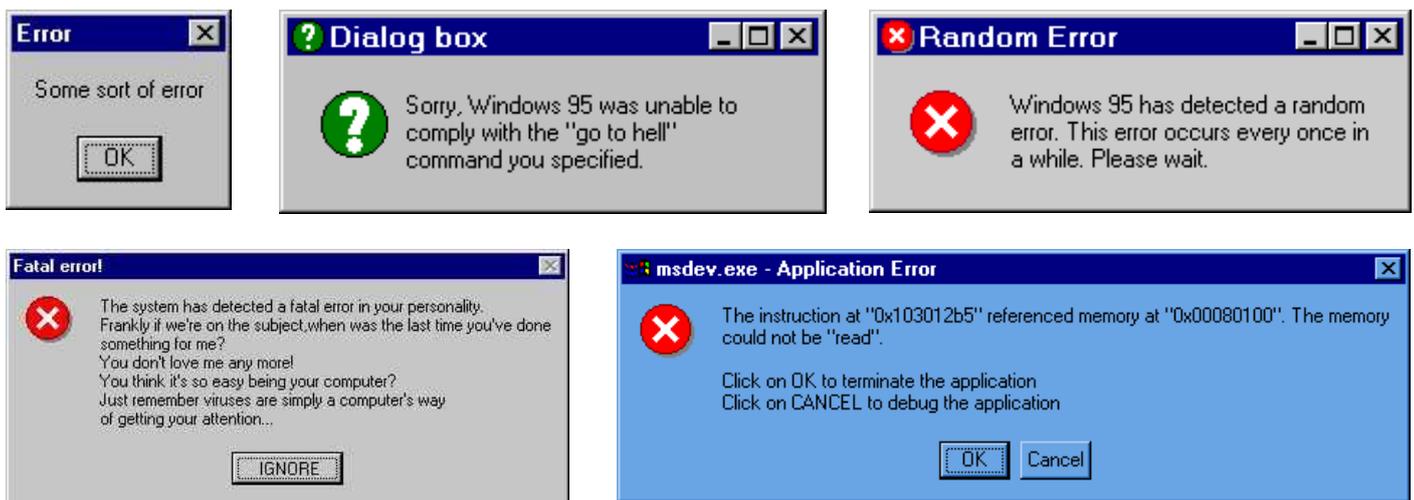
If a source document, such as an application form or an order form, is used to supply the information, then the input screen should mirror the form so that time is not wasted matching an item on the form with its equivalent on the screen. It is much better if a single screen can be used rather than multiple screens, even if it means that the screen is cluttered. It is also important that screen designs are consistent across all the departments in an organisation.

Here are some tips on screen design:

- Put the user in control.
- Do not overwhelm the user's memory. It is a good idea to put a nine-item limit on menus and lists.
- Do not overwhelm the user's senses: use colour and animation wisely.
- Make the design consistent.
- Make sure that help is never more than one click away.
- Layers should be no more than three deep.
- User mistakes should be easily rectifiable. The system should enable users to go forwards and backwards, undo and redo, enter and exit, with ease.

Error messages

Error messages can be infuriating to the user if they just say what is wrong without offering any suggestion as to what to do about it. On the other hand, there is a limit to the help the package can provide since the computer cannot mind-read. If an error message is given and it is impossible for the computer to anticipate what was intended, then the help facility should provide some guide to the likely problems.



The above examples clearly show how un-helpful error messages can be. In essence you should consider that any error messages developed within your system should be understandable by novices.

Availability of help

All modern software has a help facility but its usefulness varies. A poor help facility can serve to add to the confusion of an inexperienced user. Most help facilities enable the user to search for help on a certain topic; they sometimes look at the context in which the user requires the help. The human/computer interface can do this because it looks at the way the help query is phrased.

User-friendliness

User-friendliness is usually taken to mean the ease with which the software may be learnt and used. User-friendliness is a personal thing what one person finds user-friendly another person may find off-putting or annoying. When assessing the user-friendliness of software, it is best to get the comments of a range of users with different IT skills.

Ease of learning

There are a number of factors that make software easy to learn, including:

- Make the operation of the package similar the operation of other packages (*this is a significant factor that makes Windows-based software easy to learn*).
- Have an on-line tutorial that takes the user through the basics of the package. This should be interactive and make use of multimedia to add interest and involve the user.
- Provide a user-friendly manual, preferably written with the novice user in mind.
- A solution based on one of the popular packages for which there are many books and training guides facilitates learning.
- The software should be able to anticipate what the user is trying to do and offer appropriate help.
- The user should be able to get out of trouble easily, for example by pressing the ESC key. This will encourage the user to experiment and not worry too much about the consequences.
- Involve lots of users at an early stage so their advice can be obtained if needed.

The features of a sophisticated HCI

The features to consider including when providing a sophisticated human/computer interface are outlined below.

On-line help facility

All software packages should have an on-line help facility, where users can get help from the package rather than having to look through manuals or user guides. But bear in mind that On-line help packages increase the size of the application.

The screen shot opposite shows the on-line help screen in Microsoft Word. Typing in a word or phrase produces information on that topic.



Graphical user interface (GUI)

GUIs need a lot of main memory to enable them to run at a reasonable speed. The use of Windows-based operating systems and applications software has pushed up the average memory requirements for computers.

If you are considering writing a program or developing a system with a GUI, you should always bear in mind the extra resources it requires, which could limit its use for organisations that have older hardware and thus be unable to run your software. One way around the problem is to use only a few pictures, fewer graphics, and to limit the range of colours used.

Increased numbers of ways of performing the same operation

Many user interfaces offer more than one way to achieve the same result, with the choice left to the user. For example, a novice may prefer to use a pull-down menu or click on an icon to print out a file, whereas the experienced user may find it faster to issue a command using a sequence of keys, such as Ctrl+P

Multi-tasking capabilities

Most modern operating systems support multitasking, which makes it easy for the user to switch between applications. Multi-processing, where more than one application is open at any one time, places great demands on the processing power of the chip as well as the main memory requirements. As users demand this facility, faster processors and more main memory are needed.

Faster searching of help files

Chips with a higher clock speed are able to search for on-line help faster and display the results sooner. If help does not appear almost immediately, users may be put off using it.

The resource implications of sophisticated human/computer interfaces (HCIs)

Any sophisticated HCI is going to push the existing technology to the limit as graphical user interfaces need faster processors and more main memory. The easier the interface is to use, the more demands it places on the computer. As we move towards speech recognition systems (where the user can simply speak to the computer) the greater the required level of sophistication of both hardware and software.

Some user interfaces are quite expensive in terms of hardware and software. For example, touch-screen technology enables people who may have never used a computer before to find out about a range of products and services, but this requires an expensive screen. CAD work requires maximum use of the screen uncluttered by toolbars and menus, so these are transferred to an expensive graphics tablet instead.

Resource implications for the processor

The greater the demands placed on the processor by sophisticated operating systems and applications software, the slower they will run. Processors are continually being developed to cope with the demands placed on them by new software. Graphics-hungry applications stretch the capabilities of the chip, and to run such software quickly requires a processor with a high clock speed.

Resource implications for the immediate access store (IAS)

To manipulate large graphics files on the screen the IAS (i.e. the main memory) needs to be large, otherwise the system will be very slow and frustrating to use. Having a large main memory [e.g. 256 MB instead of 128 or 64 MB) means that many windows can be opened at the same time without any appreciable loss of speed.

Resource implications for backing storage

The large files associated with a sophisticated HCI will need a high-capacity hard drive as well as high-capacity removable storage (not floppy disks) such as a Zip drive to enable backup copies to be taken.

Resources for development

If a company is writing software to do a particular task, a sophisticated HCI will take more time to develop and test which will add to the cost of the project. However, this cost should be balanced against the lower training and support costs incurred as a result of users not getting into difficulty so often. With a good HCI they should be able to solve their own problems more often without ringing the help desk.

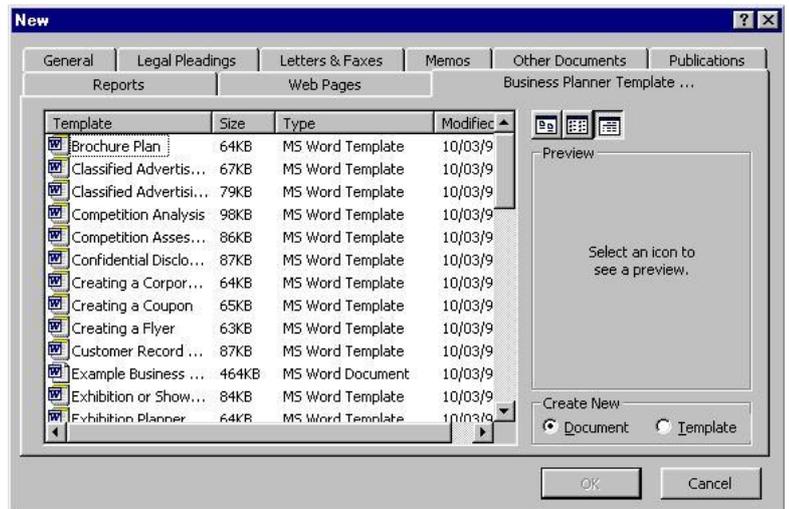
The implications for customising software to develop a specialist HCI

Most software packages, for example Microsoft Word, allow users to customise the software to suit their needs. By taking some time initially to look at the customisation options you can often save time in the long run. For instance such options in Word include:

- You can create new toolbars or menus by adding icons or groups and getting rid of any which you seldom use. In some cases, the software is automatically configured on the basis of what is used most often. In other words, it puts only those items that you use regularly in the toolbars.



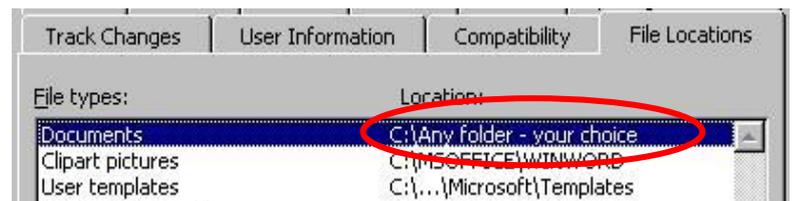
- You can customise your documents by altering templates (which come with Word) to suit your preferred document style. These template files give your documents a consistent style.



- You can alter the frequency with which automatic backup copies are made.



- You can alter the directory in which Word stores your data files, and thus avoid having to use the 'save as' function (which wastes time as you must select the directory each time).



- It is possible to change the appearance of the Word screen by either hiding or displaying screen items such as the toolbars, rulers, the menu bar, etc.

Customising

Customising is often used to simplify the HCI. For example, you may develop a database using the relational database Microsoft Access. As its developer, you will understand it fully, but will it be as simple to use for a novice? The user's job might be simply to input orders or stock details into the system, and for this task they do not need to understand the intricacies of the package. Instead, they need to be kept away from the interface supplied with the package, since this is really only suitable for experienced users. To do this, you as the developer have to develop menus from which the user can select only what they require for their particular job.

Design of forms for data entry (NOTE: Consider these factors when designing your forms for your IT06 Major Project)

Forms are used to enter the data into some packages (such as databases) in a user-friendly way. These forms must be designed carefully, since some staff will be using them for several hours a day and entering large amounts of data. Designing this type of human/computer interface is something anyone who develops databases will need to do.

Suppose you are asked to design a form for entering personal details of sports club members into the computer. As the personal details come from the members themselves, you need to ask them to complete an application form and then use this as the source document for keying the data into the database. There are a number of things that can be done to make it easy for the person who does the keying in. They include:

- The most important fields (i.e. the primary key and keys) should be situated in the most prominent place (usually the top left of the screen).
- The fields on the database form should be in exactly the same positions as the fields on the application form. This means that the person keying in the data knows exactly where to look.
- Help menus that pop up when the cursor is moved to a field can be included; these should give an indication as to the type and format of data required.
- Suitable fonts and font sizes should be used. Do not use many different styles and sizes – go for consistency.
- Colours should be used very carefully. Do not use bright colours inappropriately and choose colour combinations that work together.
- An inexperienced user might not understand specialist terms, so these must not be used on the form.
- There should be a limit to the number of fields on one screen; otherwise the user could be overwhelmed.
- Validation checks should be included, along with useful validation messages that appear if the wrong data is entered or it is entered in the wrong format.

Note in the example shown opposite. Information has been grouped into areas of the screen where there is a collective relevance.

There is also a logical flow when inputting the data starting from the upper left leading to the bottom right of the screen. Such as a book would be read.

Here is a quote that sums up the consideration needed when developing HCI:

'To the vast majority of mankind nothing is more agreeable than to escape the need for mental exertion ... to most people nothing is more troublesome than the effort of thinking'. (Bryce 1888)

ACTIVITY

1) Describe three of the features of a sophisticated user interface that makes it easy for a **novice** to use.

2) Describe three features of a sophisticated user interface that makes it quick for an **experienced** user.

3) Give three psychological factors which govern how people interact with computer systems.

4) What are the factors you would need to take into account when designing a screen layout for a database application?

5) What are the resource implications for providing a sophisticated human/computer interface?

6) A supermarket chain has recently implemented a new stock control system in each of its branches. This has affected those staff who have not used computer systems before. Many of the staff have described the system as being 'user friendly'. However, when the package was implemented in one particular store, it was not well received by its staff.

(a) Give four features of software packages that would merit the description 'user friendly'.

(b) Both physical and psychological factors can influence how people interact with computer systems. Both may have contributed to the poor reception of this system in that store.

(i) Describe two such physical factors.

(ii) Describe two such psychological factors.

Notes: