MOUNT KENYA UNIVERSITY
VIRTUAL CAMPUS

HUMAN COMPUTER INTERACTION

STUDY MANUAL

Chogi

2012

Contents

CHAPTER ONE

HUMAN COMPUTER INTERACTION OVERVIEW

What is human-computer interaction?

Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them. 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.

The History of Human-Computer Interaction

The dawn of personal computing in the early 1980s was the beginning of a rapidly evolving technology stranglehold on everyday life. The mechanical world of typewriters, dedicated word processors and adding machines with cranks were quickly left in a wake of microchips. The good news: you could do more. The bad news: you can be caught in a mire of complex and often-confusing computer-based equipment.

The study of Human-Computer Interaction (HCI) finally took center stage in the mid-1990s as the World Wide Web, e-mail, and Windows 95 burst upon the scene. Over the years, computers and programs have become easier to use. Computer programs have become more user friendly, taking advantage of "point and click" design. Minimal mouse clicks are only part of the picture to improve the computing experience. Cleaner, less cluttered "work spaces" for software are the heart of HCI. Users can focus on the tasks at hand (i.e., doing whatever it is they want to do with computers without worrying about how to "make things work").

HCI was a major factor at the Xerox Park Research Project in the late 1970s, even if the people involved weren't quite sure initially what HCI was (or the monumental impact it would ultimately have). Those researchers made pioneering efforts studying how people interacted with technology. They then redesigned software (and computers) to improve the "computing experience," boosting productivity. Mouse technology and "desktops" with icons (primitive as they were compared to today's standards) made it easier for people to work with technology that was soon to change the computing landscape of our daily lives in the 1980s.

What is HCI?

HCI has to do with the space that is created for you to work with technology. I'm not talking about space in the traditional sense of "floor space." I'm referring to the space that envelops you the minute you start concentrating. That space is what you become "submerged" in when you interact with a computer (or any kind of technological device you need to control).Think about all those flashing "12:00" digits on VCRs when they became a part of outlives in the 1980s. People quickly figured out how to press the red record button on the VCR to record a program "on the fly." But it was a much different story when it came to performing such tasks as programming the VCR and setting channels. The typical reason for the futility many people felt while trying to use a VCR was confusing or poorly written instructions. The same is true for digital watches. Setting (or turning off) an alarm on a digital watch can be difficult. And, it really all comes back to HCI. The steps to accomplish tasks with electronic equipment have become more intuitive, because the designers of the products now look very closely at HCI and the space they are creating for you to work in. Problems setting VCRs and digital watches are only a small part of the picture. Even today, people don't use many of the features on their cell phones or answering machines. The reason is that they just can't figure out what to do. After just a few attempts a person usually gives up on taking advantage of the "bells and whistles" that sounded great at first.

Affordance

Affordance is the essence of HCI. Simply put, affordance is the knowhow and skill set you brings to any new situation. *Here's an example of affordance:* You rent a car. Even though the car is a model you have never driven, when you climb into the rental car you already know how to press the button to roll down the electric window. However, the affordance you "bring to the table" in such a situation isn't going to do you much good if, instead of pressing down on the button, you need to pull up on the button to make the window work. *Here's another example:* There's a very good reason that most rental car companies do not offer cars with manual transmissions. Anyone who has a license has the skill necessary to drive a car with an automatic transmission. But you're driving skills would do you no good if you couldn't use a clutch or shift. Therein lays the conundrum for HCI. Because there is no one specific standard way to create products, there is no magic way to create user-centred design. The goal that HCI achieves is to create something the user can work with successfully. A person must be able to work within an interface (or space) intuitively. Affordance is something you keep adding to over time.

You have most likely interacted with technology in positive and negative ways. A good experience, where you can accomplish what needs to be done with minimal heartache, isn't something that can be listed in "10 easy steps" for product design. The evolving study of HCI is the major force leading to a better working experience in those technology spaces you inhabit daily.

HCI on the Internet

The Internet is riddled with great, as well as bad examples of HCI. When you visit a Website where it is easy to find your way around and seemingly effortless to accomplish tasks, you are experiencing a good example of HCI. The United States Postal Service's web site (**http://www.usps.com**) is one of the single-best models for HCI. When you visit that

Web site, you'll quickly discover that everything seems to be in the right place. Of course, you could easily put together a list of Web sites that are bad examples of HCI. I'd recommend you visit **http://www.websitesthatsuck.com** to see a regularly updated list of spaces that don't even come close to successful HCI.

HCI and Help Systems

There is no way your affordance can completely prepare you for functioning in today's technology-based world. The more you work with technology, the better you will get at "figuring things out." However, to maximize your potential, you need access to information about the technology. Although there are many vehicles you can use to convey information, a Help system is a popular choice. Simply put, a Help system allows a user to focus on working with a program without being burdened or lost when it comes to finding answers..Flash Help system's (pictured below) are similar to Web Help but feature ore dynamic animation effects.HTML Help, Web Help and Flash Help systems *can* contain a table of Contents, Search, Index and Glossary tab. Any of these tabs can be set to be the *default* tab (the tab that appears by itself when the Help system starts). The Table of Content stab typically contains books and topics conveniently listed in a point-and-click format. While easy to use, the structure of the table of Contents is dependent on the thought process of the Help author. As a result, finding information in the Table of contents is not likely to be intuitive. The user would have to know what the Help author was thinking when the author created the Table of Contents.

Most users quickly discover that the Table of Contents, while nicely structured, does not allow a user to find information quickly. Instead, users tend to spend more time on the Index tab where users find an alphabetical list of keywords. The user can type a keyword and, if the keyword is part of the Help system, the user is presented with associated topics. Of course, one drawback to the Index is that a user would have to know which keyword to type to use the Index effectively. The Search tab, on the other hand, allows a user to find any word in any topic simply by typing the word and clicking "Search." For these reasons, a Help system that uses either the Index or Search tab as the default, instead of the Table of Contents, would be an example of good HCI.

HCI and Captivate Projects

When it comes to computers and technology, people tend to learn better "by doing."That's why most professional training companies offer any, or all, of the following: certified trainers, one computer per learner, step-by-step learning books, and computer based simulations.

During this book you will learn to use Captivate to create computer-based simulations. When creating your simulations, keep in mind that for each task to be accomplished it is best to first explain what will be done and the impact it will have. Your simulations will need to make clear exactly what is going to be achieved and how that fits into the overall picture for the knowledge-base the user hopes to gain. Knowing this kind of information will make it much easier for the learner to understand, follow and complete the simulation.

Here's an example of employing HCI with your finished projects: Your projects can contain a play bar that the user can use to stop, pause and move through your project. As the developer, you will have limited control over what tools the play bar will contain and where the play bar will appear on the users screen. For instance, the play bar can be forced to appear in the upper left, top centre, upper right, bottom left, bottom centre or bottom right of the project. Once you establish the play bar's position, users will not be able to hide it, edit it or move it. For that reason, the appearance and position of the play bar is critical, yet often overlooked.

Most people who use your projects will first look in the upper left portion of a window. This area of the window is known as the "hot spot." In newspaper publishing, the upper left part of a page is often reserved for the most important news items. Positioning the play bar can be a tough decision. Should you position the play bar in the upper left, or would that be a distraction? How about positioning the play bar in the bottom centre of the user's window. That's not a bad idea until you hear from users that they didn't know there was a play bar because it was too far down the page. Unless it would be blocking important screen elements, I recommend placing the play bar in the **upper right** of the project. In my experience, this kind of positioning garners the most favourable reviews.

The bottom line where your projects are concerned is that you should create your projects with HCI in mind. If you do, your projects will build on what the user already knows and your projects will add to the success a person will have when working with technology—new and old.

On-line Meetings and Distance Learning

HCI has become a major factor in computer-based training and on-line meetings for corporations and government organizations. Corporate employees can "attend" meetings anywhere in the world by taking advantage of electronic (on-line) meetings. The HCI aspects of such meetings build on the user's skills of knowing how to use a computer and software (the affordance factor). Electronic meetings allow users to view documents on-line, which is an ingrained skill for anyone who regularly uses a computer. Add to that video and sound and you allow meeting attendees to interact in much the same way they would if everyone were actually in the room. Learners can now take classes and be part of workshops without leaving home. Known as **distance learning**, these kinds of courses have great impact in rural and geographically diverse areas. HCI manifests itself in distance learning via tools such as instant messenger and e-mail that allow learners and trainers to exchange information rapidly. Most learners begin a class knowing how to type on a keyboard. Creating e-mails and sending instant messages is simply adding to the learner's affordance.

Electronic classrooms also allow homework assignments, as well as tests and certifications, to e more easily distributed and processed. Additionally, the tutorials and training aids to prepare for such testing can be handled effortlessly and effectively. The success of electronic meetings and learning has come from HCI being interwoven into goals, planning, development and implementation. By building on skills learners already have, electronic meetings and classrooms have become productive tools for corporations and education institutions.

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,

EXAMPLE

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 is 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

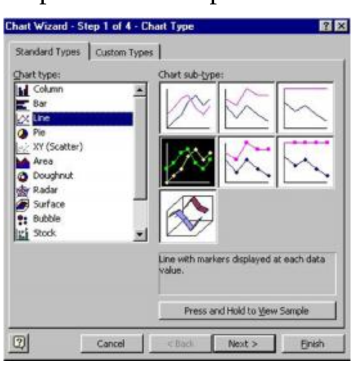
Many people, when introduced to computing, are afraid to try things out. Some people never seem to grasp the operation of devices such as video recorders, washing machines, etc. Many of these people find it difficult to understand manuals and instruction books. The human/computer interface should therefore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

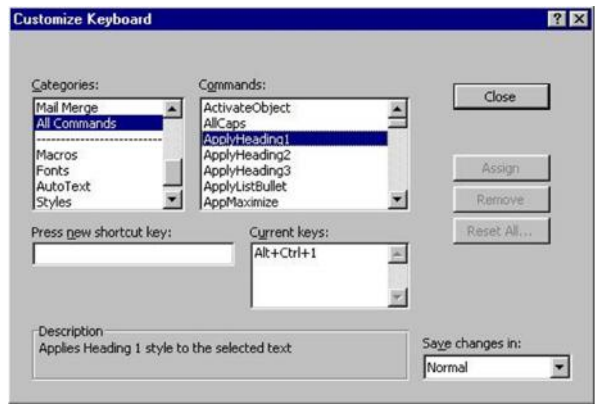


Most users perform a variety of tasks using the software and are expert at some tasks but novice at others. Besides providing help when the user needs it, the software should recognise and anticipate the user's goals and offer assistance to make the task easier. If you have used the integrated package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

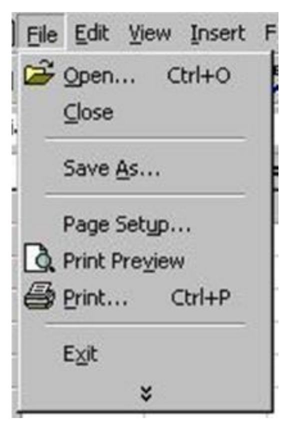
The response to an error message depends upon the experience of the user. Untrained or inexperienced users will often be scared of error messages and unable to know what to do without help; experienced users are usually able to do something about the problem themselves.

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

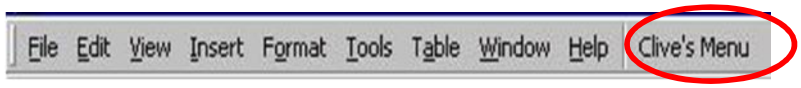
The large files associated with a sophisticated HCI will need a high-capacity hard drive as well as high-capacity removable storage (not floppy disks) such as a Zip drive to enable backup copies to be taken.

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.

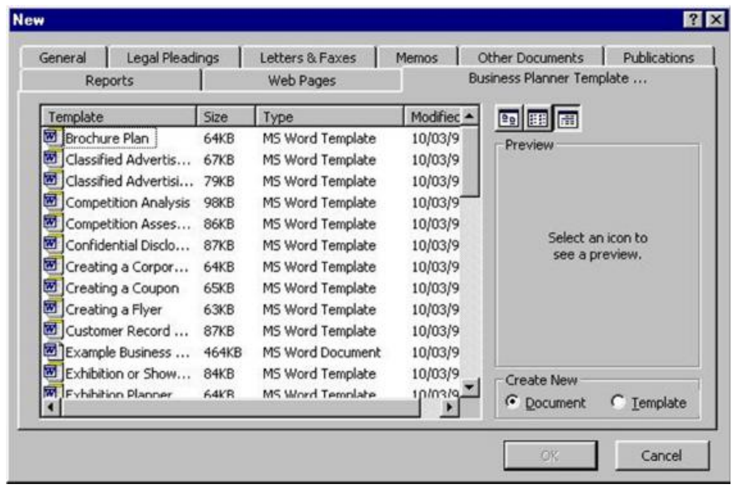
The implications for customising software to develop a specialist HCI

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:

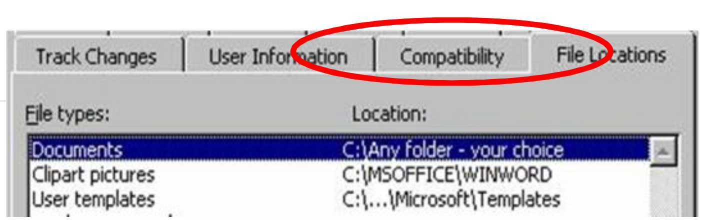


- You can create new toolbars or menus by adding icons or groups and getting rid of any which you seldom use. In some cases, the software is automatically configured on the basis of what is used most often. In other words, it puts only those items that you use regularly in the toolbars.

- 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.

Customizing

Customizing 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

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 amount 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.

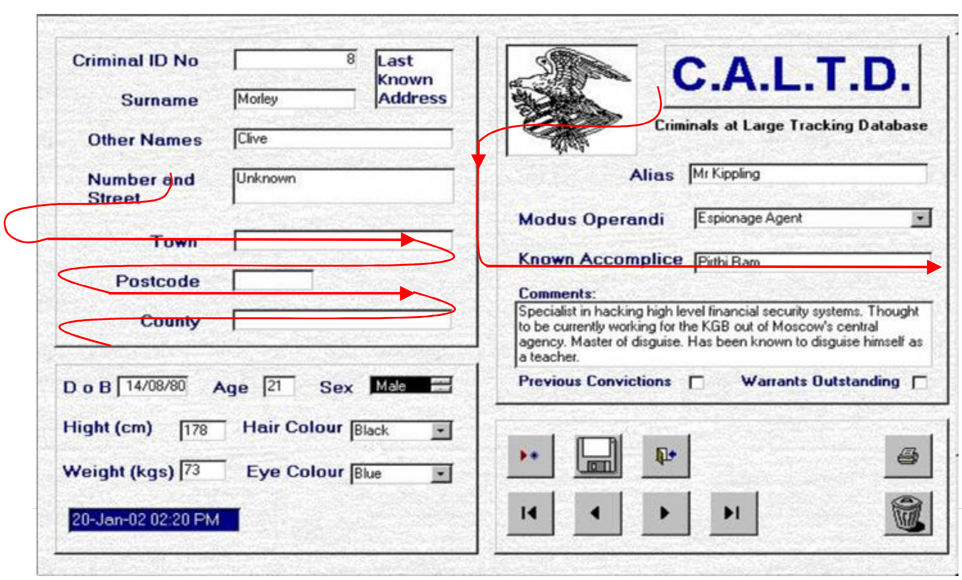
- Colours should be used very carefully. Do not use bright colours inappropriately and choose colour combinations that work together.

- 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)

CHAPTER QUESTIONS

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.

7. 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 maximize efficiency.

CHAPTER TWO

USER INTERACTION DESIGN

Design Rules

The goal of interaction design is to design for maximum usability.Design rules are rules that a designer can follow in order to increase the usability of the system/product e.g., principles, standards, guidelines.important consinderations to make are:-

- What are the differences among principles, standards and guidelines. Hint: they have differences based on level of abstraction/generality and level of authority e.g., principles are rather general, standards are specific.
- Principles: Example - usability principles by Dix et al (HCI book)
- Standards: They are often set by national (eg British Standards Institution) or international bodies (ISO). Example [of standards] - ISO 9241 "Ergonomic Requirements for Office Work with Visual Display Terminals (VDT)s"
- Guidelines: Example - Smith and Mosier's "Guidelines for User Interface Software" [MITRE Corporation 1986].
- Design rules should be used early in the lifecycle [e.g., during the design; note that they can also be used to evaluate the usability of the system]

Usability Principles

1. Learnability: the ease with which new users can begin effective interaction and achieve maximal performance.
2. Flexibility: the multiplicity of ways the user and system exchange information.

3. Robustness: the level of support provided to the user in determining successful achievement and assessment of goal-directed behavior.

1. Learnability

The ease with which new users can begin effective interaction and achieve maximal performance.

- Predictability: support for the user to determine the effect of future action based on past interaction history.
- Synthesizability: support for the user to assess the effect of past operations on the current state.
- Familiarity: the extent to which a user's knowledge and experience in other real-world or computer-based domains can be applied when interacting with a new system.
- Generalizability: support for the user to extend knowledge of specific interaction within and across applications to other similar situations.
- Consistency: likeness in input-output behavior arising from similar situations or similar task objectives.

2. Flexibility

The multiplicity of ways the user and system exchange information.

- Dialogue initiative: user freedom from artificial constraints on the input dialog imposed by the system; user vs system.
- Multithreading: the ability of the system to support user interaction for more than one task at a time.
- Task migratability: the ability to transfer control for execution of tasks between the system and the user (consider e.g., spell-checking task).
- Substitutivity: the extent to which an application allows equivalent input and output values to be substituted for each other (values in input eg fractions/decimals, values in output eg both digital and analog, output/input eg output can be reused as input).
- Customizability: the ability of the user or the system to modify the user interface. (adaptability vs adaptivity) ?-initiated modification.

3. Robustness

The level of support provided to the user in determining successful achievement and assessment of goal-directed behavior.

- Observability: the extent to which the user can evaluate the internal state of the system from the *representation* on the user interface.

- Recoverability: the extent to which the user can reach the intended goal after recognizing an error in the previous interaction.
- Responsiveness:a measure of the rate of communication between the user and the system.
- Task conformance: the extent to which the system services support all the tasks the user would wish to perform and in the way the user would wish to perform.

Heuristics and Golden Rules

Jakob Nielsen's 10 Usability Heuristics

1. Visibility of system status: the system should always keep users informed about what is going on, through appropriate feedback within reasonable time.
2. Match between system and the real world: the system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.
3. User control and freedom: users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.
4. Consistency and standards: users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
5. Error prevention: even better than good error messages is a careful design which prevents a problem from occurring in the first place.
6. Recognition rather than recall: make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.
7. Flexibility and efficiency of use: accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.
8. Aesthetic and minimalist design: dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

9. Help users recognize, diagnose, and recover from errors: error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

10. Help and documentation: even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

Ben Shneiderman's 8 Golden Rules

1. Strive for consistency: layout, terminology, command usage, etc.
2. Enable frequent users to use shortcuts: support special key combinations and sequences, macros, abbreviations, ...
3. Offer informative feedback: for every user action, offer relevant feedback and information, keep the user appropriately informed, human-computer interaction.
4. Design dialogs to yield closure: help the user know when they have completed a task.
5. Offer error prevention and simple error handling: prevention and (clear and informative guidance to) recovery; error management.
6. Permit easy reversal of actions : to relieve anxiety and encourage exploration, because the user knows s/he can always go back to previous states.
7. Support internal locus of control: make the user feel that s/he is in control of the system, which responds to his/her instructions/commands.
8. Reduce short-term memory load: make menus and UI elements/items visible, easily available/retrievable, ...

[Donald] Norman's 7 Principles

1. Use both knowledge in the world and knowledge in the head.
2. Simplify the structure of tasks.
3. Make things visible: bridge the gulfs of Execution and Evaluation.
4. Get the mappings right.
5. Exploit the power of constraints, both natural and artificial.
6. Design for error.
7. When all else fails, standardize.

USER-CENTERED DESIGN (UCD)

UCD is also referred to as the *user-centered methodology* or *human-centered design* or *human-centered methodology*.

UCD is an approach to design that grounds the process in information about the people who will use the product.

UCD intends to ensure that the user is at the center during the design process in order to realize products that meet usability requirements.

UCD follows a series of methods and techniques for analysis, design, and evaluation of software products.

The UCD process is iterative and focuses on users through all the life-cycle phases in order to realize products that the user can use with effectiveness, efficiency and satisfaction (usable products).

There is an international standard that forms the basis for UCD. This standard (ISO 13407: Human centred design process for interactive systems) defines a general process for including human-centered activities throughout a development life-cycle, but does not dictate the specific methods.

"UCD is an iterative process whose goal is the development of usable systems, achieved through involvement of potential users of a system in system design." [Karat 1996]

"… user-centered design emphasizes that the purpose of the system is to serve the user, not to use a specific technology, not to be an elegant piece of programming. The needs of the users should dominate the design of the interface, and the needs of the interface should dominate the design of the rest of the system." [Norman 1986]

How do we do that?

Consider the lifecycle:

- Gathering and specifying requirements
- Early design (Prototyping, Evaluation)
- Improved/Refined [requirements and] Prototype, Evaluation & Iterations
- Implementation, evaluation, 'final' refinements [final refinements before release/optimal design realization]

How we could involve or incorporate the user into the lifecycle:

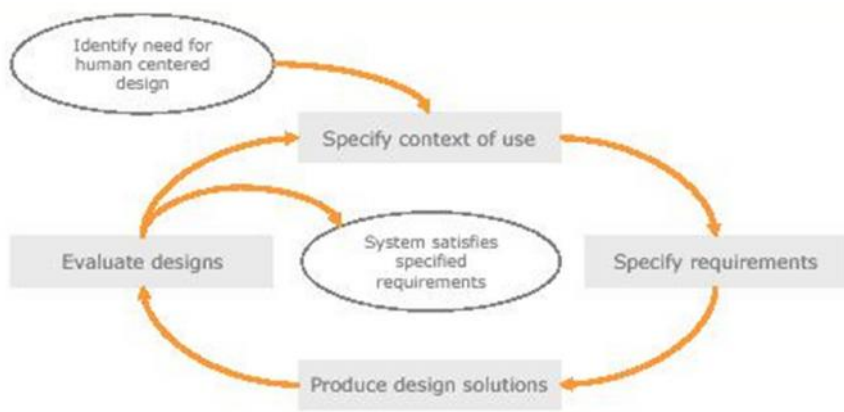
- Gathering requirements [interviews, questionnaires, personas, scenarios, direct observation, …]

- Early design (Prototyping then Evaluating) [Prototyping: low-fidelity prototypes eg paper-prototypes, video-prototypes, …; Evaluating: using esp expert-based evaluation methods (eg heuristic evaluation), questionnaires, …]

- Improved/Refined [requirements and] Prototype then Evaluate & Iterations [Prototyping: mid- then high-fidelity prototypes through the Iterations eg computer-based prototypes for instance using graphical and multimedia tools, wizard-of-oz, …; Evaluate: introduce user-based evaluation methods such as controlled experiments, and including the possibility of administering questionnaires, more structured interviews, direct observation, …]

- Implementation then Evaluation [Evaluation: using esp user-based evaluation methods such as controlled experiments, and including the possibility of administering questionnaires, more structured interviews, direct observation, …], 'final' refinements [final refinements before release/optimal design realization]

UCD illustrated



In order to support such i.e. in order to support UCD, the following are necessary [characteristics of UCD]:

- Know your users (characteristics, tasks, context/organization/environment in which they use the system)
- Actively involve users early and continuously
- Rapid and frequent iteration of designs with usability assessments
- Multidisciplinary team [this is inline with the ISO 13407 standard (Human-centred design processes for interactive systems)] ie It is important that the development team is made up from representatives of all the groups who have a 'stake' in the proposed software (stakeholders) e.g. it could include: managers, usability specialists, training and support staff, software engineers, quality assurance representatives and of course the users themselves.

Exercise: What are the benefits of UCD?

CHAPTER THREE

DIRECT MANIPULATION DESIGN PRINCIPLES

Direct manipulation

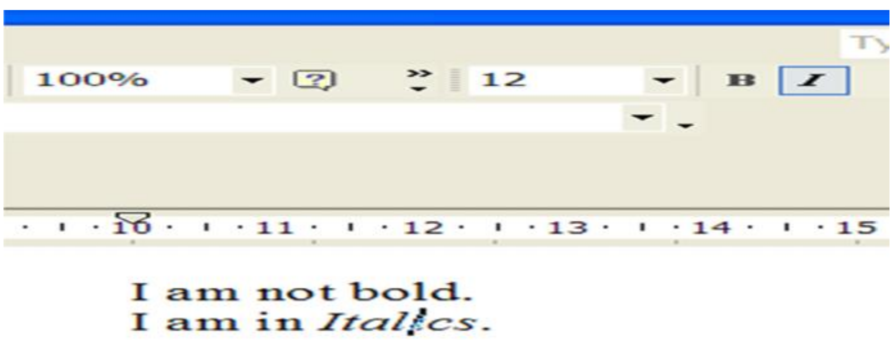
The term "direct manipulation" was coined by Ben Shneiderman.

Direct manipulation is an interactive technique characterized by:

i. Visibility of the object of interest
ii. Incremental action at the interface with rapid feedback on all actions
iii. Reversibility of all actions, so that users are encouraged to explore without severe penalties
iv. Syntactic correctness of all actions, so that every user action is a legal operation
v. Replacement of complex command languages with actions to manipulate directly the visible objects [and hence the term direct manipulation]

One of the consequences of the direction manipulation paradigm is that there is no longer a clear distinction between input and output – output expressions are used to formulate subsequent input expressions. [For instance, the document icon is an output expression in the desktop metaphor, but that icon is used by the user to articulate the move operation.]

Another example:



Note that in the above example, the visible objects for Bold and Italics are serving output expressions regarding the text, but they can also be used to provide input (to make the text bold or have it in Italics)

Well-designed direct manipulation interfaces may also engender enthusiasm and elicit enjoyment from the users because:

i. Novice users can learn the basic functionality quickly

ii. Experienced users can work extremely rapidly to carry out a wide range of tasks, even defining new functions and features
iii. Knowledgeable irregular users can retain operational concepts; in fact error messages are rarely needed
iv. Users can immediately see if their actions are furthering their goals and, if they are not, can simply change the direction of their activity
v. users experience less anxiety because the system is comprehensible and because actions are so easily reversible
vi. Users gain confidence and mastery because they initiate an action, feel in control and can predict system responses.

CHAPTER FOUR

INFORMATION ARCHITECTURE AND NAVIGATION

Information architecture

Information architecture is concerned with the structural design of information spaces.
An information space is some combination of things – objects, displays, people, signs, icons, sounds, etc - that is used to provide information.
An information space or artefact consists of two parts: a conceptual information structure and an interface onto that structure. It is only through an interface that the conceptual information can be accessed.
The designer decides what conceptual information is useful for the underlying activity – e.g. train times and stations – and then considers how to provide access to this information – e.g. paper timetable, electronic display.

Some characteristics of information spaces:

- Volatility: This is concerned with how often the types and instances of objects change.
- Size: This is primarily governed by the number of objects in the information space.
- Conceptual and physical objects: Conceptual objects are the objects we are ultimately interested in such as web pages, image files, document files, etc. Physical objects are the objects that we use to interact with e.g. mouse, browsers, menus, command buttons, etc.
- Topology: The conceptual topology will dictate where conceptual objects are i.e. how things are categorized. The physical topology relates to the movement between and through physical objects and how the interfaces have been designed.

- Distance: This refers to the amount of interaction required before one accesses the information (or accomplishes the task) of interest. Ideally we should keep distances short.
- Direction: When moving between menu items or instances of an entity, there may be a sense of next and previous.

Information architecture of websites:

It entails organizational/classification schemes and organizational structures

Organizational/Classification schemes

Organizational/Classification schemes describe a way of classifying objects in the information space (i.e. creation of categories/groups of objects).

1. Exact organization schemes: In exact organization schemes, the classification is clear-cut. Such schemes include: alphabetical (e.g. phonebook, library author index), chronological (e.g. calendar, diary), geographical (e.g. political, social).

2. Ambiguous organization schemes: In ambiguous organization schemes, the classification is not clear-cut. The schemes use subjective classification. For instance: In a supermarket, where can we find the tomato sauce? With the canned goods? Next to the pasta? In both locations? Ambiguous organization schemes can be based on: topic/subject (e.g. encyclopaedia), task (e.g. buy ticket, contact us), audience (e.g. information for staff, information for students), hybrid, etc.

Organizational structures

It is almost always certain that not everything will fit onto one screen/page.

Some decision will have to be made about how to break up the site in order to accommodate this constraint.

While *organizational/classification schemes* describe a way of classifying objects i.e. *creating categories/groups*,

organizational structures describe the *relationship between the groups*.

There are various standard organizational structures for that purpose – note that they tie in with the classification schemes chosen.

1. Hierarchical structure/tree: It arranges the information space with a single root at the top and a number of branches underneath, each of which may have several sub-branches. There is the big picture from which one can drill down to detail. For example: music > classical, rock, jazz, etc

Another example:

The arrangement is rather clear.

It can however be difficult to navigate from one lower-level menu to another i.e. there are no direct paths.

Another problem with trees is that no matter the classification chosen, some item may not fit nicely into it, and the designer may want to put under two or more headings (which would lead to a network structure – see below).

Question: In the context of hierarchical structures, what is the breadth vs. depth debate?

2. Network structure: These are structures in which the same item may be linked into different hierarchies. It is a *more natural structure* but a *more confusing one* for people to understand. Examples of network structures: hypertext links in which jumps can take place from one page to another irrespective of ordering, search interfaces where one specifies the search criteria in order to get results (note two main problems with search interfaces: the user needs to know what they are looking for and how to specify the correct search criteria).

Hypertext links

Search interface

3. Sequence – ideal for dealing with a straightforward task structure. For instance, when going through a registration process, when purchasing a product, when doing a survey, when boarding a flight online, when installing software, when applying for a job, etc.

Example:

Navigation

Navigation is concerned with finding out about, and moving through, an environment. It includes three but related activities:

- Object identification – which is concerned with understanding and classifying the objects in an environment.

- Exploration – which is concerned with finding out about a local environment and how that environment relates to other environments.
- Wayfinding – which is concerned with navigating toward a known destination.

Designing for navigation

The practical aim of navigation design is to encourage people to develop a good understanding of the space in terms of landmark, route and survey knowledge.

Another aim is to create spaces which are enjoyable, engaging and involving.

Signage

Good, clear and unambiguous signposting of spaces is critical in the design of spaces. There are three primary types of signs that designers can use: informational signs, directional signs, and warning and reassurance signs.

- Informational signs: they provide information on objects, people and activities.

  e.g.1:

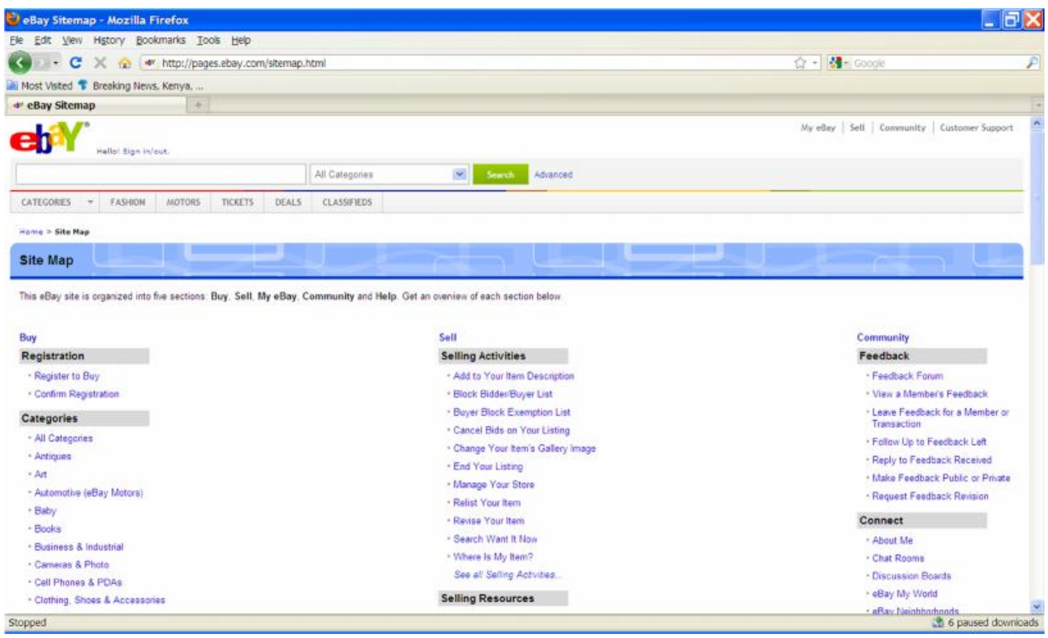
  e.g.2 labels: these are used for internal and external web links, headings and subheadings, titles, and related areas.

- Directional signs provide route and survey information. They do this often through sign hierarchies, with one type of sign providing general directions being followed by another that provides local directions.

  e.g.1:

e.g.2 labels: It is common to have a navigation bar across the top of a site which points to the main, top-level categories. Within each of these there are subcategories (eg as a menu on the left-hand side or as a drop-down).

e.g.3 site maps are useful since they display the structure and content headers of the various categories.

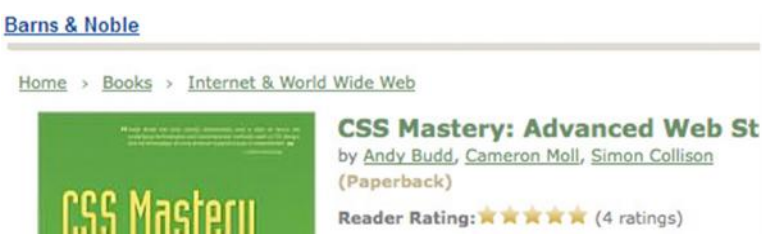


- Warning and reassurance signs: they provide feedback or information on actual or potential actions within the environment.

  e.g.1:

e.g.2 “You are here” sign, or by having current page highlighted

e.g.3 using breadcrumbs (is a common way of showing people where they are and how they could find their way back)



Maps and guides

Maps can be used to provide navigational information.

When maps are supplemented with additional detail about objects in the environment, they become guides.

Environmental cues and social navigation

Even in a well-designed environment, people will often turn to other people for information on navigation rather than use more formalized information artefacts. When navigating cities, people tend to ask other people for advice rather than study maps. Information from other people is usually personalized and adapted to suit the individual's needs. Sometimes we use a wide range of cues in order to manage our activities e.g. by observing the behaviour of other people, our friends, etc.

This is often supported using online social media features such as forums, blogs, chats, etc. We sometimes rely on other types of environmental cues e.g. assuming that the reception is at the front of the building, restrooms are located at the back of a restaurant, etc.

CHAPTER FIVE

PROTOTYPING

Introduction

A prototype is a concrete but partial representation or implementation of a system design.

Prototypes may be used for, among other things, to:

- demonstrate a concept in early design
- elicit feedback from the users or audience
- test details of a concept at a later stage

Prototyping helps us to achieve two main goals:

1. Reduce the cost and risk involved in implementing systems
2. Ensure the quality of the final system

Types of prototypes

Prototypes can be classified based on the precision level of the prototype relative to the anticipated final product.

It is considered that there are two major precision levels and therefore two main types of prototypes: low-fidelity prototype and high-fidelity prototype.

Low-fidelity prototype (or lo-fi prototype)

A low-fidelity prototype is a cheap, simplified mock-up of the user interface.

It is often common and used early in the design process.

Low-fidelity prototypes do not have to really look like the actual user interface, as long as they mimic or mock the same.

Since low-fidelity prototypes are cheap, in terms of money and time, designers can afford to have more cycles of evaluation and corresponding refinement.

High-fidelity prototype (or hi-fi prototype)

A high-fidelity prototype is as close to the anticipated final product as possible in terms of look and feel, and interaction.

The prototype accepts inputs like the actual user interface would, and responds to those events in the same way (e.g. displaying a particular window or message, changing state, etc.).

Question: Why do we talk of the possibility of having mid-fidelity prototypes?

Question: What other ways can be used to classify prototypes?

Requirements for prototyping tools

Here are some requirements for prototyping tools

- Ease of use: Prototyping tools should allow all members of the development team to participate in the development and refinement of the prototype.
- Fast turn-around: Prototyping involves making many small refinements to the user interface. Tools should allow designers to quickly make the changes and immediately see the interface in action again.
- Extensive control over prototype features. Prototyping tools should be very flexible. One of the purposes of prototyping tools is to try out new ideas. Tools should therefore support a large variety of designs, and should give designers extensive control over design details.
- Data collection capabilities: Ideally, prototyping tools should capture all the different kinds of information required during the interaction.

- Executable prototypes: Prototypes should ideally exhibit reliability close to real systems, because designers need to make them in order to increase the reliability of the information collected. An executable prototype is one that can respond to user input and provide appropriate responses. However, it is not always necessary that prototypes be connected to real data and that they perform real computations (e.g. simulations could be sufficient).
- Lifecycle support: Prototyping tools should help with all the stages of the entire design process starting with the early design stages through the advanced and detailed screen and behaviour design.
- Team design: Software products are developed by teams. Prototyping tools should support groups of people working together either simultaneously or asynchronously, and perhaps remotely.
- Version control: In prototyping, it is important to explore alternative designs. Many versions of a prototype might be built while exploring different alternatives. Designers might want to revisit previous designs, so keeping and managing prototype versions is important.

Prototyping tools

Paper and Pencil

Paper and pencil prototyping allows the development of paper prototypes, which are low-fidelity prototypes.

Paper and pencil prototyping is almost mandatory in early stages of design.

They produce low-fidelity prototypes which are used during early stages of design for conceptualizing and envisioning the application.

The method itself involves creating rough, hand sketched, drawings of an interface to use as prototypes of a design. User interface interaction objects such as checkboxes, text fields, and drop-down lists can be cut out of a sheet of paper and added to the interface to simulate the interactivity. These prototypes can then be evaluated by asking users to perform some tasks using the roughly sketched out interface, while designers observe their actions to gauge where they face problems, and what they like. This method is very successful at discovering usability issues early in the design process.

One of paper prototyping's main drawbacks is lack of interactivity. It can be countered by combining classical prototyping tools like HyperCard, PowerPoint, etc.

Facade tools

Facade tools allow designers to construct the screens of a user interface without a real application behind. Of course, each tool has its own domain of applicability, but globally, this category of tools takes the advantages of Paper and Pencil, by adding the basic interactivity of an executable prototype: a response to inputs such as a click on a button.

Facade tools create mid-fidelity prototype which are used to design and evaluate mostly the content, layout, and some interactive aspects, including navigation functionality.

These tools are usually easy to handle. They however do not produce reusable code.

Examples:

- Microsoft's PowerPoint
- Apple's Keynote
- Apple's Hypercard (for hypertext applications, but is getting old)

Macromedia Flash, Director, and Microsoft's Frontpage

Macromedia Flash and Director are powerful vector based development tools. Director is a more complex multimedia authoring tool than Flash. Adapted to large graphical components, Director is an effective tool for producing impressive simulations, visualizations, presentations and of course for prototyping user interfaces. Microsoft Frontpage offers a simple way of developing a website.

Flash, Director and FrontPage can produce any kind of prototypes, from low to high fidelity, including a full Flash-based application, depending on the depth of the work on the prototype.

User interface builders

GUI builders are tools developed to increase the productivity of user interface development teams, and to lower the cost of user interface code in both the development and maintenance phases.

They are the construction tools that give interface developers a drawing-like interface to specify the layout. Their main function is to save time by generating the executable code of the interface from a few mouse clicks. For instance: Microsoft Visual Basic, Microsoft Visual C++, which are part.

Microsoft Visual Studio software development suite, have a direct-manipulation user interface builder, in conjunction with a corresponding programming environment/code. The designer is able to create and place standard interface elements, set and modify their properties, and associate them with codes that execute when they are invoked by a user of the interface. The development suite/environment makes it easy for the designer to convert a Visual C++ project into an executable file that can be distributed and run by users who do not have the Visual Studio Environment on any computer that runs Microsoft Windows.

CHAPTER SIX

MULTI-MODAL INTERFACES

Introduction

Recall that there are five human senses: sight, sound, touch, taste and smell.

Since our interaction with the world is improved by multi-sensory input, it makes sense that interactive systems that utilize more than sensory channel will also provide a richer interactive experience. Moreover, users can interact with the system using the mode of interaction that is most appropriate to their abilities.

A multi-modal interactive system is a system that relies on the use of multiple human communication channels. Each different channel for the user is referred to as a modality of interaction. However, genuine multi-modal systems rely to a great extent on simultaneous use of multiple communication channels for both input and output.

Sound in the interface

Sound is beneficial in interaction in several ways;

- there is experimental evidence to suggest that the addition of audio confirmation of modes (eg by changing the keyclicks), reduces errors.
- it enables users with visual impairments to be able to interact with the computer
- it can convey transient information
- it does not take up screen space (making it potentially useful for mobile applications)

Exercise: What are the disadvantages of sound?

However sound is relatively little used in standard interfaces, and where it is used it is often peripheral to the interaction. Information provision is predominantly visual.

Exercise: What are the corresponding dangers if sound is not used and we primarily rely on visual?

There are two types of sound that can be used on the interface: speech and non-speech.

1. Speech

Language is rich and complex. The complexity in general makes speech recognition and synthesis by computers very difficult.

Speech recognition:

There are several practical issues that speech recognition faces.

- Background noise can interfere with speech recognition.
- Redundant or meaningless noise too can interfere with speech recognition.
- Speech recognition also has to deal with the uniqueness of a specific voice, different accents, and different intonations.

Situations where speech recognition may be appropriate:

- When a user's hands are already occupied (eg in a factory, at a construction site), speech recognition may prove to be the ideal input medium.

- Moreover speech input does not require the use of a cumbersome keyboard and is therefore appropriate in mobile situations.

- Speech input also provides an alternative means of input for users with visual and physical impairment.

Speech synthesis:

Speech synthesis complements speech recognition. The notion of being able to converse naturally with a computer is an appealing one for many users, especially those who do not regard themselves as computer literate, because it reflects their natural, daily medium of expression and communication. However, speech synthesis faces many challenges.

- We are highly sensitive to variations and intonation in natural speech, and are therefore intolerant of the imperfections in synthesized speech (ie imperfections as in the monotonic tones that the speech synthesizer may produce).
- Synthesized speech is transient, and therefore the spoken output cannot be reviewed or browsed easily.

- Spoken output is intrusive (consider, eg the user may have to use headphones).

Speech synthesis is applicable in several areas.

- For users who are blind or visually impaired, synthesized speech offers an output medium which they can access eg through screen readers.

- Speech synthesis is useful as a communication tool to assist people with physical disabilities that affect their speech.

- Speech synthesis is also used as a supplement to other output channels to enhance applications where the user's visual attention is focused elsewhere, such as warnings in aircraft cockpits, cars, etc.

2. Non-speech

Here are some advantages of non-speech sounds:

- Since speech is serial, we have to listen to most of sentence before we understand what is being said. However, non-speech sounds can often be assimilated much more quickly.

- Speech is language-dependent (and therefore a speech-based system requires translation for it to be used for another language group). However, the meaning of non-speech sounds can be learned regardless of language.

- Speech requires user's attention. However, non-speech sound does not require much attention and can make use of the phenomenon of auditory adaptation: background sounds are ignored unless they change or cease.

Disadvantage with non-speech sounds is that they have to be learned, whereas spoken messages are obvious (at least to a user who is familiar with the language).

Non-speech sounds can be applied in providing transitory information, providing status information on background processes, and supporting the visual mode and providing confirmation for the user.Some examples of non-speech sounds: Auditory icons, Earcons

Exercise: What are Auditory icons, Earcons?

CHAPTER SEVEN

VIRTUAL REALITY

Introduction

Virtual reality (VR) refers to the computer-generated simulation of a world, or a subset of it, in which the user is immersed.

Since the user has to "see" a new environment, a headset is usually used in a VR setup. It comes with an independent screen for each eye in order to give a 3D image. The headset can be a large, relatively cumbersome piece of head-mounted gear or even smaller, lighter virtual reality goggles [which are (the goggles) now available].

Having to produce and render realistic images in real time requires *vast amounts of computing power*. Moreover, the *resources for full realism are rarely available* (or even not available as yet).

Input to VR systems is often accomplished through a dataglove that captures gestural information.

Applications of virtual reality

Here are some applications of VR:

- Flight simulation – in a flight simulator, a full cockpit system is placed in a hydraulically supported container, with large screens replacing the cockpit windows. Images are generated and projected onto the screens, whilst the box can be moved rapidly in any direction hydraulically [ie by the hydraulic rams]. Flight simulators are used extensively in pilot training programs.
- Military training – the military is heavily involved in VR, allowing war scenarios to be fought out with great realism. The increased realism is intended to save lives later on, when, by being prepared, people are more able to cope with whatever arises.
- VR holidays – by walking into a VR environment, one can go on holiday to the tropical rainforest, go on safari, walk on the moon, etc.

Augmented reality

In augmented reality systems, electronic images are projected over the real world – virtuality and reality meet. [The head-up displays in many aircraft and even some automobiles can be regarded as an example of this, but the data in such displaces are not typically connected to the objects seen through them, and hence the blend between virtuality and reality is quite weak.A stronger sense of connection can be obtained using semi-transparent goggles. Users can move around in the real world and see real objects, but computer images are reflected off the inside of the glass and overlay the physical objects.

The great difficulty with such systems is in ensuring that the physical and virtual worlds are correctly aligned [or synchronized], a problem called *registration*.Example: In the electronics of a large aircraft, the wiring looms that run from end to end may include dozens of colored wires, each of which has to be routed to the right place. With augmented reality, the schematic wiring diagrams can be overlaid onto the physical wiring, helping the engineer to correctly identify and route each wire.

CHAPTER EIGHT

COMPUTER SUPPORTED COOPERATIVE WORK (CSCW)

Introduction

CSCW refers to collaboration between individuals via the computer.

The main distinction between CSCW systems and interactive systems designed for a single user is that designers can no longer neglect the society within which any single user operates. CSCW systems are built to allow interaction between humans via the computer and so the needs of the many must be represented in the one product.

Features and classification of CSCW systems

Two important *features* of CSCW systems are:

- the mode of interaction they support: this can asynchronous (ie occurring at different times) or synchronous (ie occurring at the same time)
- the geographical distribution of the users: this can be local (ie users are co-located in the same environment) or remote (ie users are different locations eg different buildings, countries, etc)

This leads to a four-category *classification* of CSCW systems

Synchronous-local (same time/same place):
Face-to-face eg in classrooms, meeting rooms

Synchronous-remote (same time/different place):
They are synchronous and distributed eg shared editors, video windows, video conferencing, instant messaging, etc

Asynchronous-local (different time/same place):

They take the form of asynchronous interaction e.g project scheduling, coordination tools, etc

Asynchronous-remote (different time/different place):

They are asynchronous and distributed e.g version control, bulletin boards, wiki, blogs, etc

CHAPTER NINE

HUMAN- COMPUTER INTERFACE

An interface is defined as the point where two objects meet.

Human/computer interfaces provide the means by which the user tells the computer what to do and at the same time the computer can interact with the human user by producing a response.

These interfaces are important because they determine the ease with which the computer can be used. When the manufacturer of systems software or applications software gets it wrong, then using the software can prove very frustrating and the user will be less likely to buy one of their products again.

The standard interface for inputting data into the computer is the keyboard with the computer giving its response on the screen. This is not the only type of human/computer interface, although it is the most common. There are many other systems that make use of IT and need a different type of interface.

Process control screens, computer games, cockpit controls on fly-by-wire aircraft, information systems which can be used by members of the public, all make use of innovative user interfaces.

Why Do We Need to Consider the HCI?

Good HCI means that new systems are easy to use and users are productive and make fewer mistakes.

System designers must think about:

- Who will the users be - primary school children or brain surgeons?
- What tasks will the system be doing?
- What environment will the system be used in - noisy (factory), quiet (office), changeable (paramedics change their environment and location every half an hour).
- What is technologically possible (voice data entry)

They then design a system for those users which uses appropriate:

- Input and output devices
- Navigation
- Validation
- Error messages and help

ACTIVITY 1:

Discussion – how many different types of interface can you recall? What type of response does the interface give?

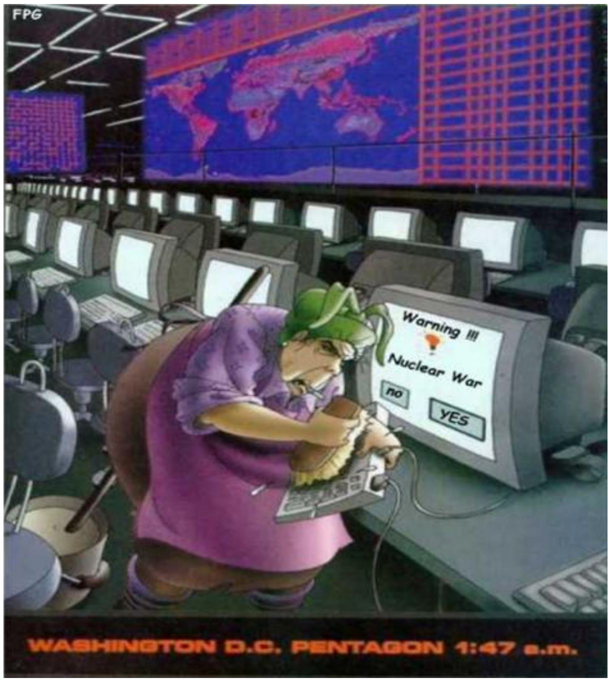
- Recall different approaches to the problem of communication with ICT systems and discuss the resource implications of sophisticated HCI.

- Discuss the implications for customising software to develop a specialist HCI.

Improving productivity

Studying the human computer interface is important from the point of view of improving productivity and therefore job satisfaction.

Also in certain instances there is the need to ensure that work practices are safe i.e. that interfaces do not mislead the user or indicate incorrect actions.



In essence there is the need to facilitate an effective dialogue between the user and the computer.

ACTIVITY 2:

Choose one of the interfaces listed below and discuss how it is specifically designed for the task. Look out for any features of the interface which are specific to its use.

- A video recorder handset
- The interface for a games console
- The interface used in a cash dispenser
- The interface between the driver and the controls of a car.

The different capabilities of humans and machines

Humans receive information about the outside world using the senses of taste, touch, sight, smell and hearing. The ideal interface between humans and machines should incorporate as many of these as possible.

TOUCH. Multimedia applications make use of our sense of touch via the keyboard, mouse, joystick and touch-screen monitors.

SOUND. By using speakers, the computer is able to issue instructions or even encouragement to the user. Sound may also be used to input instructions or data to the system by using voice recognition software.

Virtual Reality. Another interface, which makes use of as many of our senses as possible, is in the area of virtual reality. Virtual reality applications are programs that envelope the user within a simulated, three dimensional world of sight, touch and movement. With such an interface, the user is able to interact in a virtual world.

When user interfaces become sufficiently user friendly, communicating with computers will be almost like communicating with another human being. We are clearly some way from this at the moment but successive versions of new operating systems and applications software are making interfaces easier to use.

ACTIVITY 3:

Explain the type of interface that would be most suited to the following applications:

a) Looking at the flow of chemicals to a reaction vessel in a process control system.
b) Seeing the speed of a car or the amount of fuel left in the petrol tank on a car dashboard.
c) Giving the user of a cash machine instructions on how to use the machine.
d) Enabling a customer to see the services offered by a bank in the foyer of the bank.

Choosing the right interface

Suppose you are playing a computer game, such as guiding a Rally car around a track. The game's manufacturer and you will want to make the game as realistic as possible, as this will enhance your enjoyment. The graphics showing the track and the other vehicles can be made realistic, as can the actual performance of the car (cornering, braking, accelerating and so on). This all adds to the realism of the game, but the thing that can let the game down is the human/computer interface.

The worst interface would be the cursor keys and other keys to steer the car, change gear, accelerate and so on. A better interface would be a joystick, although this is not ideal as cars are normally fitted with steering wheels, gear sticks and foot pedals. However you can now actually buy such interfaces to make controlling the car as realistic as possible.

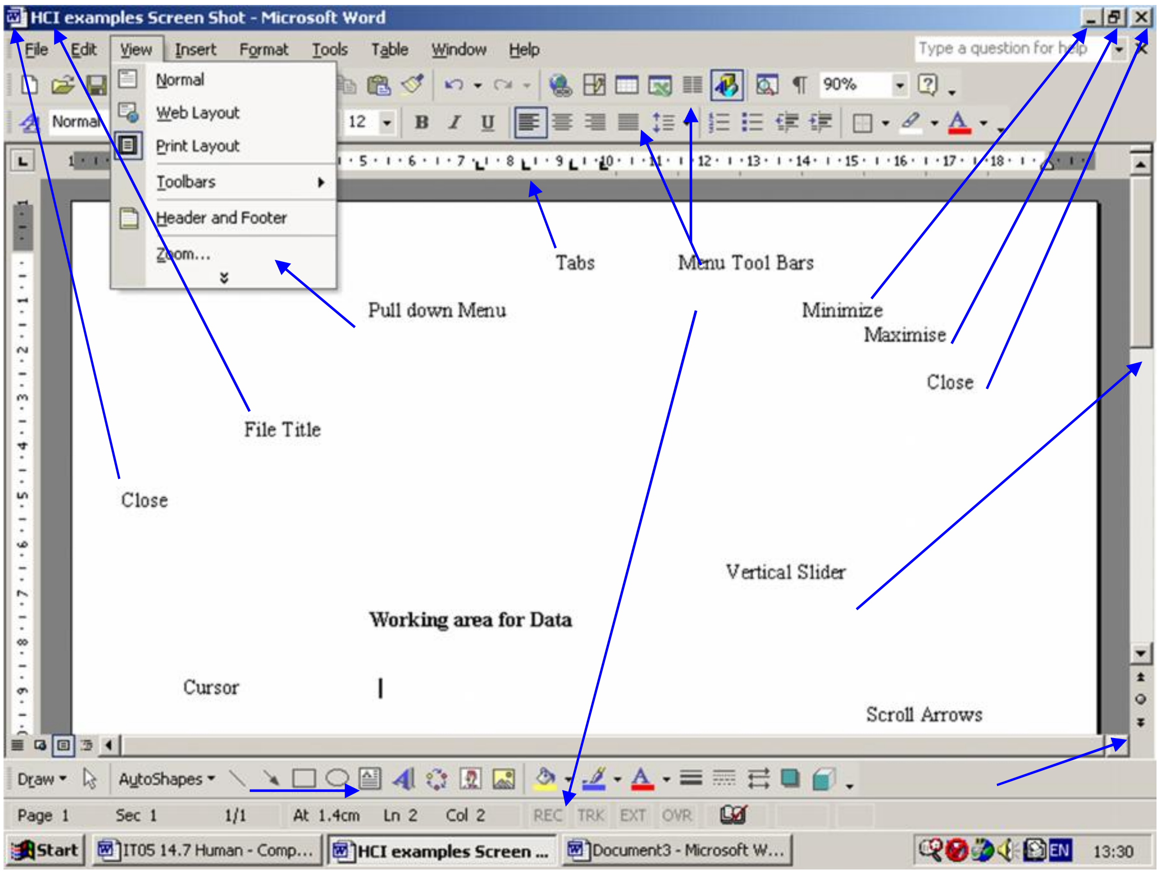
Graphical user interfaces (GUI)

A graphical user interface (GUI) is used by many manufacturers with their operating systems. Microsoft Windows 2000, ME, XP and the Apple Macintosh have GUIs. All the memory and file management activities are taken care of by these operating systems and it is possible for them to multi-task, with the user running more than one application at the same time. All these GUIs use a mouse to navigate around the screen; the mouse buttons are used to make selections from icons and buttons.

Dialogue boxes and pull-down menus are used as part of these interfaces. The work area is located in the centre of the screen and users are usually able to choose which toolbars, rulers and icons are displayed around this area. The user work area provides a moveable `window' through which one can see the data being worked on; to move the data, the horizontal and vertical sliders at the side and the bottom of the screen are used.

A typical window is shown in diagram below what makes windows-based software particularly easy to use is that the interface is standardised across different applications.

Some components of a typical window



Most windows computers will look similar to the previous screen shot and within any suite of applications there will be very identical icons, button bars, dialogue boxes and pull-down menus. This makes learning windows-based packages much easier.

A graphical user interface does more than simply control the hardware; it can be used by the programmer to influence how the user interacts with a program. In particular, it allows the programmer to standardize the way a program works. If a user knows how to open a document in a word processing package, then if the interface is standardized, they will also know how to open a worksheet in a spreadsheet package.

So in summary, standardizing programs makes it easy for the user to transfer skills from one application to another.

The main features of a GUI

There are a number of features common to all GUIs and these are:

- A mouse is used as the main input device. By moving a mouse on a flat surface, the cursor can be made to move across the screen; the left mouse button is used to make selections.

INSERT GRAPHICS HERE

- Overlapping windows are used. Many windows, even in different applications, can be opened simultaneously. You can therefore have a spreadsheet and a wordprocessed document on the screen at the same time and this makes it much easier if you are going to import data from one package into another.

- They make use of many graphics features. There are many graphical features incorporated into the design, such as icons, pull-down menus, toolbars, slide bars, selection boxes, dialogue boxes, etc.

ACTIVITY 4:

Discuss in small groups the following questions.

a) All systems software and applications software have user interfaces. Some systems software makes use of a command line interface while others use a graphical user interface (GUI). Discuss the main differences between them.

b) Discuss – In your opinion what are the five most important features of a graphical user interface?

c) Find three advantages in using a GUI rather than a command line interface.

Clarity of structure and layout

If a graphical user interface is to be easy to learn and use, there are a few guidelines to bear in mind during its design and these include:

Reduce the mouse movements.

Put items such as icons and menu selections close together if they are likely to be used together.

Use pull-down menus.

The use of pull-down menus means that the screen is not cluttered with items to choose from, so the user has more of the screen available as their working area.

Design pull-down menus

The selections used most frequently should be situated at the top of the menu. This avoids the need to move down through the menu more than is necessary.

Include the facility to select which icons are displayed.

There are usually many more possible functions that have icons than there are icons on the screen and it is possible with most user interfaces to choose a selection that the user is most likely to need. As well as specifying which are shown, the user can usually put the icons in any convenient position on the screen.

The advantages of having a common user interface for different generic application programs

The advantage of most users having the same operating system is that people can move between computers and still know how to operate them. The same can be said of different generic packages, such as word processors, databases and spreadsheets.

Common commands

Where a number of commands can be issued using the keyboard, it makes sense to use the same combination of keys to perform the same task no matter which manufacturer has produced the software. This needs a certain amount of co-operation between rival companies and it can also mean that newer, improved user interfaces are harder to introduce. You have only to look at the standard layout of the typewriter keyboard to see how users like uniformity. Can you imagine what it would be like if every computer manufacturer decided they would have a different arrangement of keys on a keyboard?

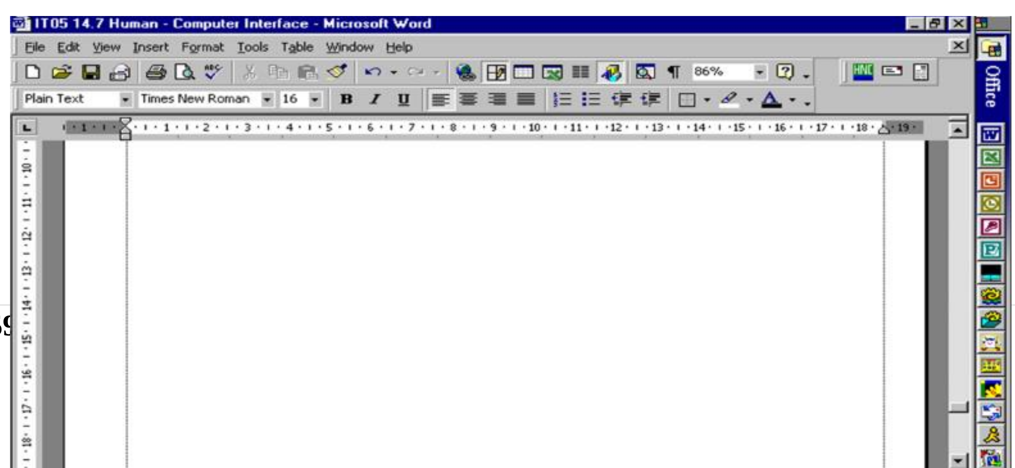
The present arrangement was designed originally to cope with the mechanical properties of the typewriter, but ended up making it slow. Other shapes of keyboard have been developed over the years; they are more ergonomically designed but people's resistance to making the change has meant that there are very few of them in use, even though most users find they are an improvement.

<u>Increased speed of learning</u>

Once a user has been trained on, and has mastered, one package, other packages with similar user interfaces are much easier to understand. The user will understand how to pull down menus and make selections using the mouse and also know what each of the icons means. An icon for print or save will usually look the same no matter which package is being used.

<u>Features of user interfaces that are common to all packages</u>

The more packages you look at, the more you will notice the similarities between them. It is not that the software manufacturers are copying each other, but that they are satisfying user demand by incorporating features common to many packages.



ACTIVITY 5:

Common user interfaces

Discuss some of the different user interfaces in different packages (if possible, from different manufacturers) and you will notice that there are many recurrent features.

Debrief

Interfaces from software created by the same manufacturer are usually quite similar, particularly if the individual software components form part of a software suite. Did you look at the following?

- The arrangement of the overall screen.
- The use of pull-down menus and buttons.
- The design of the icons.
- The use of colours on the screen.
- The order of the menu selections in pull-down menus.
- The operations that need to be performed when files are imported from another package.
- The way in which you change the font size or font type.
- The way in which the printer settings are changed.
- How you can run another task at the same time (i.e. multi-tasking).

Designing systems and interfaces appropriate to disabled users

Interfaces for people with sensory impairments must be aimed at those senses which are unaffected by their disabilities.

A visually impaired person has two problems with most common interfaces. First they cannot look at the screen to see the menus, icons, etc. that enable them to make a particular selection, and secondly, they are not always aware when a mistake has occurred. The usual graphical interface is difficult to use and it is better if commands are issued using the keyboard.

Before the introduction of the Windows operating system, an operating system called DOS (disk operating system) was used, which meant the user had to learn a series of commands to do certain things such as format a disk or copy a group of files. Many of these commands did not need to be typed in full, but could be issued using a single keystroke or a combination of keystrokes. The visually impaired user can issue DOS commands to the computer relatively easily. When hard copy output is needed, it is possible for an impact printer (such as a dot matrix printer) to produce the output on Braille which can be read by other visually impaired users.

Another approach is to use special software that converts the text or commands into speech. For example, wordprocessing software for visually impaired people enables the user to move through a document word by word, while the system reads them. By using this method, the user can detect any mistakes made. The user of this system also hears commands as they are issued, so mistakes can be detected and the command corrected.

Many users, particularly disabled users who have difficulty in pressing individual keys, may find speech recognition systems much easier to use. The main problem arises when what has been typed needs to be edited. There are, however, ways in which the user can direct the cursor for editing and then issue spoken instructions.

The importance to companies of adopting a common user interface

If you regularly work at a college or school, you will probably have to use lots of different computers in the course of a week. This means you are likely to experience some problems when using operating systems software or applications software.

The problems usually involve the previous user having changed some of the settings on the interface. For example, they could have changed the screen or character colours.

If you are using applications software, such as wordprocessing, there may be different margins set or toolbars showing on the screen. These are just some of the frustrations that occur when you have to share computers with other people.

Many commercial organisations do not have a desk for each of their employees; they consider it wasteful to do so because not all employees are working at their desks at the same time. This means that employees have to find a vacant desk with a computer if they want to do some work. So that changes to the user interface are not passed on to the next user of the system, companies make use of a network where the software is stored on the server. When a user logs on to the system they are presented with the same interface, and it is then up to them to change the settings for their own use during their log-in period, should they want to.

This avoids unnecessary calls to the help lines; most large companies operate, to sort out problems caused by changed settings.

Summary of the benefits of providing a common user interface between packages

- The operational basics of one application can easily be applied to other applications.
- Key commands can be found in the same place for each application.
- There is consistency in toolbars and menus.
- Dialogue boxes, customisable features and operational features are similar.
- On-line help is provided in each application in a similar way.

Other types of user interface

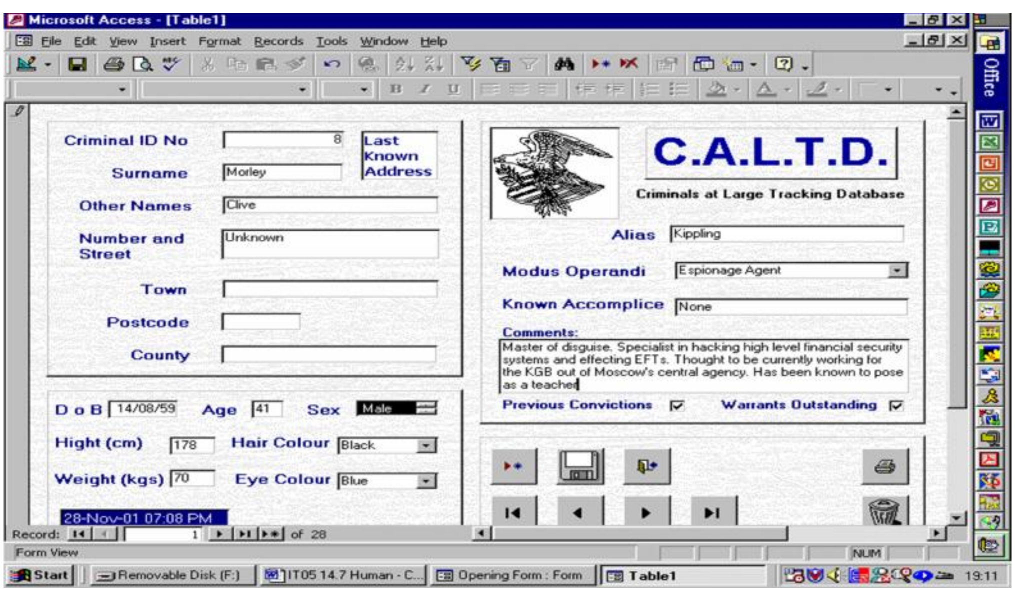
Sometimes, users need only simple interfaces, when the variety of tasks to be performed is quite limited, for instance, entering customer order details using a terminal.

We will no look at some other types of interfaces.

Forms dialogue

Forms on the screen are used in a similar way to paper forms. They enable data to be entered into the system in a pre-determined and structured way. The forms usually have the name of the data to be entered into each box at the side of the box and this tells the user exactly what they have to type in. If this is not sufficient, additional instructions can be added by way of an explanation. Forms can also have buttons added and pick lists that give the user a dropdown list of options to choose from. Check boxes can also be used, where the user can select one or more items by clicking on the appropriate boxes in a list. This arrangement is very popular when entering details into a database; if you do a database project you will probably have to design such a form for data input.

Forms dialogue can be classed as either formatted or free format. The difference between the two is in the flexibility offered to the user. With formatted dialogue interfaces there are fewer ways for the user to enter data into the computer and for this reason they are better suited to novices. With free format dialogue the interface is more complex and there are lots of ways of entering data. It is important that the designer of the interface matches it to the capabilities of the user, since a simple formatted dialogue design can annoy an experienced user with its lack of scope, while a novice could find overwhelming the choices in a free format dialogue system.



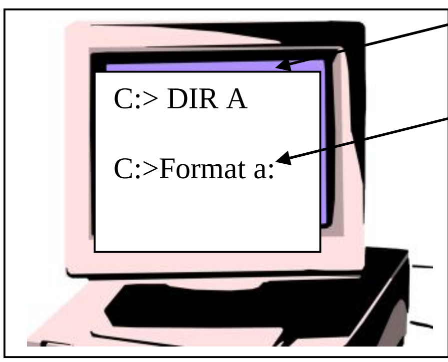
Command-driven interfaces

The main problem for users of command-driven interfaces is that to use them successfully, it is necessary to remember a large number of commands and also how to construct them. Although these interfaces often use help screens, in case you forget a command or need to look up the syntax, they are still very hard for inexperienced users to master. The users of command-driven interfaces have to learn a command language similar in many ways to a specialist programming language and which is almost as difficult to learn.

INSERT DOS COMMANDS HERE

<u>Interface Styles</u>

<u>1. Command Line Interface</u>



Lists what is on a floppy disk

Formats a floppy disk

- The user types in commands. The computer responds by displaying text on the monitor.
- The user must remember all of the commands.
- MS-DOS, some databases, e.g. dBase4 (LDL taught this in 1993!)
- Good for experts - the commands can be entered quickly and take up little computer memory.

<u>TASK 2</u> Explain three advantages and three disadvantages of using a command line interface.

Users usually know what they want to do, but they do not know how to translate this into a series of commands. For this reason command driven interfaces have become much less popular over recent years, and have been overtaken by graphical user interfaces.

Command-driven interfaces are also called command line interfaces because it is necessary for the user to type in a command next to the cursor on the line. Many operating systems such as MS-DOS (Microsoft disk operating system) and UNIX use a command line interface and in MS-DOS the user has to type `C>DIR' to obtain a list of the files stored on the hard drive. Once all the commands are learnt, a command line interface can be quite fast, but you may waste time looking up commands and syntax details or making mistakes and then having to correct them.

Here is a summary outlining the advantages and disadvantages of using a command-driven interface.

Advantages

- They are very powerful and the user can achieve a lot with a single command.
- They are very quick provided you are an experienced user.
- They are very flexible and you can alter the parameters to do different things.

Disadvantages

- They are quite difficult to learn.
- They are less suitable for novice or intermittent users.
- They sometimes use obscure abbreviations or keywords and syntax is important.
- They are prone to typing mistakes.

Graphical User Interface (GUI)

Makes the computer as user-friendly as possible by using graphics, icons and pointers (mouse, trackball etc).

Features of GUIs

Uses the WIMP interface. **W**indows, **I**cons, **M**ouse, **P**ull-down (or **P**op-up) menus.

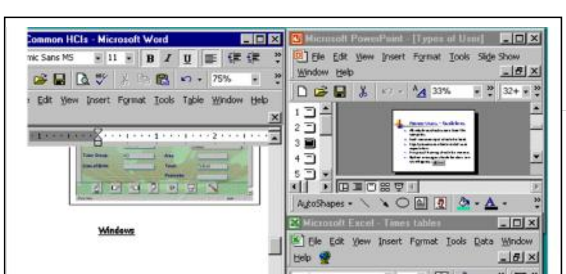
Pull Down Menus

Pull-down menu and sub-menu
A pop-up menu is displayed when you right click the mouse.

TASK 4

1) What do the ticks signify?
2) Why is one row blue?
3) Why have a menu and sub-menu instead of just one menu?

Windows



A window is an area on the screen through which you can view a piece of software or a file.

You can have several windows on the screen at the same time - this shows

WYSIWYG (wizzy-wig) **W**hat **Y**ou **S**ee **I**s **W**hat **You** **G**et

- What you see on screen matches what is printed.
- Used on word processors and DTP packages.
- Used to be different **-** you had to put codes in to denote **B**old, *I*talic, Underline.

TASK 4 List two advantages and two disadvantages of GUIs

Menus

There are several different types of menu that can be used:

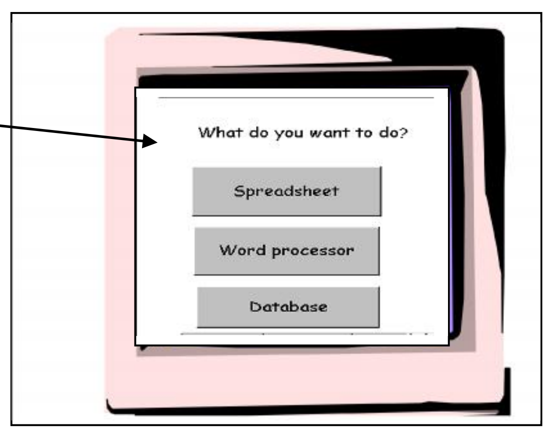
- Full screen menus
- Pop-up menus
- Pull-down menus.

Full screen menus

These are menus that take up the whole screen, which remain in view until the user makes a selection. Many opening menus for application packages are of this type.

Full Screen Menu Interface

- The user selects one option from a small range of choices.
- User may point and click on an option or use a shortcut key.
- Users cannot cope with too much choice, so you may need additional pages of menus.
- The menus and sub-menus should be appropriately grouped.
- Good for inexperienced users because they do not need to remember commands.
- Takes time to go through each menu.

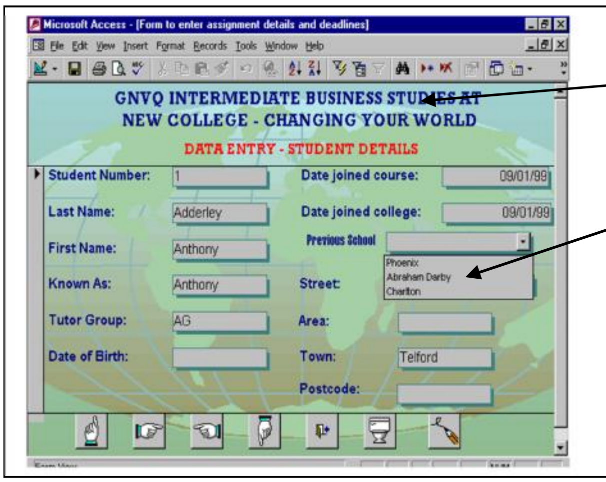


These are usually brought up by clicking the right-hand button of the mouse; the user is then able to make a selection from a list.

Pull-down menus

To save space these menus are only shown if the user clicks on a particular item. To make a selection, the user clicks on one of the items in the menu. To cancel the operation, escape is pressed.

Form based data Entry



The user should be guided through the form as if it were a paper form - top to bottom and left to right.

Entries can be by text or from a list.

Default values can be included.

<u>**TASK 3**</u> Explain two disadvantages and two advantages of using forms for data entry.

<u>**Natural Language Interface**</u>

Humans communicate in natural language, e.g. English.

- *Computers respond to a small number of precise commands and do not understand ambiguous language.*
- *Ask Jeeves is a good example. Users ask Jeeves questions in plain English and receive links to appropriate websites.*
- *Easier to ask for 'A list of student s who need to resit a module than using SQL :*

SELECT [Student test result].[Student Number], Student.[Last Name], Student.[First Name], Student.[Tutor Group], Test.[Test Number], [Student test result].Pass
FROM Test INNER JOIN (Student INNER JOIN [Student test result] ON Student.[Student Number] = [Student test result].[Student Number]) ON Test.[Test Number] = [Student test result].[Test Number]
WHERE ((([Student test result].Pass)=No));

Problems with Language

Language is ambiguous:

- Are Lynn and Mike married?
- Are Lynn and Ian married?

The same word can have different meanings

- Did Shrewsbury Town win any matches this season?

TASK 5 Explain two advantages and two disadvantages of a natural language interface.

Resource Implications

GUIs and Windows are user-friendly, but they need better and faster computers. We could not have such systems running on old computers.

Operating System	**Capabilities**	**Minimum Requirements**
MS-DOS 6 (1981)	Command line interface (with some extras)	Processor 486 (clock speed 10Mhz) RAM 512 Kb Hard drive space 5Mb
Windows 2000	GUI Internet capability	Processor 133Mhz RAM min 64Mb Hard drive space 650Mb
Windows XP	As 2000 + networking, movies, music etc	Processor 300Mhz RAM min 128Mb Hard drive space 1.5 Gb

TASK 6 Explain why the newer software (GUI) needs:

a. Faster processor
b. Greater RAM
c. Greater Hard drive space

than the old command line systems.

We now also look for extra hardware:

- Larger VDUs for the graphics
- Modem for the Internet

- Camera for video conferencing.

The potential for a natural language interface

It would be convenient if we had the same human/computer interface as the computers in science fiction such as Star Trek and 2001, A Space Odyssey, in which people could simply talk to computers in the same way as we talk to another person. Since this is the most natural way of communicating, such an interface is called a `natural language interface'.
`Natural' in this context means human-like and the idea is to get computers to behave more like humans so as to make it easier for us to communicate with them.

The most important aspect of a natural language interface is that the computer should be able to understand what the user wants it to do without requiring correctly structured commands and data names in a particular order. All you should need to do is to express yourself clearly, either by typing or speaking to the computer.

Since people can say the same thing in a variety of ways, the computer would need to be able to interpret and understand what is being said accurately, and this is a major hurdle for developers of natural language interfaces.

One person might say `Can I have the sales of Mars bars for March?', and another might say `Give me the revenue details for Mars bars sold in March'. In either case the computer should be able to give the same details. A natural language interface also needs to be able to cope with misspelled words, bad grammar and slang, either `understanding' them or asking the user for clarification. If speech recognition is used, the interface will also need to cope with mispronounced words, different regional accents, etc.

The use of a natural language interface to access a knowledge base is the foundation of the area called artificial intelligence.

Pointer-based interfaces

There are other up-to-date interfaces besides GUIs and these are called pointer-based or gesture-based interfaces. They work by using a pen-like stylus or pointer to interact with the computer. With some systems, you can simply write in ordinary handwriting using the stylus on a special pad which represents the input device for the system. These systems need special software called handwriting recognition software. In some systems, particularly CAD (computer-aided design) systems, the user uses a graphics tablet or pad and the stylus is used to point to certain shapes or commands on the pad. The advantage with this system is that it allows the user to use freehand, using the stylus like a pencil. This is much easier to control than a mouse.Some pointer-based interfaces do not even have a stylus and they instead make use of a touch sensitive screen which can detect the pressure of a user's finger on the screen. You may have seen such devices being made use of in quiz

machines. You can also see them in banks where they are used to present the user with a series of options and the user can make their selection by pressing the screen at a certain point.

Pointing devices

Pointing devices enable the user to move the cursor to anywhere on the screen and point to a tool, icon, menu selection or button. The commonest pointing device is the mouse which seems to come with any new computer purchased. The mouse is the part of the computer system that wears out soonest because it is usually mechanical and has moving parts which wear. Mice usually come with either two or three selection buttons for making selections and some have a small wheel in the middle.

As well as traditional mice, there are also some other pointing devices which are better suited to some applications.

The trackball

If you turn an ordinary mouse upside down and rotate the ball using your hand, you have a simple trackball. Although it is quite difficult to use, after a while you get used to it. Trackballs are particularly useful if there is no flat surface on which a mouse can be moved and for this reason they are provided on laptop computers. Systems designed for members of the public or for children often make use of trackballs. One such trackball, called EasyBall, is designed specifically for very young children, so that they can learn to interact with the computer from an early age. It is also ideal for anyone who lacks the manual dexterity needed for using an ordinary mouse.

The touch-sensitive pad

There are two types of touch-sensitive pad; the mouse pad and the bit pad. The mouse pad consists of a touch-sensitive pad which works in a similar way to a mouse, in that it senses the relative position of either your finger or a special stylus on the pad. By moving your finger (or the stylus) over the pad, the movement is transferred to the movement of the cursor on the screen. Mouse pads are useful when space is at a premium and for this reason they are popular with laptop computers.

The bit pad works in a slightly different way, by making use of absolute positioning. Unlike the mouse pad, if you move the pointer from one corner of the pad to the other, the cursor jumps from one corner to the other. This makes it quicker to use than the mouse pad since it is not necessary to move the stylus across the surface of the pad. The bit pad is ideal for drawing freehand on the screen since using the stylus on the pad is just like using a pencil or pen on paper.

Full size bit pads are ideal for drawing or painting. The pad also has icons from which certain tools and functions may be selected by touching them.

Speech recognition

Using speech to supply both instructions and data to the computer moves us nearer to the natural language interface. It allows humans to communicate directly with the system and although it can be used in a wide variety of applications, it has only become popular as a method of communicating with the computer via the operating system OS/2, or when inputting text into a word-processing package.

The main advantage with speech recognition as part of the user interface, is that there is no longer any need to learn commands or complex procedures. Speech recognition also avoids any need for difficult-to-use devices such as mice and keyboards.

Using speech recognition, the user has only to state what they want the computer to do. Of course, things are not that simple and there are a number of difficulties that need to be overcome. When speech recognition is used to issue commands, it is necessary for complex speech recognition software to understand what the user means, translate this into an actual machine command and then execute it.

Problems include being able to understand different kinds of voices (e.g. male and female) and different accents. In addition to these problems the system also has to deal with background noise such as telephones ringing or people talking in the background.

You may have already used a speech recognition system in the shape of a telephone service where the computer asks you simple questions and then reacts to the responses given.

Using speech recognition and a natural language interface means that users will no longer need to interact physically with the computer using a keyboard or mouse. Instead, as long as the computer system can hear them, this will be enough. Just think, you could ask the computer to load the word-processing software and create a new document, then dictate your letter while hanging wallpaper! This would also open up lots of possibilities to disabled people who could then interact with the computer more effectively

Speech recognition offers an improved interface for most people, who can generally talk faster than they can type. Also, if you tell the computer in general terms what you want, it takes a lot less time and explaining than it does typing the instruction.

Speech recognition will be particularly useful for the Internet since a user can simply describe what they want the computer to search for and an `intelligent agent' will go away and look for the information.

There are some areas where speech recognition and the natural language interface may not be useful. For example, mathematical equations and programming steps are both difficult to describe using the spoken word. Handwriting recognition or a keyboard are more useful in these cases.

There are occasions, such as when you are preparing a confidential document, when it would be inappropriate to speak aloud, so some silent form of HCI would be needed. Another example is the selection of options from a list, in which case the options would need to be read out. A keyboard or mouse is better suited to this type of application.

<u>Keyboards</u>

When you buy a computer it comes with a keyboard and mouse (or other pointing device in the case or portables and laptops). Coupled with a screen, these provide the most usual interface. However, this arrangement does present a problem for users who only type slowly and even more of a problem for users who have never used a computer before. It is for this reason that other interfaces have been developed for these situations. Even when software makes extensive use of a GUI, you can often still use the keyboard to work the computer, using a combination of keys. For an experienced typist, removing their hands from the keyboard to use the mouse will slow them down considerably. Instead they can use a combination of keys (such as Ctrl and P at the same time).

Dedicated keys

A dedicated key is a key on the computer keyboard which is used for only one purpose. The purpose of the key cannot be altered using the applications software. The page up and page down keys are examples of dedicated keys.

Soft keys

A soft key is a key on the computer keyboard which may be used for different things by different packages. For example, the function keys F1-F12 all carry out different commands depending on which software is being used.

Chapter summary

The human computer interface (usually shortened to HCI) is the term used to describe the interaction between the user and a computer. Many different things could be regarded as part of the HCI, like the way the screen looks, or whatever the program makes it clear to the user what they have to do next. This is where the term **user friendly** originates from.

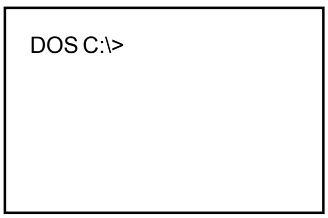
It is important **not** to allow the word “computer” to limit your vision to a PC sitting on an office desk. HCI’s are also found in the following situations: -

- Cash machines at banks / building societies
- A pilot checking his instrument panel on a jumbo jet
- A musician composing a symphony
- A scientist monitoring a chemical reaction

Organizations tend to use one of three types of HCI

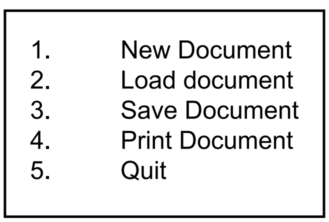
i) Command Driven

Command Driven Interface



a direct method of access for more experience users - requires typing a command to make something happen, but you must already know what that command is. Dos C:\> e.g. DIR gives directory listing- the fastest way to issue commands and explore the computer system. Must type in exactly.

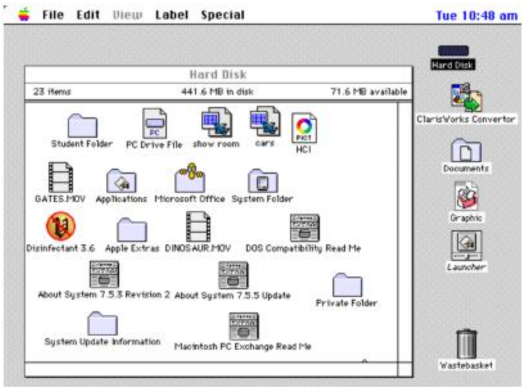
ii) Menu Driven



Menu Driven Interface

Menu driven systems are slightly more user friendly than command driven systems because you are less likely to make mistakes if you do not remember the commands. The user can choose from the menu

ii) Graphical User Interface (GUI) **(See ICT2)**



GUI's (Gooeys) Graphical User Interface - for novices and less technically minded users.

GUI's require far more memory and operate more slowly because of added graphics processing

(WIMP Windows Icons Mouse Pointer and Pull-down menus)

Benefits All possibilities as a list Error trapping is simple *Context sensitive help can be provided*	Minimal typing Inappropriate choices can be withheld from user
Drawbacks Tedious for experienced users Several screens might be required	Extended hierarchy of menus can be difficult to follow

Advantages of common user interface between different generic applications include

common commands ease of use reased range of tasks solvable by experienced users	Increased speed of learning Confidence building in novice users Greater range of software accessible to average users

For users to communicate effectively with IT systems, a good user interface design is essential.

Well-designed systems can improve the output of employees, improve the quality of life and make the world a safer and more enjoyable place to live in.

Designing New Software

Research into Human-Computer Interaction (HCI) involves the study of good software design to see what makes it good. Researchers observe people interacting with computers to see what they find intuitive and what they find confusing.

- Good interfaces provide:

- Help for novice users
- Short-cuts for experienced users
- Metaphors or images (e.g. a picture of a printer on a print button)
- Consistent behaviour, which makes use of long-term memory e.g. always using F1 for Help or ESC to stop a process. There are certain functions that have become de facto standards.
- Clear and helpful error messages.
- Uncluttered screens with effective use of colour.

Properties of a good user interface

The best interfaces are those that are those that are: -

ATTRACTIVE	• Interfaces are more likely to be used if they are attractive. • Screen arrangement i.e. colours, typefaces (font & sizes), graphics elements, all impact on the look of the interface
FORGIVING	• Users should be able to recover easily from mistakes users should be able to get on line help easily • Easily cancel wrong selections • Should be intuitive when things go wrong
CUSTOMISABLE	The user should be able to tweak the interface to his or her own habits e.g. • Ability to size or move windows • Design icons and tool bars • Design desktop patterns
TRANSPARENT	• This implies predictability; that is the interface should provide choices that are logical and reflect how users are likely to react to situations
UNBURDENSOME	Implies that the software rather than the user should be the most accommodating • e.g. menu systems that require users to simply recall what a command does rather than remember a specific language syntax • Several ways to execute a command, thereby making it easier for the user to work in the way they find most comfortable
SAFE	e.g. the pilot of a jumbo jet
EFFICIENT	• Users do not spend five minutes trying to find the correct way to insert their card and type in their PIN and the amount of cash they

	want, and then leave without remembering to take their card
ENJOYABLE	• a primary school pupil performing a certain task within a program • a new user finding his/her way around the system
USER FRIENDLY	• e.g. screen which leaves error message on screen can be confusing • Repeated rejection of data without explanation can be frustrating • Should be concise but intuitive/easy enough to allow the user to correct the error
AVAILABILITY	• Users still complain that programs are hard to use. What these programs need are **of help** better built in training and troubleshooting features e.g. on line help screens that the user can pull up on the screen for assistance when they are stuck • indexed alphabetically • Context sensitive - refers to an on line feature that provides assistance relating to the type of operation the user is currently trying to perform

USABLE should be a product of collaboration between the designer and the users

User, not designer, convenience should be paramount	Interface is consistent throughout the system
Built in help and advice accessible at different levels	Spacing is important
Techniques of highlighting such as blinking and colour should be used sparingly	Desktop
Filing cabinets for disc drives	Documents for files
Folders for directories	Waste paper baskets for deletion of files
Control Panel	Buttons for initiating action e.g. print
Switches for setting options on and off e.g. a grid on a spreadsheet	Radio buttons for choosing sizes of paper
Lights to indicate active events e.g. printing	Sub-panel menu to select system defaults

Resource implications of a sophisticated HCI

- Processing power is required (processing power is needed to draw the interface, leaving less for the application itself)
- The overhead of increased processing time due to complex use of graphics and dynamic objects/windows, etc.
- need for increased memory resource Backing Store (a GUI takes up more disk space than a command line interface)
- Immediate Access Store (a GUI will hog RAM)+ hard disk as virtual memory.
- sound Card
- colour monitor

Examples of specialist HCI's include:-

Automatic pilot systems in an aircraft	Embedded computers- washing machines, microwaves
Navigation systems (land, sea and air)	Machines for developing photographs
Touch screens in tourist information centres	Flight simulators
Speech input (voice recognition)	Automated teller machines

Implications include: -

- Security - ATMs - navigation systems
- Safety - pilot of jumbo jet
- Who is HCI for? blind person, disabled person
- Accuracy (voice recognition)

ACTIVITY 6:

Variety of new terms has been introduced in this module, many of which may be new to you. It is important that you build up vocabulary which can be used when answering exam questions or writing project documentation

Write a definition for each of the following terms used in this section

- Interface
- HCI
- Graphical user interface (GUI)
- Icon
- Natural language interface
- Pointer-based interface
- Trackball
- Bit pad

CHAPTER QUESTIONS

1) A different human/computer interface would be needed for each of the following users:

(i) a young child in a primary school,

(ii) a blind person,

(iii) a graphic artist.

For each user describe and justify an appropriate human/machine interface.

(NEAB, Module ITO2, Specimen Paper, q7)

2) A college uses a range of software packages from different suppliers. Each package has a different user interface. The college is considering changing its software to one supplier and to a common user interface.

(a) Give four advantages of having a common user interface. (4)

(b) Describe four specific features of a user interface which would benefit from being common between packages. (4)

(c) Discuss the issues involved, apart from user interfaces, in the college changing of upgrading software packages. (8)

(NEAB, Module ITO2, May 1997, q8)

3) Speech recognition systems for Personal Computers are now becoming more affordable and useable.

(a) State two advantages to a PC user of a speech recognition system. (2)

(b) Give two different tasks for which a PC user could take advantage of speech recognition. (2)

(c) Speech recognition systems sometime fail to be 100 per cent effective in practice. Give three reasons why this is so. (3)

(NEAB, Module ITO2, May 99, q6)

CHAPTER TEN

HUMAN FACTORS ENGINEERING (HFE)

The Importance and Meaning of Human Factors

Human factors concerns are unique to I-O psychology because they typically involve many other areas of psychology, including in particular physiological and cognitive psychology. As a result, courses in human factors are less common than courses in work motivation, or training, or stress. Nevertheless, to give complete appreciation to the complexity of modern work, and in particular to issues related to work safety, we present this chapter as a primer or perhaps a sampler of human factors issues.

Workers are exposed to a wide variety of work "conditions." We covered many of those conditions in Chapter 10. They included physical conditions such as heat, light, and noise. They also included psychological conditions such as work pace, conflict, and responsibility. We also introduced the notion of compensation for working conditions in our discussion of job evaluation. In each of these discussions, working "conditions" were taken as a given. The individual worker was expected to either adapt, or at least put up with, these conditions.

We will use the term **human factors** to represent synonymous terms such as **human factors engineering or human factors psychology.** Human Factors overlaps with related disciplines such as **ergonomics**, applied experimental psychology, occupational medicine and exercise physiology. We will not deal directly with these latter areas but you should understand that since Human Factors is related to these other disciplines, human factors research and practice may be the most interdisciplinary area of I-O psychology.

The human factors approach assumes that workers are a constant, and that the work needs to adapt to the worker. Human factors engineering can be defined as follows:

> *"[HF] uses knowledge of human [capacities] and limitations to design systems, organizations, jobs, machines, tools, and consumer products for safe, efficient, and comfortable human use" (Helander, 1997, p. 4*).

These capacities and limitations include physical and cognitive abilities, knowledge, personality, and even physiology. The goal of the human factors psychologist is to develop an environment (both physical and psychological) that is optimally compatible with the capacities and limitations of humans. Rather than accepting the environment as a constant, and selecting those few individuals who may be most compatible with it, the human factors psychologist catalogues the capacities and limitations of humans, and develops an environment that is as ideally suited as possible to those humans.

But this was not always the case. In the early days of the factory system, machines were designed by mechanical engineers who had little concern for the capacities and limitations of humans. An example of this can be seen in the design of a popular machine for working on metal parts called a lathe. The purpose of a lathe is to create a shape in a piece of steel or aluminum by spinning that piece at a high speed and applying a sharp bit to its surface while it is spinning. Before computers were introduced to the factory floor, lathes were operated by hand with lots of manual controls, usually in the form of wheels, levers, and buttons which were used to bring the bit into contact with the piece to be shaped.

The person in the top portion othe figure represents a typical lathe operator (height, reach, build, etc.). But the "ideal" operator of a lathe designed by mechanical engineers in the 1920s, as sketched in the lower portion of the figure, would look much different. The "ideal" operator would be slightly over four feet tall and twelve feet across the shoulders, and would have an eight-foot arm span. This "ideal" operator was determined by the way the mechanical engineers designed the lathe, and we can be fairly sure that few real-life lathe operators resembled the ideal. It is most likely that the engineers began their design with standard measurements of the components from which the machine would be built. From the human factors perspective, this was foolish. The mechanical engineers should have *begun their design* with an appreciation of the range of likely characteristics, capacities, and limitations of the operators of the lathe. The implications are clear. If the equipment and environment are not compatible with the humans who will use that equipment and populate that environment, we can expect problems in the form of lowered production, injuries, and accidents. We should also expect to see unhappy workers who are continually "taxed" by their work.

we often take human factors for granted. Imagine an elevator with such a control panel. Imagine the extra time you might need to locate the button corresponding to the floor you wanted to visit. Worse than that, imagine the chaos of a telephone keypad with randomly arranged numbers. There are literally hundreds of devices you use every day that have been designed or modified by human factors specialists – the configuration of an automobile dashboard, the height and tilt of a chair, the keyboard and screen at a computer work station, the arrangement of knobs and burners on your stove top, the positioning of the brakes on your mountain bike, even the way a radio dial or a TV remote control works. All of these are examples of products or objects designed to be compatible with human capacities and limitations. The goal has been to achieve "**user friendliness**." In this chapter, we will consider the concept and discipline of human factors engineering as it applies to work.

Human Factors Models

Howell (1993) identifies human factors as a dynamic force in both technology design and society.In the column on the right is the human factors issue or area of concentration. This figure provides a good overview of the substance of human factors. Two very simple models can be used to position human factors in the broader perspective of the study of work behavior. the worker is embedded in a series of increasingly larger environments which include, respectively, equipment (e.g., computers), physical workspace (e.g., work cubicle or office), social workspace (e.g., teams), and organizational workspace (e.g., climate or culture). Each of these environments has an influence on the performance of the individual. Traditionally, human factors has tended to concentrate on the interface between the worker and the equipment (and in the last several decades, the "equipment of choice" has been the computer).

As you can see, there are several components to this model. There is the worker, the equipment, the way in which the worker receives information from the equipment, and the way in which the worker controls the equipment . Both the equipment and the worker have **input** and **output** components. An everyday example of this model may help you to understand it more clearly. Suppose you had a paper due in one of your classes and you were preparing it on your personal computer. As you sit in front of that computer, you see a screen and a keyboard. What appears on the screen represents output from the computer and input to you. What you type on the keyboard represents output from you, but input to the computer. You and the computer are connected through this information flow loop. You ask the computer to access information in a literature base related to industrial safety by first activating a search engine, then typing in a website, and finally inputting some key words for the search. With that instruction, the computer accomplishes the search and provides you with the relevant journal articles. There have been a string of interactions between you and the computer. You turned it on, you activated its operating system, you started the search engine, you identified the website, and you listed the key words. At each point in this process, the computer asked you to make choices. You made these choices with the mouse or the keyboard, and when you made each choice, the computer went and "did its thing." There was a lot of input and output on both sides of the keyboard.

This simple example introduces two additional technical terms that are important in human factors: **displays** and **controls**. Displays (like the computer screen) provide an individual with information, while controls (like the keyboard or mouse) permit an individual to take actions. There is a rich history as well as a very active current research interest in the design of the most effective methods for display and control (Salvendy, 1996; Wickens, Gordon, & Liu, 1998). But displays and controls are just two components of a more elaborate model of work from the human factors perspective.

Virtually every topic that we have covered in this book is represented in one or more of the blocks of the model. The "machine," the operator, and the environment all form an integrated system and interact to yield productivity on the positive side; and accidents, injuries, and even violence on the negative side.

Human factors is a global discipline. The same HF issues affect virtually any industrialized country. Helander (1997) identified a number of human factors challenges that characterize work in twenty-five different countries. These include the change of work organization and design, work-related **musculoskeletal disorders**, **human-computer interface**, change in social systems of work environments, high technology system design (particularly nuclear power plant control rooms), mental workload, and human reliability.

A Modern Human Factors Challenge: Cell Phones and Driving

The next time you are driving or riding in a car, take a minute and gather some data. Simply observe how many drivers you see who have a cell phone to their ear; you will probably be astonished by the number. You might have thought you were the only one on the road who used a cell phone while commuting. The very fact of your astonishment can help to make our point. While you were driving and using your cell phone, you never *noticed* how many of your fellow drivers were doing the same thing. If you have a little extra time, don't just count them, but watch them drive. They are clearly involved in their conversations. They may look distracted, happy, sad, or angry depending on the substance of the conversation. They are often driving more slowly than others, who may be backed up behind them and annoyed. Drivers behind them may need to beep their horns to prompt them to step on the gas when a light turns green. They may pull into other lanes and into the path of other drivers without signaling. They may breeze through yellow or even red lights without touching the brakes. Since they are often holding the phone to their ear with one hand and controlling their vehicles with the other, or cradling the phone between their shoulder and ear, their maneuvers are often jerky and clumsy. In short, the highways are full of accidents waiting to happen.

Citizens in many areas have mounted a growing backlash against the use of cell phones while driving. Consider the following two examples. An elderly passenger was seriously injured in a collision between the car she was riding in and a car driven by a lumber sales representative who was talking on his cell phone when the collision occurred. Although the sales representative denied he was on the phone, his cell phone records demonstrated that he was seventy-four seconds into a call at the time of the collision. Observers reported that the sales representative accelerated through an intersection and never touched his brakes before the collision. The jury awarded the victim $21 million (Heller, 2002). In another incident, a driver on the Capital Beltway in Washington, D.C. was talking on her cell phone when her car went over a guardrail and into oncoming traffic, resulting in the death of five people (NY Times, 2002).

As a result of incidents like these, as well as scores of "near misses," states and local jurisdictions are passing laws banning the use of hand-held cell phones while driving. A law passed by New York State in 2001 is representative of this new legislation (Perotta, 2002). Drivers can be fined $100 for a first violation, $200 for a second violation, and $500 for additional violations. Interestingly, the law forbids the use of hand-held phones but permits the use of hands-free systems that use speakers or headsets. To be sure, accidents can occur when drivers are fumbling to dial a number or receive a call, or trying to continue a conversation with a phone held to their ear while simultaneously trying to avoid an obstacle, change lanes, turn a corner, or shift gears. But holding the phone is only a part, and possibly only a small part, of the problem. An equally important issue is the attention of the driver. A hands-free systems is not a mind-free system. The critical issue is whether or not the driver is paying attention to the control of the vehicle in the dynamic, often confusing, and dangerous environment of the road – or to the "stimuli" being presented via the cell phone.

A growing body of research confirms the danger of driving and cell phone use. Lesch and Hancock (2004) studied the driving behavior of thirty-six drivers on a test track in Massachusetts. The study was intended to distinguish between the structural problems of cell phone use (i.e., the challenge of holding the phone while driving) and the functional problems (i.e., the cognitive challenge of performing two demanding tasks at once). An automobile was designed with a touch screen on the dashboard to simulate a hands-free cell phone system. The test track was arranged to have several critical demands (e.g., a signal to stop). The experiment looked at the performance of the drivers who were engaged in the use of the cell phone at a "demanding" moment. The drivers who were using the simulated cell phone failed to stop for a red light significantly more frequently than those not using the device. In addition, even when they did notice the red light and stop, they braked much harder and stopped much more abruptly than those not using the cell phone. Abrupt stopping often leads to rear end collisions.

The effect of this cell phone distraction was considerably more pronounced in older drivers than younger ones. Younger drivers were between the ages of twenty-five and thirty-six while older drivers were between fifty-five and sixty-five. We suspect that the effect on drivers younger than twenty-five would be more pronounced, particularly among young drivers who have just begun to drive. These inexperienced drivers must devote greater mental energy to routine driving operations. Since they are inexperienced, every operation requires attention. Thus, the demands of the cell phone impair not only unexpected actions, such as stopping for a light that turns red, but also the routine actions such as maintaining a constant speed, which require more attention for the new driver than the experienced driver. Although the same distracting effects are likely to be seen in any inexperienced driver, regardless of age, it will be most clearly seen in the very young drivers.

The Lesch and Hancock (2004) study is very appealing because it is a field study using a real vehicle on a real road with a variety of participants. Nevertheless, laboratory studies find much the same result. Strayer and Johnston (2001) used a simulated driving task and found that while cell phones were distracting and affected driving performance, neither music nor books-on-tape had the same distracting effect. In a follow-up study, Strayer, Drews, and Crouch (2006) found that "when driving conditions and time on task were controlled for, the impairments associated with using a cell phone while driving can be as profound as those associated with driving while drunk" (p. 381). In the case of cell phones, laboratory research, field research, observation, and personal experience point to the same conclusion: the active use of cell phones and driving don't mix (Beede & Kass, 2006; Horrey & Wickens, 2006). From a human factors standpoint, there is no easy "fix" to the problem. The "hands-free" fix solves the structural part of the problem, but not the functional part. The problem as a whole resides in the way in which humans process information. They pay attention to the most salient information channel when choosing between information sources. For all practical purposes, the selection is automatic – it is not under our conscious control. The only effective way we can choose driving over cell phone conversation is to turn our cell phones off. In this case, the contribution of human factors psychology is to provide a scientific foundation for a shift in public policy, a shift toward limiting the use of cell phones while driving.

Another example of the same phenomenon is the pairing of global navigation systems (GPS) and video screens in cars to provide directions to destinations. The drivers will be required to input information somehow and view or listen to output while simultaneously driving. Consider that you most often look for directions when you are lost and you already have a condition of stress/arousal. The competing demands for cognitive attention presented by the interactive directions device will most certainly degrade performance.

GLOSSARY / KEY TERMS FOR MODULE I

Human factors (human factors engineering or human factors psychology)	Approach that uses knowledge of human capabilities and limitations to design systems, organizations, jobs, machines, tools, and consumer products for safe, efficient, and comfortable human use

Ergonomics	The study of the physical demands of work such as reaching, stretching, lifting, and carrying
User friendliness	Positive characteristic of machines, tools, and consumer products that are designed to be comfortable, easy to use, and compatible with human capacities and limitations
Input component	Component that provides information to a human or computer
Output component	Component that receives information from a human or computer and converts that information to action
Display	Device such as a computer screen that provides an individual with information
Control	Device such as a keyboard or mouse that permits an individual to take actions
Musculoskeletal disorders	Disorders of the lower back and upper extremities (arm/shoulder/wrist). They are the most commonly studied injuries related to workplace safety
Human-computer interface	The interaction between a human and a computer

CONTEMPORARY WORK DESIGN CHALLENGES

Technology

As we have pointed out in the earlier chapters, and as any observer of today's workplace would agree, the nature of work is changing rapidly. This adds to the challenge of designing work and organizations. Although it is impossible to anticipate exactly what changes will occur in technology or process in the next decade, we can look at some current technologies and examine how human factors research has studied and modified these technologies. The two most obvious and pervasive of these technologies are automation and computer use.

Automation

Automation describes a way of completing work through the use of mechanical or electrical devices (Wickens & Hollands, 2000) rather than through direct human action. Robotics is a good example of automation. In an auto assembly plant, robots can be used to weld, to turn car bodies at angles, to distribute materials, or to carry out tests of stability of assembled parts. At one time, all of these activities were performed by humans. From the human factors perspective, automation poses a challenge. Because the human operator invariably interacts with the automated process, the challenge is to design an interface that is effective, safe, and comfortable for the human operator.

One common motivation for automating is to eliminate humans from a system. The term "**human error**" is often used as an "explanation" for a catastrophic accident. In response to this, the logic is that if we can just get the human out of the system, the threat of accidents will be greatly reduced. Examples of completely or partially automated systems include refineries, space ships, engine block plants, nuclear power plants, and even the lowly automatic teller machine (ATM). What is not so obvious, however, is that these automated systems have not really eliminated the person from the system. People are still required to monitor, maintain, and troubleshoot the system, often from remote locations. This is a good news/bad news situation. The good news is that automated systems, on the average, are more reliable than human systems. The bad news is that when automated systems go out control, they can go *wildly* out of control. An oil refinery shuts down automatically, creating dangerous toxic and flammable fumes. An engine block assembly operation drills holes where they shouldn't be for 700 engine blocks processed during a four-hour period. An ATM dispenses $100 bills instead of $10 bills for several hours. While you may wonder why the last example would be a "problem" for you, it is certainly a problem for the bank.

Perhaps one of the most dramatic examples of "failed" automation was the incident at the Three Mile Island nuclear power plant on March 28, 1979. This potential catastrophe was the result of a flawed interaction between an automated control system and human control room operators. As part of a routine maintenance program, the secondary or backup system for cooling the superheated water from the nuclear reactor was shut down. Plant personnel forgot to turn this backup system back on when the maintenance was completed. Two weeks later, a pump in *primary* cooling system failed, and because the secondary cooling system had not been turned back on, the temperature in the reactor chamber began to climb. Then a valve stuck. Simultaneously, an automatic emergency system was activated to address the rising temperature. This automatic system was unrelated to the stuck valve, but as a result of that stuck valve, the action of the emergency system was exaggerated. Finally, the control room operators, confronted with massive amounts of information, some of it inconsistent, misinterpreted what was happening. The situation came close to becoming a genuine disaster in which thousands of people might have died. Of course, we can't blame the accident on one person or one event, but it is clear nevertheless that human decisions played a major role in the accident, in spite of (or possibly even because of) automated systems or subsystems. Similarly, it is not unheard-of for both military and commercial pilots to fly past a coordinate when the aircraft is on autopilot and the pilot is daydreaming or even dozing.

On the positive side, it is clear that automation has taken a great deal of the drudgery, inhumanity, and danger out of work, thus representing a tremendous contribution to organizations, workers, and society in general. As Wickens and Hollands (2000) suggest, automation can play several valuable roles by 1) carrying out functions that humans can't (e.g., using robots to handle dangerous materials), 2) carrying out functions that humans can do but do poorly (e.g., warning pilots when they are too close to the ground), 3) assisting humans in areas where they have limitations (e.g., using a voice recognition system to allow a pilot to input completed elements of a pre-flight checklist and alert the pilot if he or she forgets a step), and 4) freeing humans to do more satisfying and valuable work (e.g., using robots to spot weld auto bodies). Nevertheless, automation has not taken the "person out of the system." It has simply changed the role of the person in the system. Human factors addresses this interaction between worker and machine or process, particularly in terms of costs or problems. Wickens and Hollands (2000) identified some of these costs.

1) **Complexity**: automated systems typically replace the functions of many different individuals and, as a result, represent a "job" that is much more complex than the work done by any one individual. This increases the potential for system failure. The greater the number of components to the automated system, the greater the opportunity for a failure of the system due to a failure of one of the components. In addition, these new and complex systems "do" the work in a very different way from what the human operator might have done. As a result, individuals may have difficulty in understanding the system (or even distrust in the automated system) because it does things differently.

2) **Reliability:** operators may either under trust the reliability of an automated system (e.g., because they don't understand it) or over trust its reliability. In an

under trust situation, the operator may ignore an alarm from the system, or possibly even turn off the alarm component. In an overtrust situation, the operator depends on the system to provide more information than it is designed to provide. As an example, the operator ignores various information about the automated system until or unless a signal of some kind (e.g., an audible alarm) indicates a malfunction. But if the operator has not been monitoring various sources of information all along, by the time an alarm sounds, there may be too much complex information for the operator to process in a short period of time.

3) **Communications:** although automated systems are not human, they still need to communicate with human monitors, operators, or users. Consider the following example of a "dialogue" between a customer calling with a question about a telephone service and a phone menu. The customer heard from a friend that the phone company would provide one-touch local weather updates and wants to get one.

Case study

Automated System (AS): Please listen carefully to the following options

- *to report a problem with your service, press 1*
- *to question a recent bill, press 2*
- *to terminate service, press 3*
- *to apply for a job with the company, press 4*
- *to inquire about new service, press* 5

The Customer presses 5

AS: please choose from the following options

- *to add a phone line, press 1*
- *to change a phone number, press 2*
- *to convert a number to "unlisted," press 3*
- *to convert a number from unlisted to listed, press 4*
- *to transfer a number from one location to another, press 5*
- *to repeat these choices, press 6*
- *for all other questions about service, press 7*

The Customer is confused and can't remember all the choices, so presses 6. After listening to the choices again, the customer presses 7.

AS: please choose form the following options

- to add a phone line, press 1

- to change a phone number, press 2
- …
- …
- for all other questions about service, press 7.

This is an unusual of a communications problem between a human and an automated system. What the customer might not have known, but might have discovered if he or she chose none of the options, and simply waited, was that a service representative would come on the line. Instead, the customer hangs up, looks up another number, and calls it in hopes of being able to talk with a real person. (The chances are good that the customer will instead get a different, even less helpful, phone menu.) The automated system has failed because of a communications flaw. In designing automated systems, it is critical to consider the nature of the "conversation" that will transpire between a user and the system.

Wickens and Hollands (2000) suggest five principles for making sure that automated systems are compatible with human operators.

1) Make sure that the displays and controls are effective.
2) Keep the human operator informed about the nature of the automated process and its status.
3) Make sure that the operator is trained in the process.
4) Make the transition to automation "gracefully" by introducing the automation as an aid to human performance, not a replacement.
5) Make the automation flexible so that an operator can decide when to use it, or to decide how much or what level of automation to use.

Computers

Computers are so pervasive in our culture that a general discussion of the design of computer systems at the workplace would fill a book, not a mere section of a chapter. Even the word "computer" can be misleading. As examples, electric musical keyboards, drum sets, and guitars are computers, although we don't apply that term; we don't say we are going to the garage to play the computer. In contrast, a personal computer is neither personal, nor do we use it to compute much anymore (Craiger & Weiss, 1998). In this discussion, we will limit ourselves to a central aspect of computer use in the workplace -- some characteristics of human-computer interaction (HCI).

Case study

An example of human-computer interaction

Buchanan and Boddy (1983) describe the introduction of a computer into the process of cookie making. The computer standardized the mixing of dough as well as the monitoring of the weight and size of cookies after baking. Both of these operations are critical for successful mass production of cookies. Before the computer was introduced, dough was mixed by master bakers who directed a group of subordinates. Like a chef, the master baker approached the job almost as an art, adding a pinch here, a dash of water there, and personally directing the mixing of every batch of dough. The master baker was universally admired by others in the plant because of the high degree of skill and experience required. After computerization, the job was greatly simplified. It only took one person. This person needed no special skill other than to respond to the computer when it requested ingredients. After taking a sample of the completed dough, this person simply moved the dough on to the person who operated the oven, called the "ovens man."

The effect of computerization on the ovens man was very different. Prior to the introduction of the computer, the ovens man had to check the size, weight, and thickness of a sample of cookies in order to tell the wrapping department whether they needed to make any adjustments in their equipment. The ovens man would also direct the cookie makers to adjust their rollers if the cookies were too thick or too thin, but only after hundreds and sometimes thousands of cookies had already been made. Because this was all done by hand, it was very inefficient. When the computerized weighing system was introduced, the ovensman could get information much more quickly and, as a result, instruct the cookie makers to adjust their rollers much sooner. The cookie wrappers also got much more useful information, and in a more timely manner. As a result of the computerization, the ovensman's position came to be viewed as a much more responsible one, one that was the kingpin to the entire quality control process. From this example, we can see that there is no general statement that can be made about the effect of computerization on any particular job, job family, or sector of the economy. Computerization may make some jobs more important and others less so. In addition, the stature of some jobs may increase while the stature of others decreases, leading to corresponding changes in employee status, satisfaction, and motivation.

User-Centered Design

Human-computer interaction research has evolved into a **user-centered** method of system development that is heavily influenced by applied psychology (Carroll, 1997). In the 1970s, there was little concern or appreciation for the preferences, capabilities, or limitations of the computer user. Remember the idealized lathe operator we examined in Figure 16-1. The same would have been true if we sketched the idealized computer user in the 1970s. This user would not have resembled any human being we had ever seen.The keyboard would have been flat, the screen would have had glare and colors that made reading tiring. Various disk drives would have been tucked away under desks. Screens and keyboards would not have been adjustable for height. The development of modern computer systems is dominated by issues of usability, not technology. If a system cannot be used effectively by the intended end-user group, it has a bad design.

User-centered design is accomplished through **usability engineering** (Carroll, 1997; Craiger, 2000). This process is iterative in that a basic system is designed and then redesigned with input from users. Carroll (1997) gives the example of word-processing system in which actual secretaries would be involved in both the initial and re-design teams. This has been referred to as a **participatory design**, and it is reminiscent of the behaviorally anchored scale development (BARS) for performance evaluation and management. In addition, the goal of the design would be stated in user-directed terms, such as "two thirds of the users will be able to prepare a two-page business letter in less than ten minutes with fewer that three errors after thirty minutes of training" (Carroll, 1997, p. 69). The design and pilot testing of the system would also include procedures like cognitive task analysis and think-aloud protocols, much as we discussed in Chapter 4 in the section on job analysis. The simplest way to contrast the "old" method of system design with the modern approach is to say that design came out of the laboratory and into the field. There is even a branch of this design method known as the **ethnographically-informed design** (Bentley et al., 1992) that takes into account power relationships, tacit knowledge of the organization and its procedures, and organizational climate and culture. The research of Buchanan and Boddy (1983) involving the production of cookies anticipated this move toward a greater recognition of the importance of social-organizational issues in systems design.

In summary, the new model of systems development is user oriented and heavily dependent on users as subject matter experts. The concentration on usability signals a shift from a technological focus to a strategic focus. It is equally, if not more, important to the organization that the end user of the system feel comfortable with a system than that the system employs the full potential of available technology. If this comfort level is missing, the most sophisticated and powerful system is nothing but a method to collect dust. It is tempting for engineers to construct a technical masterpiece to impress other engineers, but the user is often not impressed. The moral: just because you *can* do something from a technical standpoint doesn't mean that you *should.*

It is the job of the I-O psychologist to address these issues of usability using the same tools that he or she would use to develop any system intended to enhance human performance. These tools are found in the chapters on performance measurement, training, motivation, leadership, and organizational design. They include methods like cognitive job analysis (e.g., think-aloud protocols), critical incidents interviewing, and statistical analysis of the relative effectiveness of different designs.

Approaches to Work Design and Redesign

We have examined a number of different work design issues. These have included technological variables (computers and automation) and social variables (work scheduling). Campion and Thayer (1985; 1987; Campion, 1988; 1989) have proposed that one might take many different approaches to designing or redesigning work, and that since each approach has different goals, we might expect different outcomes. Campion and Thayer (1985) examined 700 different "rules" that have been suggested for designing work and reduced them to four different approaches or models, which are presented in Table 16.2. As you can see, each approach is based on a different theoretical approach, has a different goal, and seeks to affect a different outcome. In Table 16.3, you will see the specific questions that might be asked in designing or redesigning a job. If we use this framework to reconsider the design issues we have discussed thus far in the chapter, you can see that any given design change includes several different approaches, not just one. Our example of the introduction of the computer into cookie making, as well as the process of usability engineering, included elements of the **motivational**, **mechanistic**, and **perceptual-motor** approaches. Automation included elements of the mechanistic, biological, and perceptual motor approaches. Work scheduling is related to the motivational and **biological** approaches. When we discuss accidents and accident reduction techniques in a later section of this chapter, you will recognize elements of all four approaches.

Campion's (1988; 1989) models have several implications. The first is that in the design or redesign of work, we need to be clear about what outcomes we expect or desire. If we are redesigning work to increase worker satisfaction and to reduce turnover, we may want to choose the motivational approach. If, instead, we are trying to reduce injuries and increase the physical comfort of the workers, we would rely on the biological model. To reduce errors or accidents, we would choose the perceptual motor approach. A second implication is that conflict may occur between the approaches, resulting in both anticipated and unanticipated outcomes. So, if you were to use the mechanistic approach to increase productive efficiency, you would simplify work. But by doing that, you would also make the work less interesting and motivating for the worker.

As an example, consider the job of a receptionist. Initially, he or she may have diverse duties such as answering the telephone, greeting visitors, maintaining office supplies, and scheduling various in-house conference rooms. Assume that, when the receptionist was busy attending to other duties, callers occasionally got a voice menu instead of the live receptionist. Assume further that the CEO, feeling that voice menus were too impersonal, suggested that all other duties except answering the phone and greeting visitors be eliminated from the receptionist's duties. Although it might be true that the instances of a caller encountering a voice mail menu would be reduced, it is also likely that the interest value of the job of receptionist for the incumbent would be degraded.

Thus, work design and redesign are complex undertakings; organizations need to be aware of the anticipated outcomes – and on the lookout for unplanned outcomes – resulting from a design change. Campion's work design models would be a good architecture to use for planning design changes. Campion's model, like any model or theory, is not necessarily a statement of "truth." Theories and models are not true or false, they are useful or useless. Campion's theory of design approaches is a useful one, one that has been replicated several times as well (Campion, 1989; Edwards, Scully, & Breck, 2000).

Work Design and Disabilities

As you will recall from chapters 6 and 11, the **Americans with Disabilities Act** protects the rights of workers with covered disabilities. In addition to addressing hiring issues, the ADA also covers aspects of work design and redesign. Caplan (1992) has defined a disability as "an inability to accommodate the world as it is currently designed" (p. 88). If a disabled worker can perform an essential function with an **accommodation** such as a job design change, and the accommodation is reasonable and feasible, then the employer is required to make that accommodation. Some accommodations are relatively simple, such as enhanced lighting for workers with a visual impairment, or ramps for workers in wheelchairs. Other accommodations are more challenging. Noe et al. (2000) present examples of the types of accommodations that might be implemented:

- Eliminating marginal tasks that pose challenges for the disabled worker, or shifting them to other workers
- Redesigning work procedures
- Altering work schedules
- Reassigning a disabled worker to a job with essential functions that he or she can perform
- Providing technology or support in the form of readers or interpreters for employees with reading or visual disabilities
- Allowing an employee to bring a guide dog to work.

Noe et al. (2000) describe the innovative design efforts of a company that specializes in training the disabled (Ricklefs, 1997). One of the problems they addressed is the shaking that workers with cerebral palsy have in their hands; this makes it difficult for them to use a computer keyboard because they often strike keys inadvertently. The accommodation that the company introduced was a clear plastic shield over the keyboard which requires the user to put a finger through the hole above the particular key chosen. Similarly, for a worker with muscular dystrophy who would have difficulty moving his or her arms, the company designed a compact keyboard so that the worker could strike every key without any arm movement.

Vanderheiden (1997) has provided a detailed treatment of the design and redesign approaches for accommodating various disabilities. He suggests three basic approaches: 1) change the individual (e.g., teach them "tricks" or "secrets" for doing things more easily), 2) provide the individual with tools (e.g., telecommunication devices for the hearing impaired), or 3) change the way the work is designed (e.g., rearrange the essential functions of jobs). The most common forms of accommodation involve what are known as "assistive technologies." Examples of these technologies are presented in Table 16.4. You will recall that in an earlier section of this chapter, we introduced the terms "display" (output to the worker) and "control" (input from the worker). Vanderheiden suggests following a "maximization" principle in designing work environments for the disabled. The designer should maximize the number of people who can receive output, and maximize the number of people who can provide input. Remember the figure that showed the "idealized" lathe operator? Old time lathes were not designed for maximizing input (except for those workers who were twelve feet across the shoulders!). Examples of how the maximization principle would be applied appear in Table 16.5 It is important to note that by following maximization principles , designers may make the work better suited not only to disabled workers, but to the non-disabled as well.

A Cross-Cultural Issue in Disability and Design

Integrating disabled workers into the workforce raises interesting cross-cultural issues. You will recall from Hofstede's model of culture that countries vary on the individualism-collectivism dimension. The United States is seen as predominantly individualistic, while China would be characterized as collectivist. Since individuals in collectivist cultures are more concerned about what others think, there is a tendency to "hide" disabled family members to avoid shame and feelings of guilt (Aycan & Kanungo, 2001). In the U.S., individuals with disabilities are not hidden or ignored, and are more likely to be present in the workplace. For multinational organizations, it is important to examine the extent to which the spirit and the letter of the "accommodations" requirement is met in non-U.S. facilities.

GLOSSARY / KEY TERMS FOR MODULE II

Automation	A method of completing work through the use of mechanical or electrical devices rather than through direct human action
Human error	Often used as an explanation for a catastrophic accident, which results in the view that if humans can be taken out of the system, the threat of accidents will be greatly reduced
Complexity	Property of automated systems that typically replace the functions of many different individuals and, as a result, represent a "job" that is much more complicated than the work done by any one individual

User-centered design	Approach to human-computer interaction research that focuses on the user during system development
Usability engineering	Approach that involves an iterative process in which a basic system is designed and then redesigned with input from users
Participatory design	Design adopted in usability engineering that is stated in user-directed terms
Ethnographically-informed design	Type of **user-centered design** that takes into account power relationships, tacit knowledge of the organization and its procedures, and organizational climate and culture
Shift work	The scheduling of work into temporal shifts. Common in particular occupational groups such as nurses, blue collar workers, and public safety personnel
Flex-time	Schedule in which individual workers are given discretion over the time they report to work and the time they leave work on a given day
Compressed workweek	Schedule that permits an employee to work for longer than eight hours per day and fewer than five days per week. Most common is the 4-10 plan, which permits the worker to accumulate the 40 hours of the workweek in 4 days
Circadian cycle	The 24-hour physiological cycle in which humans tend to be active during hours of light and inactive (e.g., sleeping or resting) during hours of darkness
Fixed shift	Workers are permanently assigned to a particular shift
Rotating shift	Workers are moved from shift to shift over a certain period of time
Motivational approach	Approach to work design and redesign that is used to increase worker satisfaction and reduce turnover through modification of motivational levels
Mechanistic approach	Approach to work design and redesign that is used to increase productive efficiency through the modification of tasks or equipment
Perceptual-motor approach	Approach to work design and redesign that is used to reduce errors or accidents through knowledge of perceptual motor skills and abilities
Biological	Approach to work design and redesign that is used to reduce injuries

approach	and increase the physical comfort of the workers through the reduction of fatigue and discomfort
Americans with Disabilities Act	A Federal civil rights law designed to prevent discrimination and enable individuals with disabilities to participate fully in all aspects of society including in the workplace. Applies to a person who has a physical or mental impairment that substantially limits one or more major life activities (like sitting, standing, or sleeping)
Accommodation	Adjustments or modifications to the work environment provided by an employer to enable people with disabilities to have equal employment opportunities. A reasonable accommodation must be provided if a person with a disability needs one in order to apply for a job, perform a job, or enjoy benefits equal to those offered to other employees

CHAPTER ELEVEN

MEMORY AND MENTAL MODELS

MEMORY

In psychology, researchers talk about people having long-term memory and something called working memory. Long-term memory is as having unlimited capacity and we can old information in our long-term memory for long periods of time (although this ability often seems to desert us when it comes to sitting exams). On the other hand, working memory (or short-term memory as it used to be referred) has a limited capacity and we can only hold information in our working memory for a short period of time before it 'decays'. The concept of working memory was developed by Baddeleyand Hitch (1974) and Hitch and Baddeley (1976). They argued that the passive phrase 'short-term memory' should be replaced by the term 'working memory' as this aspect of memory is concerned with the active processing and temporary storage of visual and verbal information.

What affects could working memory have on an individual's ability to successfully use a mobile device or application? Well, when it comes to mobile applications such as speech-based mobile phone services, one can see the importance of taking into consideration the limitations of an individual's working memory. For example, when you are interacting with a speech-based mobile phone service (such as a cinema listings service) you have a single channel of serial information that cannot be scanned or browsed. As a user, you must try to remember the service structure, the menu options, and their location within the service hierarchy.

This serial presentation of auditory information has been found to place great demands on working memory (Tun and Wingfield, 1997), especially in systems that have many menu options and levels. This point relates to what I said earlier in the section on spatial ability of the need for mobile application developers to reduce the complexity of the information they present to users in order to prevent problems of working memory overload and people feeling 'lost' whilst they are using the service or application.

The main elements of human memory are

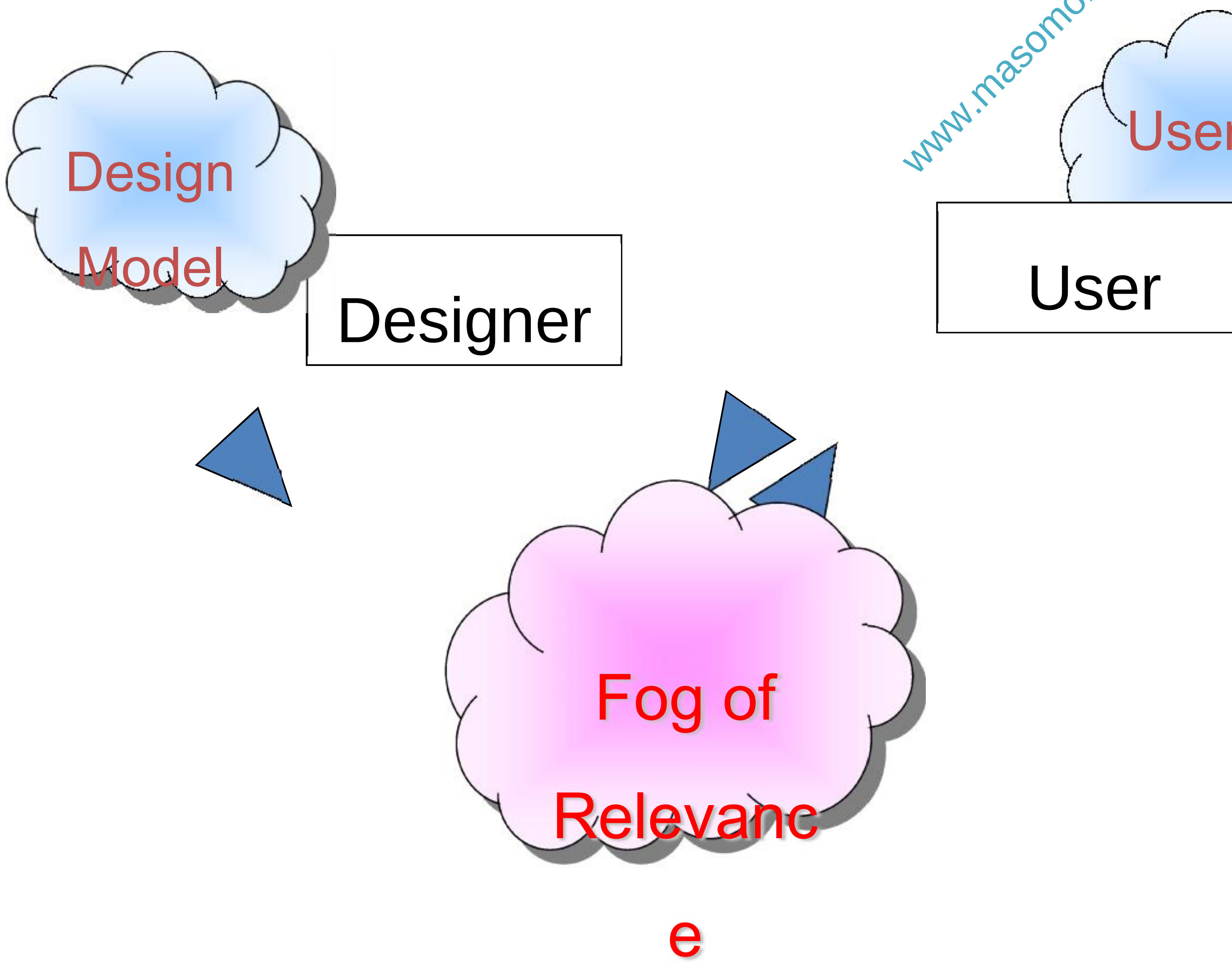
1. Long term memory
2. Working memory
3. Sensory memory

There are three interacting subsystems

- Perceptual
- Motor
- Cognitive

Sometimes subsystems can be categorized as serial, sometimes parallel where serial is in action & parallel in recognition

- pressing key in response to light
- driving, reading signs, & hearing at once



Memory (WM)

Working memory is a short term memory with small capacity and chunked into many parts (Miller 1956) this means that

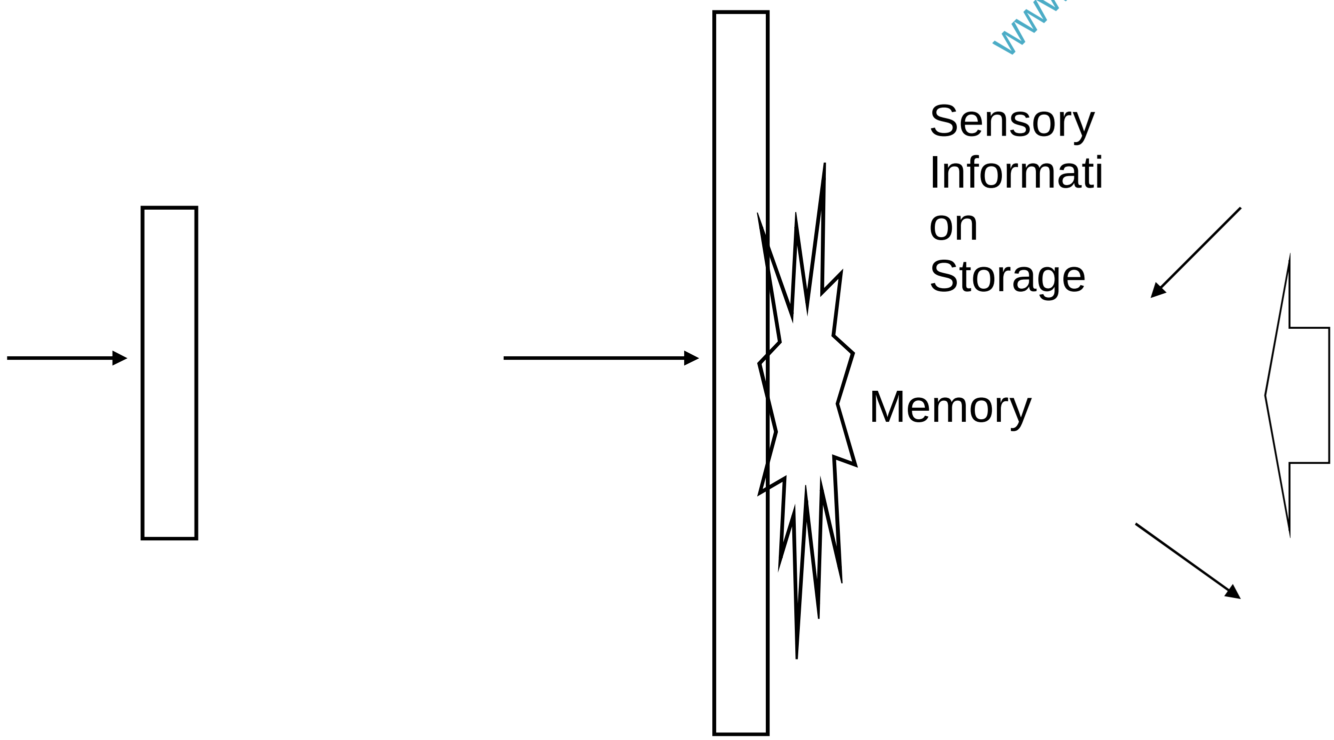
- Locus of attention - *never* your system
- Reduce the need for users to "carry" more than 7 things at once
- Don't require them to remember something the system can provide

Memory (LTM)

- Its is a Long-term memory characterized by
    - Huge (perhaps "unlimited")
    - Slower access time but little decay

- Semantic, episodic, associative

Simplified model of human memory (ii)



Memory in use: in and out of LTM

- Recognize-Act Cycle: “associativity”
  - on each cycle contents in WM initiate actions associatively linked to them in LTM
  - actions modify the contents of WM
  - can’t control what it associates *with*
- Discrimination Principle: “cues”
  - retrieval is determined by candidates that exist in memory relative to retrieval cues (list 2)
  - interference by strongly activated chunks (“reading” colours rather than colour names)
- Memory in use: not just cognitive
- Power Law of Practice
  - task time on the nth trial follows a power law

- $T_n = T_1 n^{-a}$, where a = .4
- i.e., you get faster the more times you do it
- applies to skilled behavior (sensory & motor)
- does not apply to knowledge acquisition or quality

- This means that things (especially motor) become "automatic", even when you don't want them to.

- Users can – and will – assign a meaning or an interpretation to a wide variety of things, regardless of whether that meaning was intended by the designers

CASE STUDY

Problem: freezer too cold, but fresh food just right and *Don Norman's* Fridge: Controls are:-

1. Set both controls

2. Allow 24 hours to stabilize

How can you fix the problem?

- Creating Mental Models
- "Fog of Relevance" Example
- No way of telling what's important
- No way of accessing LTM
- Relies on how information is organised and integrated: on shared "mental models"
- Once the model is shared, all is well
- Where do Mental Models come from?
- One approach: schema theory.
- Memory takes the form of "schema" which provide a mental framework for understanding and remembering information.
- New information is meaningful to the extent that it can be related (attached, anchored) to what is already known.
- Suppose you overheard the following conversation …

When we encounter something repeatedly, such as eating in a restaurant, we develop an abstracted, generic set of expectations about what we will encounter in a "restaurant".

So, if someone tells you a story about eating in a restaurant, they don't have to provide all of the details about being seated, giving their order to the waiter, leaving a tip at the end, etc., because your schema for the "restaurant experience" can (does) fill in these missing details.

1. Schema modification
2. Accretion
3. Tuning
4. Restructuring
5. Schema modification: Accretion
6. New information is remembered in the context of an existing schema, without altering that schema.

For example, suppose I go to a bookstore, and everything I experience there is consistent with my expectations for a "bookstore experience". I can remember the details of my visit, but since they match my existing schema, they don't really alter that schema in any significant way.

1. Goal of the <spit> intuitive </spit> interface?
2. Schema modification: Tuning
3. New information or experience cannot be fully accommodated under an existing schema, so the schema evolves to become more consistent with experience.

- For example, when I first encountered a bookstore with a coffee bar, I had to modify my bookstore schema to accommodate this experience. (And think about some of the examples from Assignment One – coffee bar in library, "bag-it" method of ordering sandwiches).

When new information cannot be accommodated merely by tuning an existing schema, it results in the creation of new schema. For example, my experience with online bookstores may be so different from my experience with conventional ones (cataloguing is different, browsing is different, they don't just sell books) that I am forced to create a new schema.

Features of schemata

1. Schemata are composed of generic or abstract knowledge; used to guide encoding, organization, and retrieval of information.
2. Schemata reflect prototypical properties of experiences encountered by an individual, integrated over many instances.
3. A schema is often formed and used without the individual's conscious awareness. (So as designers, we more-or-less unintentionally define the users conceptual model as well as the interface we intentionally design).
4. Although schemata are assumed to reflect an individual's experience, they are also assumed to be shared across individuals. (There are cultural implications here).

References

- *The Design of Everyday Things*, Don Norman, MIT Press 1998
- *Expert Judgement on Markers to Deter Inadvertent Human Intrusion into the Waste Isolation Pilot Plant,* Sandia National Laboratories report SAND92-1382 / UC-721, p. F-49, November 1993 (pdf available from course webpage)

DEVELOPMENT AND DESIGN METHODOLOGIES

There are many different elicitation and usability/accessibility evaluation techniques (Cooke, 1[i]994) and selecting the "right" technique in a particular situation is not trivial.

Burge's Table of Knowledge Elicitation Methods (Burge, 2001) provides an extensive comparative View of almost all the common KE techniques found in HCI.

In addition, usability and accessibility evaluation techniques are often grouped into two broad areas:

- User-based (that often include user testing) and expert based (that often include heuristic evaluation and cognitive walkthrough) techniques.
- King et al. (2004) presented what they called "An incremental usability and accessibility evaluation Framework for digital libraries". Their framework is broken down into seven key activities and addresses all stages of a design of a system.

Activity 1: Conduct Query **-** Requirement Gathering

Identify satisfaction levels of current users of the system and establish key positive and negative aspects of the interface, what features they would like to see, and so forth.

Activity 2: Analysis

Evaluate current findings and identify issues not yet addressed

Activity 3: Perform Empirical (user) Evaluations

We regard user testing as the strongest evaluation technique, allowing us to identify real user problems by observing users interacting with the system. Retrospective focus groups or interviews conducted after the evaluations also provide a volume of qualitative data.

Activity 4: Analysis Establish key problems and assess if any areas of the service have not been covered by user evaluations

Activity 5: Expert Evaluations

Appropriate modification of expert evaluation techniques maybe required so that they supplement previous evaluation findings, and address any areas or issues that have not as yet been covered

Activity 6: Analysis

Analyze all data identifying key issues that need to be addressed in the redesign of the service.

Establish new usability and accessibility goals for the design

Activity 7: Iterative Process

Re-conduct all stages in the iterative framework to evaluate redesign

We now describe some of the key methods that are associated with the above mentioned framework.

USABILITY PARADIGMS AND PRINCIPLES,

Usability Testing Software

Due to the escalating importance of HCI in software and Web design, some tools have been developed to assist the process of usability testing. Traditionally, usability testing involves the use of video cameras, screen capture tools, microphones, and so forth. One of the main challenges lies in the integration of the media in various formats in a coherent way. Together with cheap computer hardware, usability testing software overcomes this problem by providing a coherent solution, in which Webcam, microphone and screen capture software operate concurrently under one system managed by the usability software. The output is seamless presentation of all media file which can be edited and annotated. Typically, usability testing tools (hardware and software) consists of the following functionalities:

User experience recording:

The most common activity in usability testing is probably recording user performing tasks with the software or Web site under testing. This includes video recording (of the user), screen capture and voice recording (verbal protocol). With the usability testing tools, we can set the tasks needed to be performed, record the performance and administer questionnaires in an integrated environment.

Observation and logging:

Some usability tools support the connected of network computers thus allowing multiple observers to monitor the users and annotate the videos collaboratively. Furthermore, most tools can also log the mouse click and keyboard pressing.

Analysis

Although still limited to basic visualization, usability tools can support data analysis by calculating aggregate usability metrics for users or for tasks. For instance, we are able to quickly identify tasks with low usability metrics and thus focus on improving the design relevant to the tasks.This reduces the amount of work and time significantly and it makes usability testing less costly.

UTILIZATION AND APPLICATION OF HUMAN COMPUTER INTERACTION

PRACTICAL APPLICATION OF HCI.

1. Computer–Augmented Environments

One application area in which HCI plays an important role is the computer-augmented environments, or commonly known as augmented reality or mixed reality. It refers to the combination of real world and computer-generated data visualization. In other words it is an environment which consists of both real world and virtual reality. For instance, a surgeon might be wearing goggles with computer generated medical data projected on it. The goggles are said to augment the information the surgeon can see in the real world through computer visualization. Therefore, it is not difficult to see the connection of augmented reality with ubiquitous computing and wearable computers. Since its inception, augmented reality has had an impact on various application domains. The most common use is probably the support of complex tasks in which users need to perform a series of complicated actions while having access to large amount of information at the same time, such as surgery, assembly and navigation. Apart from these, augmented reality is also used for learning and training, such as flight and driving simulations.

Augmented reality implementation usually requires additional devices for input and output in order to integrate computer generated data into real world: A Cave Automatic Virtual Environment multi-user, room-sized, high-resolution, 3D video and audio immersive environment in which the virtual reality environment is projected onto the walls. The user wearing a location sensor can move within the display boundaries, and the image will move with and surrounds the user.

A head-up display (HUD) is transparent display that presents data without obstructing the user's view. It is usually implemented on vehicles in which important information is projected directly in the driver's viewing field. Thus the user does not need to shift attention between what is going on in the real world and the instrumental panel.

A head-mounted display is a display device, worn on the head or as part of a helmet that has a small display optic in front of one or both eyes. Some of these devices have become commercially available and increasingly affordable. The challenge of HCI lies in the design of information visualisation which is not obtrusive to the users' tasks.

2. Computer–Based Learning

A lot of effort has been put in coupling learning and technology to design effective and enjoyable learning.

Various areas, namely e-learning, computer-based learning, serious games, etc have emerged, hoping to utilize the interactive power of computers to enhance teaching and learning experience. A myriad of design strategies have been proposed, implemented and evaluated, these include the early use of computer in presentation, drill and practice (the behaviourist paradigm), tutorials (cognitive paradigm), games, storytelling, simulations (constructivist paradigm), and so forth. As we progress from behaviourist to constructivist, we notice an explosion of user interface complexity. For instance, drill and practice programs usually consist on a couple of buttons (next, previous buttons, buttons for multiple choice, etc) while simulations could involve sophisticated visualization (outputs) and various user interface elements for manipulating parameters (input). Recently computer-based learning has moved from single user offline environments to online network spaces in which a massive number of users can interact with each other and form a virtual learner community. This social constructivist learning paradigm requires not only traditional usability treatment, but also sociability design in which the system includes not only the learning tools, but other sociability elements such as rules and division of labours.

3. Information Visualization

Information visualization is an area in HCI which can be related to many other areas such as augmented reality just described before. Most modern computer applications deal with visual outputs. Graphical user interface has almost entirely replaced command-based interaction in many domains. Information visualization can be defined as "the use of computer supported, interaction, visual representations of abstract data to amplify cognition" (Shneiderman, 1992). To amplify cognition means that visualization shifts cognitive loads to the perceptual system, thus expanding working memory and information storage.

Visualization provides a more perceptually intuitive way of viewing raw data, thus allowing users to identify relevant patterns which would not have been identified in raw data.

Therefore, it has a huge impact on many applications domains, ranging from engineering, education, various fields in science, and so forth.In HCI, the most obvious application is the use of visualization is in the design of graphical user interface that allows more intuitive interaction between human and computers. Various innovative interaction styles have been developed such as WIMP (window, icon, menu, pointing device) which is so familiar in today's software. Three-dimensional graphics are also emerging although currently they are mostly used in computer games and computer-aided design. One recent example of 3D graphical interface is the new windows navigation and management known as Windows Flip 3D in Windows Vista which allows the user to easily identify and switch to another open window by displaying 3D snapshot thumbnail preview of all windows in stack.

Today, Information visualization is not only about creating graphical displays of complex information Structures. It contributes to a broader range of social and collaborative activities. Recently, visualization techniques have been applied on social data to support social interaction, particularly in CMC. This area is known as social visualization by (Donath, Karahalios, & Viégas, 1999). Other technique such as social network analysis has also become increasingly important in visualization social data.

Other areas where HCI plays an important role include: Intelligent and agent systems; Interaction design; Interaction through wireless communication networks; Interfaces for distributed environments; Multimedia design; Non–verbal interfaces; Speech and natural language interfaces; Support for creativity; Tangible user interfaces; User interface development environments and User support systems.

ORGANISATIONAL AND SOCIAL IMPLICATIONS OF HUMAN COMPUTER INTERACTION

The application of HCI can have an effect of organisational and social dimensions. For example, the area of computer supported collaborative work (CSCW) explores the effect the introduction of technology can have an effect on the organisational structure and the way of work of companies and organisations. Similarly the study of how we use technology to communicate with each other is gaining strong interest in the HCI research community.

The expansion of the Internet has resulted in an increase in the usefulness of Computer Mediated Communication (CMC) and the popularity of online communities. It is estimated that 25% of Internet users have participated in chat rooms or online discussions (Madden & Rainie, 2003). It is by now no secret how vital the Internet was, is, and will continue to be in our lives. One of the most important characteristics of this medium is the opportunities it offers for human-human communication through computer networks. As Metcalfe (1992) points out, communication is the Internet's most important asset. E-mail is just one of the many modes of communication that can occur through the use of computers. Jones (1995) points out that through communication services, like the Internet, Usenet and bulletin boards, online communication has for many people supplanted the postal service, telephone and even the fax machine. All these applications where the computer is used to mediate communicationare called Computer-Mediated Communication (CMC).

December (1997) defines CMC as "the process by which people create, exchange, and perceive information using networked telecommunications systems (or non-networked computers) that facilitate encoding, transmitting, and decoding messages". He emphasizes that studies of CMC view this process from different interdisciplinary theoretical perspectives (social, cognitive/psychological, linguistic, cultural, technical, and political) and often draw from fields such diverse as human communication, rhetoric and composition, media studies, human-computer interaction, journalism, telecommunications, computer science, technical communication and information studies.

Online communities emerge through the use of CMC applications. The term online community is Multidisciplinary in nature, means different things to different people, and is slippery to define (Preece,2000). For purposes of a general understanding of what online communities are, Rheingold's definition of online communities is presented:

[Online] communities are social aggregations that emerge from the Net when enough people carry on those public discussions long enough, with sufficient human feeling, to form Webs of personal relationships in cyberspace.

Online communities are also often referred to as cyber societies, cyber communities, Web groups, virtual communities, Web communities, virtual social networks and e-communities among several others. The cyberspace is the new frontier in social relationships, and people are using the Internet to make friends, colleagues, lovers, as well as enemies (Suler, 2004). As Korzeny pointed out, even as early as 1978, online communities are formed around interests and not physical proximity (Korzeny, 1978).In general, what brings people together in an online community is common interests such as hobbies, ethnicity, education, beliefs. As Wallace (1999) points out, meeting in online communities eliminates prejudging based on someone's appearance, and thus people with similar attitudes and ideas are attracted to each other.

Preece et al. (2002) states that an online community consists of people, a shared purpose, policies and computer systems. She identifies the following member roles:

- **Moderators and mediators:** who guide discussions/serve as arbiters
- **Professional commentators:** who give opinions/guide discussions
- **Provocateurs:** who provoke
- **General Participants:** who contribute to discussions
- **Lurkers:** who silently observe

Social Network Analysis (SNA)

"Social Network Analysis (SNA) is the mapping and measuring of relationships and flows between people, groups, organisations, computers or other information/knowledge processing entities. The nodes in the network are the people and groups while the links show relationships or flows between the nodes. SNA provides both a visual and a mathematical analysis of human relationships" (Krebs, 2004, pp.1). Preece (2000) adds that it provides a philosophy and a set of techniques for understanding how people and groups relate to each other, and has been used extensively by sociologists (Wellman, 1982; Wellman 1992), communication researchers (Rice, 1994; Rice et al., 1990) and others. Analysts use SNA to determine if a network is tightly bounded, diversified or constricted; to find its density and clustering; and to study how the behavior of network members is affected by their positions and connections(Garton, Haythornhwaite & Wellman, 1997; Wellman, 1997; Henneman, 1998; Scott, 2000).Network researchers have developed a set of theoretical perspectives of network analysis. Some of these are (Bargotti, 2002):

- Focus on relationships between actors than the attributes of actors.
- Sense of interdependence: a molecular rather atomistic view.
- Structure affects substantive outcomes.
- Emergent effects.

"The aim of social network analysis is to describe why people communicate individually or in groups"(Preece, 2000, pp. 183), while the goals of SNA are (Dekker, 2002):

i. To visualize relationships/communication between people and/or groups using diagrams.
ii. To study the factors which influence relationships and the correlations between them?
iii. To draw out implications of the relational data, including bottlenecks.
iv. To make recommendations to improve communication and workflow in an organisation.
v. Preece (2002) and Beidernikl & Paier (2003) list the following as the limitations of SNA:
vi. More theory that speaks directly to developers of online communities is needed
vii. The data collected may be personal or private.
viii. The analysis of the data is quantitative and specific to the particular network, while common survey data are qualitative and generalize answers on the parent population.

It is also worth pointing out that network analysis is concerned about dyadic attributes between pairs of actors (like kinship, roles, and actions), while social science is concerned with monadic attributes of the actor (like age, sex, and income). There are two approaches to SNA:

I. **Ego-centred analysis:** Focuses on the individual as opposed to the whole network, and only a random sample of network population is normally involved (Zaphiris, Zacharia, & Rajasekaran, 2003). The data collected can be analyzed using standard computer packages for statistical analysis like SAS and SPSS (Garton, Haythornthwaite, & Wellman, 1997).

II. **Whole network analysis:** The whole population of the network is surveyed and this facilitates conceptualization of the complete network (Zaphiris et al., 2003). The data collected can be analyzed using microcomputer programs like UCINET and Krackplot (Garton et al., 1997).The following are important units of analysis and concepts of SNA (Garton et al., 1997; Wellman,1982; Hanneman, 2001; Zaphiris et al, 2003; Wellman, 1992):

- Nodes: The actors or subjects of study.
- Relations: The strands between actors. They are characterized by content, direction, and strength.
- Ties: Connect a pair of actors by one or more relations.
- Multiplexity: The more relations in a tie, the more multiplex the tie is.
- Composition: This is derived from the social attributes of both participants.
- Range: The size and heterogeneity of the social networks.
- Centrality: Measures who is central (powerful) or isolated in networks.
- Roles: Network roles are suggested by similarities in network members' behaviour.
- Density: The number of actual ties in a network compare to the total amount of ties that the network can theoretically support.
- Reachability: In order to be reachable, connections that can be traced from the source to the required actor must exit.
- Distance: The number of actors that information has to pass through to connect the one actor with another in the network.
- Cliques: Sub-sets of actors in a network, who are more closely tied to each other than to the other actor who are not part of the subset.
- Social Network Analysis is a very valuable technique when it comes to analyzing online communities as it can provide a visual presentation of the community and more importantly it can provide us with qualitative and quantitative measures of the dynamics of the community.

Summary

human factors (human factors engineering or human factors psychology)	Approach that uses knowledge of human capabilities and limitations to design systems, organizations, jobs, machines, tools, and consumer products for safe, efficient, and comfortable human use

Ergonomics	The study of the physical demands of work such as reaching, stretching, lifting, and carrying
User friendliness	Positive characteristic of machines, tools, and consumer products that are designed to be comfortable, easy to use, and compatible with human capacities and limitations
Input component	Component that provides information to a human or computer
Output component	Component that receives information from a human or computer and converts that information to action
Display	Device such as a computer screen that provides an individual with information
Control	Device such as a keyboard or mouse that permits an individual to take actions
Musculoskeletal disorders	Disorders of the lower back and upper extremities (arm/shoulder/wrist). They are the most commonly studied injuries related to workplace safety
Human-computer interface	The interaction between a human and a computer

CHAPTER FOURTEEN

EMERGING TRENDS IN HUMAN COMPUTER INTERACTION

Computer Game, Play and Immersion

Researchers from various areas have begun exploring the role of emotion in human activities. This research has proven to yield important results that not only deepen our understanding on human emotion, but potentially improves the design and development of artefacts. This goes without saying that the topic of emotion is becoming increasingly important in HCI. Probably one of the most studied domains in emotional design is the computer game. Computer games have been widely recognized as the software which is potentially addictive due to their capability to engage and immerse the players for hours of game play. Therefore, it can be claimed that emotion is vital to maintain players' focus and to create enjoyable experiences. Unlike work-oriented activity (and thus work-oriented software that mediate them), game play traditionally does not focus on productive outcome. In this case, usability of game play cannot be understood in a traditional sense. Already new metrics of usability (or playability) have been proposed for designing playful artefacts/software. Some effort has also been taken to incorporate game elements into productive activities. For instance, a project has been carried out to use games to label the contents of images meaningfully. Others have used games for education and training. An area of research, known as serious games, is expanding quickly to study productive games. Recently, playability design has undergone a major transformation as games are becoming increasingly collaborative with the emergence of massively multiplayer online role-playing games (MMORPGs).

These games are becoming one of the most interesting interactive media of computer-mediated communication and networked activity environments (Taylor, 2002). Understanding the pattern of participation in these game communities is crucial, as these virtual communities function as a major mechanism of socialization of the players. Some usability studies have shown that MMORPG design should incorporate what is known associability. For instance research has found that game locations can be designed to encourage different styles of social interactions

CMC, Online Community and Social Computing

Secondly, with the development of high performance computer networks, the use of computers is becoming increasingly collaborative. Therefore, topics such as CSCW, CMC, or social computing in general will continue to thrive.

One of the studied issues in this research topic is the use of non-verbal cues in the CMC technology. Some newer CMC media have attempted to address this issue through the implementation of video streaming, although a more common way is the use of "emoticons" that allows the users to express their emotions through graphical icons that show symbolic facial expressions.

Identity and anonymity is another popular topic regarding CMC since it is possible for users to hide their real identity simply by choosing a different name. In most cases, even the administrators do not know their real identity. Furthermore, users may treat CMC as an experimental environment and thus "play" with their identity and try out different personalities (Turkle, 1995). Some studies have been carried out to compare the nature of friendship in conventional CMC and offline communication. Wellman and Gulia (1999) for instance found that people have a higher number of online friends because it is easier to make friends online than offline. However, the quality of the relationship is weaker in an online setting. Although Walther (1995) contended that it is possible to develop strong online friendship, it takes longer than in offline communication.

3D CMC and Virtual World

Conventional CMC technologies are mainly text-based (Pfeil, 2007). Although the fundamental technology of CMC has not changed in the last decade (Preece & Maloney-Krichmar, 2003), the way of using it for human-human communication has evolved considerably, from e-mail and synchronous chatting to online forum, blogs and Wikis.

New technologies such as 3D Virtual spaces, such as MMORPGs and Second Life (Linden Lab, 2003) offer a much richer form of social interactions where users can not only communicate through text-based chat, but also interact with each other through virtual artefacts. Furthermore, these technologies not only mediate social interaction, but also support group formation, which leads to community building. As such, user interaction is considerably more complicated than the conventional CMC tool.

These 3D worlds often feature a large number of virtual locations users can visit and a variety of virtual artefacts users can use to interact with, with each other. In some cases, they can even construct new artefacts and locations utilising simple modelling tools and scripting languages. Whilst some 3D virtual worlds are designed with game-like goal structures that impose obstacles or challenges, some are completely open, meaning that the users are free to do as they please.

Although sociability issues of conventional CMC are well studied and documented, we have very little understanding on social interactions in 3D CMC. Therefore, it is worth investigating user activities in such environments in order to cast some light on the group formation process and other sociability issues. It might be potentially more challenging in researching this 3D CMC, in which communication takes place both through texts and other "virtual actions" users can perform with their 3D avatars. Unlike the avatar in conventional CMC which is often a static graphical or animated representation of the user, in 3D CMC, the avatar can interact with other avatars directly in the virtual space. A 3D avatar can perform a wide range of actions on the 3D world and other avatars. For instance, it is not uncommon that avatars can hug, kiss or wave to each other. There is also research on the facial expression of 3D avatars (Clarkson et al., 2001), implemented in the 3D CMC context with the intention to enhance non-verbal communication. Moreover, given the fantasy theme of some 3D CMC environments, new sets of rules for virtual communication which are completely different from physical communication might arise. This is worth investigating as well as groups open operate within the boundary of norms and rules that emerge through user interaction.

Ubiquitous Computing

Another exciting future trend in HCI is the emergence of ubiquitous computing, in which information processing is thoroughly diffused into objects and experiences of everyday life. In another word, computers are disappearing. Users are no longer consciously engaged in using the computers. Instead they are operating these devices which are so well integrated into artefacts of everyday activities without being aware of using the computers. Perhaps the most obvious example is the mobile phone, and indeed mobile computing has witnessed an explosion of research interest within and beyond HCI community. Other less obvious examples could include computerized refrigerators which are able to detect their contents, plan and recommend a variety of recipes, automatically shop according to the users' needs. The focus of ubiquitous computing from the point of view of HCI suggests a shift from tool-focused design to activity-focused design. The primary objective is thus to design tools which can be seamlessly mediate everyday activities without interfering with users' tasks. One such area is wearable computing which has the potential to support human cognitions, facilitate creativity and communication. Unlike mobile devices, wearable computers are attached to human, thus reducing the possibility of interruption or displacement.

Novel Interaction Techniques

With the integration of the computers into everyday life, there is a call for new interaction techniques as keyboards and mice are no longer sufficient as input devices.

Techniques such as natural language processing (NLP), a sub area of artificial intelligent, have been incorporated into the design of computer devices to recognize speeches, thus making interaction more intuitive. This automatic speech recognition can not only convert spoken words into machine readable strings, but also recognize and identify the person who speaks. Already, it has some impacts on our everyday life particularly on disable people who suffer from motor control impairment. Perhaps a more ambitious application of NLP is to go beyond text-speech or speech-text conversion to actually understand human natural language. The research goal in this area is to develop a machine that can engage in meaningful conversation with human. Although still in its infancy, it is certainly a promising area which will benefit HCI greatly. Another interesting interaction mode is what is known as haptic interaction which draws largely from the infamous interaction style in HCI, direct manipulation. The idea is to provide the users with greater freedom of manipulation that includes touch and feel. This technique aims to effectively bridge the gap between tangible interfaces and virtual/intangible representations. One example is the augmentation of everyday objects in which digital data is superimposed on physical objects which can be used to manipulate the digital data.

Recent development has also begun to utilize other parts of our body for interaction. Eye tracking for instance, which has been used extensively in research in psychology, cognitive linguistic and other areas, has been adopted in HCI research to study usability of Web sites and other software. Recently, its use in HCI has expanded to be applied as an interaction technique. Eye tracking system is now being used especially for the disabled to surf the Web, e-mail, and so forth with eye movements.

Accessibility and Universal Design

Finally we believe that Endeavour in designing assessable technology will continue to be a main focus in HCI. One major accessibility research is the Web design since the Web has rapidly become more and more pervasive in almost everybody's lives.

The challenge lies in the creation of resources that can be used by the widest spectrum of potential visitors rather than an idealized "average," there is a need to consider universality of Web delivery.

The idea accessibility has gone beyond the basic concept of "designing for disability" to universal design, allowing access by everyone.

This requires consideration of the needs and requirements of individuals with disabilities, older persons, people for whom English is a second language, people whose cultures and backgrounds are dissimilar to those of Web developers, and those using outdated hardware and software, to name a few. For that reason, the key objective of this research direction is to look at various aspects of universal Web design and evaluation in a new perspective by focusing on the user aspect of universal Web design and interaction and to present the wide range of advanced technology that can help disadvantaged users get access to Web information.

The same idea also applies to software design. For instance, child-computer interaction has emerged as a sub-field of HCI that deals with designing particularly multimedia learning software for children.

Other active areas of research include wed design for elderly people, gender differences, cultural issues, and so forth.

All these efforts in accessibility design no longer contribute to the development of ease-of-use applications for specific user groups, but also improve interaction experiences for main stream users in general.

The Usability Engineering

Software Engineering

Software is an intellectual artefact and a kind of instructive information that provides a solution for a repeatable computer application, which enables existing tasks to be done easier, faster, and smarter, or which provides innovative applications for the industries and daily life. Large-scale software systems are highly complicated systems that have never been handled or experienced precedent by mankind.

The fundamental cognitive characteristics of software engineering have been identified as follows (Wang, 2006g):

- The inherent complexity and diversity
- The difficulty of establishing and stabilizing requirements
- The changeability or malleability of system behaviour
- The abstraction and intangibility of software products
- The requirement of varying problem domain knowledge
- The non-deterministic and polysolvability in design
- The polyglotics and polymorphism in implementation
- The dependability of interactions among software, hardware, and human beings

The above list forms a set of fundamental constraints for software engineering, identified as the cognitive constraints of intangibility, complexity, indeterminacy, diversity, polymorphism, inexpressiveness, inexplicit embodiment, and unquantifiable quality measures (Wang, 2006g).

A set of psychological requirements for software engineers has been identified, such as:

(a) Abstract-level thinking; (b) Imagination of dynamic behaviours with static descriptions; (c) Organization capability; (d) Cooperative attitude in team work; (e) Long-period focus of attentions;

(f) Preciseness; (g) Reliability; and (h) Expressive capability in communication.

Users' Mental Models

Along with their capabilities and limitations, users bring with them their previous experience of computers and Web sites. We assume they have built mental models, that is, psychological representations of the ways in which computers and Web sites work (Carroll, 1990; Johnson-Laird, 1983; Van Der Veer & Del Carmen PuertaMelguizo, 2003).Highly experienced users can have mental models of different categories of Web sites, for example, sites for entertainment, information gathering, and e-commerce. According to the user's mental model, an entertainment site should use bright colours and animation, whereas an information site should use subdued colours and minimal animation, for example, only to demonstrate an integral concept. Abstracting and representing users' mental models is another job for the UE. Tools for doing this include cognitive task analysis

(e.g., Crandall, Kline, & Hoffman, 2006).User expectations for Web sites are based on their experience with the Internet. They have expectations for the behaviour of controls, for example, the back and forward button, Tab key, and Enter key. They have expectations for hyperlinks, for example, a link once activated will change its colour. Some expect to be told whether clicking on a link will take them outside the site. Sometimes, users worry about whether they will be able to get back to where they are if they click on a link because they have been unable to form a definite mental model for link behaviour. This uncertainty can arise when they have experienced links behaving inconsistently across sites.

Heuristic Evaluation

Usability: Fit for use; convenient to use

Does it do what users need? **Usability** and **utility** are equally important: It matters little that something is easy if it's not what you want. It's also no good if the system can hypothetically do what you want, but you can't make it happen because the user interface is too difficult. "If the user can't find it, the functionality's not there"

Phases of Heuristic Evaluation

1. Pre-evaluation training gives evaluators needed domain knowledge (and scenario, if necessary - not necessary if product is for walk-up use)
2. Evaluation individuals evaluate and then aggregate results
3. Severity rating determine how severe each problem is (priority) can do this first individually and then as a group
4. (Optional) Debriefing from evaluators discusses the outcome with design team. Way of getting additional design advice

How to Perform Evaluation

At least two passes for each evaluator

1. First to get feel for flow and scope of system second to focus on specific elements
   If system is walk-up-and-use (or evaluators are domain experts) no assistance needed otherwise might supply evaluators with scenarios.
2. Each evaluator produces list of problems explains *why* with reference to heuristic or other information which are *specific* and list each problem separately
3. Severity Rating: Used to allocate resources to fix problems.

4. Estimates of need for more usability efforts combination of
   - Frequency
   - Impact
   - Persistence (one time or repeating)

0 - don't agree that this is a usability problem

1 - cosmetic problem

2 - minor usability problem

3 - major usability problem; important to fix

4 - usability catastrophe; imperative to fix

GLOSSARY

Accessibility: The measure of whether person can perform an interaction, access information, or do anything else. It does not measure how well he or she can do it, though.

Content: The information, such as thoughts, ideas, and so forth, that someone wishes to communicate. Examples of content could be the ideas and concepts conveyed through this article, the fact that you must stop when a traffic light is red, and so on. Importantly, content is what into be communicated but not how it is to be communicated.

Encoding: Encoding is the process by which the content and meaning that is to be communicated is transformed into a physical form suitable for communication. It involves transforming thoughts and ideas into words, images, actions, and so forth, and then further transforming the words or images into their physical form.

Object Orientation: A view of the world based on the notion that it is made up of objects classified by a hierarchical super class-subclass structure under the most generic super class (or root) known as an object. For example, a car is a (subclass of) vehicle, a vehicle is a moving object, and a moving object is an object. Hence, a car is an object as the relationship is transitive and, accordingly, a subclass must at least have the attributes and functionality of its super class (es).Thus, if we provide a generic user-presentation object with a standard interface, then any of its subclasses will conform to that standard interface. This enables the plug and play of any desired subclass according to the user's encoding and decoding needs.

Physical Form: The actual physical means by which thoughts, meaning, concepts, and so forth are conveyed. This, therefore, can take the form of any physical format, such as the writing or displaying of words, the drawing or displaying of images, spoken utterances or other forms of sounds, the carrying out of actions (e.g., bodily gestures), and so forth.

Software Architecture: Rather like the architecture of a building, software architecture describes the principled, structural design of computer software. Contemporary software architectures are multitier (or *n*-tier) in nature. Essentially, these stem from a two-tier architecture in which user-presentation components are separated from the information-content components, hence the two overall tiers. Communication occurs through standard interface between the tiers. This enables the easy swapping in and out of presentation components, thus enabling information to be encoded into the most appropriate physical form for a given user at any given time.

Usability: A measure of how well someone can use something. Usability, in comparison to accessibility, looks at factors such as ease of use, efficiency, effectiveness, and accuracy. It concentrates on factors of an interaction other than whether someone can perform something, access information, and so forth, which are all handled by accessibility.

MODULE EXAMINATION SAMPLE QUESTIONS AND MODEL ANSWERS

EXAMINATION QUESTIONS

Q1 Name some other tasks for which computers are used, and for which special purpose interfaces are required.

- *Navigation systems (land, sea and air), washing machines, machines for developing photographs, touch screens in tourist information centres, flight simulators.*

Q2 In the early days of cash machines, it was found that users sometimes forgot to remove their cards after withdrawing their cash. What simple change was made to eliminate this fault?

- *Cards have to be removed before the cash appears. Change HCI design by prompting user to remove card before cash is given out.*

Q3 Identify TWO situations in which a command driven interface would be appropriate.

- *A command-driven interface is appropriate for a technical user or a user who will undergo training and use the same software all the time.*

Q4 Name some other situations where voice recognition would be appropriate.

- *Data entry for people affected by RSI (Repetitive strain injury); military aircraft, lifts (elevators)*

Q5 Give other examples of sophisticated HCIs

- *Digital watch, video recorder, touch screen on sophisticated photocopying machine.*

Q6 A different HCI would be needed for each of the following users:

i) a young child in a primary school

ii) a blind person

iii) a graphic artist

For each user describe and justify an appropriate HCI. **(9 marks)**

i)WIMP interface: Touch sensitive screen or concept keyboard or mouse (accept keyboard), because easy to use or appropriate reason e.g. child cannot read.

(ii) Command driven interface: Keyboard (Braille) or voice I/O, Braille printer, because cannot see screen

(iii) WIMP; graphics pad, high resolution screen, digitiser, plotter. Must respond to sensitivity of touch/resolution (or any reasoned alternative)

7. User interfaces have gradually become more and more oriented to the needs of users over recent years.

a) Briefly describe three features of user interfaces which have been developed and explain how each has benefited the user (3 marks)

b) Describe two ways in which user interfaces need to be developed further to make computers more accessible and friendly to untrained users. (2 marks**)**

(a) *We would like to see a 3-3 split here but accept up to 4-2 in either direction.*

- ***Physical factors****: Max. 4 @ 1 each from : position of screen, lighting conditions, seating conditions, choice of colour schemes, etc., ergonomics/design of mouse/keyboard ventilation/room temperature*

- *Psychological factors: Max. 4 @ 1 each from: user friendly interface (qualified), help available for novice users, short cuts for expert users, make use of human long term memory to maximise efficiency, functionality, technophobia*

(b) Three points. In each case:

for the factor (1) for a clear explanation of its impact on systems resources (1)

- *NB: More than one of the resource implications: a greater demand for memory/IAS/backing store and processor functionality and time/speed, might apply to the same factor of the H.C.I. However, candidates can only gain 1 mark for the resource implications of each factor.*
- *on-line help availability- increased need for backing store (2,1,0)*
- *complexity of interface/ multiplicity of menu routes adds to size of resultant code thus increased IAS demands (2,1,0)*
- *use of GUI- increased IAS demands (2,1,0)*
- *need for multi-tasking/ability to switch between applications/tasks - processor functionality overhead (2,1,0)*
- *Faster searching of help file -processor speed overhead. (2,1,0)*

Q8 Many machines now offer graphical user interfaces such as Windows and the Mac OS.

i) Describe **two** features of such interfaces, which are likely to be helpful to a non-technically minded user. **(2 marks)**

ii) Describe **three** advantages of this type of interface. (3 marks)

a) Features include:

icons which indicate their meaning	*pointer controlled by mouse*
easy selection from a menu	*choose by clicking on icon, or*
use of hot keys	*cancel/undo commands*

(b) Able to respond to speech input

point and touch screens	*more help facilities*
better error messages	*commonality*

Q9 A school runs two versions of the same word processing package on its network. Both versions allow users to type in text, but Version A accepts only keyed commands whereas with Version B the user may use a mouse as well as the keyboard.

a) Describe how the use of a mouse can be helpful during Word Processing operations. Give a range of examples to justify your answer. **(6 marks)**

b) At some point in the future it seems likely that computers will be able to receive input in spoken form. Discuss, with examples, how this development could affect the design and use of word processing packages**. (6 marks)**

a) A mouse is helpful because it is faster to perform tasks and easier to see what you are operating on (selected text highlighted). e.g. moving round text is faster with point and click, selecting text is faster using point and drag. Option selection is easier using dialogue boxes, or pop up menus using right mouse button.

b) At the simplest level text could be entered in spoken form rather than being typed in. or punctuation, could use icons on a tool bar which could be selected and placed where required or there could be a punctuation button to accept the next word as punctuation e.g. "full stop" in a similar way there could be a command button which when selected would take the next word as a command.

e.g. speak the text to be highlighted, press command button then say "Centre"

or could have the three mouse buttons mapped to text Punctuation Command

Other commands such as page set-up could be spoken, and the dialogue box could appear on the screen and the required settings spoken.

Q10 Considerable efforts have been made to provide powerful yet intuitive user interfaces to a wide range of application packages, operating systems and programming languages.

a) Identify a range of facilities that could be provided and discuss the perceived need for such facilities. **(4 marks)**

b) Describe how you might assess the effectiveness of such interfaces. **(1 mark)**

a) An operating system could provide ***icons*** *representing software packages and* ***mouse support*** *so that the user can load software by clicking on the icon. It could provide easier ways of copying files, by using a mouse to move a file from one directory list to another.*

Editing facilities *in all types of software could be provide to so that the mouse is used, for example, to highlight text, and a* ***menu*** *of options allows the user to select, cut and paste. These facilities are need because it is hard to remember the correct syntax for performing infrequently used commands. Computers are no longer the province if the trained expert; people from many different skill areas want to use, for example word processors, spreadsheets and databases and do not want to learn complex operating system commands.*

b)The effectiveness of such interfaces could be assessed by the number of copies of a package like Microsoft Windows, which uses these techniques, which are sold worldwide.

11. A particular institution uses a range of software packages from different suppliers, each with a different user interface. You are asked to advise the institution on the advantages of using software packages with a common user interface.

Give FOUR advantages of having a common user interface (**4 marks)**

Describe ONE specific aspect of a user interface that would benefit from being common between software packages **(2 marks)**

A user interface which has already been configured by an IT expert may not have been customised to the exact needs of the user. This can result in poor use of the package by the user who may not know how to reconfigure the interface.

Describe a feature that is subject to this kind of configuration **(2 marks)**

(a)	
common commands *ease of use - ease of navigation* *nsistency between packages*	*increased speed of learning - training not needed* *confidence building of users - particularly novices* *Reduces support overheads*
(b) *buttons are same type* *colour of screen* *style/colour of error messages* *sition of menus*	*consistency (1) qualified (1)* *mouse buttons* *position of error messages* *same use of hot keys* *nsistency of defaults*
(c) *page view or layout set incorrectly* *wrong toolbar selected*	*on-line spell checking rather than at end* *wrong colour, reconfiguration of colour*

12. The choice of a user interface has been described as 'one of the most critical areas of software consideration'. Many machines now offer a front end GUI (Graphical User Interface) or WIMP environment.

(a)

i. Describe the advantages of this type of interface to an average user (**2 Marks)**
ii. Describe the potential disadvantages of this type of interface to an experienced user (2 marks

iii. Describe the advantages and disadvantages in the use of this type of interface to the system. **(4 marks)**

(b) A particular user installs a word processing package, a spreadsheet package and a DTP package on a machine; all operate under such a front end environment. Describe TWO examples of the types of data exchanged between the packages and explain why this process may be easier than on a system which does not use a front end. **(4 marks)**

(a)user friendly point and click, providing easy access to a range of facilities e.g. filing, fonts, device management

gets in the way', more time consuming/greater number of operations to do a simple task, may be several levels of menu before the one required, direct example contrasts to command line, provides consistent environment, easier to data exchange using 'clipboard', device management globally set, multi-tasking, DDE (dynamic data exchange)

(b) data exchanged wp-dtp... raw text for formatting/page layout

sp-dtp or wp sheets for inclusion

easier environment takes care of formatting changes of file transfer between the packages automatically.

13

a) Give **six** of the physical and psychological factors which govern how people interact with computer systems. **(6 marks)**

b) Give **three** factors which should be considered when providing a sophisticated human computer interface, explaining the impact of each one on the system's resources **(6 Marks)**

14.

A university uses a complex CAD (computer aided design) package. The package has a sophisticated human-computer interface which also places considerable demands on the system's resources.

(a) Give **two** examples of a system's resources that would be affected by such a package and explain the demands placed upon them (4)

(b) Describe **three** features you would expect to find in the human-computer interface which would merit the description 'sophisticated'. (6)

(a) In each of 2 examples: 1 for the resource and 1 for the demand on it.

***Backing storage (1)** - requires sufficient capacity to cope with large graphics files / large help-files/ size of application code (1)*

***IAS (1)** requires sufficient capacity to cope with multiple graphical user windows (1)*

***Processor (1) :** needs sufficient clock speed to cope with additional processing involved in smooth presentation of Graphics display (1)*

etc. Avoid repetition any 2 x (1+1)=4

(b) In each case give 1 for naming the feature and then 1 for describing it. Descriptions must make it clear that some level of sophistication exists.

on-line help (1) with context sensitive searching of different topics (1)

effective use of colour (1) to highlight on screen message e.g. warning messages in red, suggestions in a different colour (1)

well designed command/menu structures taking into account skills of CAD designer (1) with shot cut keys for experienced users (1)

use of a range of input/ output devices appropriate to CAD design work (1) e.g. graphics tablet, plotter, etc. (1) – 2 examples gets 2 marks

user friendliness (1) takes account of design skills and terminology used by designers (1)

GUI (1) – presents complex information in graphical/icon format (1)

– features must be appropriate to CAD e.g. not voice recognition.

Max. 3x (1+1)=6

15. The workstations on a particular company network are set up to allow each user to change software, menus, icons and color schemes to suit his or her own preferences. These variations make support for users difficult to manage.

The network manager wants to change to a standard user interface, so that all the users will be given the same set of menus, icons and color schemes.

(a) (i) Describe two benefits for the users of this standardized approach, other than improved support. (4marks)

(ii) Describe two of the disadvantages for the users of this standardized approach. (4marks)

(b) What are the resource implications for planning this standardized interface? (4marks)

(a) (i) Benefits for users:

- *ease of learning - training material can he written to match workstation I-ICI ,(2,1,0)*
- *easier transfer of skills to new packages if interface is maintained, (2, 1,0)*
- *consistency of interface maintained when users move between workstations, (2,1,0)*
- *better self support between users, (2, 1,0)*

- *'standard settings for defaults' e.g. Word starting each sentence with a capital letter. (2,1,0)*
- *Easier distribution/use of standardized items such as templates/logos etc*

(ii) Disadvantages for users:

- *Level of skills for different users - standard may suit less able but not highly skilled, (2,1,0)*
- *Have to wait for changes in software configuration rather than do it themselves, (2,1,0)*
- *May not be able to use favourite specialised software unless is included in standard (2,1,0)*
- *Standard colour sets may not be appropriate for colour blind or other disabilities (2,1,0)*
- *User no longer has control over their own desktop environment plus expansion (2,1,0)*
- *Either need for additional training/support OR user may be confused plus expansion (2,1,0)*

6 (b) Resource implications:

- *Need to upgrade some workstations if they cannot support standard (1)*
- *Upgrades may include hard-drive, screen (1)*
- *May involve moving software from local workstations to a server (1) May involve increased network traffic (1)*
- *Need to upgrade server storage capacity to accommodate move of apps from workstations to a server (1)*
- *Need to upgrade networking infrastructure from ring to star (1)*
- *Time consideration for restructuring (1)*
- *Use of staff for restructure (1)*
- *Time lost during changeover (1)*
- *Network management software enables this change to occur (1)*

16. A mail order music company has decided to expand and has established a retail outlet in a busy shopping centre.An important feature of the mail order system is the interface for the staff who use it.

a) State three features you would expect the human/computer interface to have in such a system and give a different reason for each one. (6 marks)

b)

i. Name an appropriate device for capturing data on each item that is sold via the retail outlet. (1 mark)

ii. Describe one advantage for the company of using this device. (2 marks)

a.

- *Consistency with other systems (1) so that users are less likely to make errors (1)*
- *Automated data entry (1) to reduce errors (1)*
- *Cater for different levels of user expertise (1) workers may not be very ICT literate (1)*

- *Cater for different end user's physical abilities (1) e.g. for partially sighted (1)*
- *Consistent at both sites (1) so staff do not have to learn two systems (1)*
- *Sensible use of colour (1) as the system will be used fairly intensively (1)*
- *Help features accessible (1) so that users are able to assist themselves when they need to (1)*
- *Menu based system (1) so that input choices are restricted to items sold (1)*
- *Graphical user interface (1) to build on users previous experience/ to avoid language issues/ etc (1)*

Credit any feature related to interface that fits within the given context.

Second mark is given for expansions that can be justified within context.

3 x (2,1,0) marks

b.

i.

- *Bar Code Scanner (1)*
- *Key Board (1)*

1 mark

ii. This description must relate to the answer to (i) in order to get the marks

e.g. Bar Code Scanner

- *simple method (1) needs minimal training so staff can be working quickly (1)*
- *stock is already provided with bar code from supplier (1) so little extra work required in preparation (1)*
- *Speed of data capture c/w other methods (1) so that store increases productivity (1)*
- *Etc.*

e.g. Key Board

- *cheap method c/w bar code scanner (1)*
- *simple method (1) needs minimal training so staff can be working quickly (1)*
- *Etc.*

17. *A* school has approached you for advice as it plans to develop a computer-based learning environment for its pupils aged from 6 to 10.

(a) Describe **two** factors the school should consider whilst designing the interface for the computer system. *(4 marks)*

(b) State, with a reason:

(i) **One** example of a suitable input device that could be used by the pupils; *(2 marks)*

(ii) **One** example of a suitable output device that could he used by the pupils. *(2 marks)*

18. There are several types of human/computer interface.

(a) (i) Describe one feature of a command line interface. *(2 marks)*

(ii) Name, giving **one** reason, one application where this interface would be appropriate.

(2 marks)

(b) (i) Describe one feature of a menu driven interface. ***(2 marks)***

(ii) Name, giving **one** reason, **one** application where this interfaces would he appropriate.

(2 ***marks)***

a. (i) Describe **one** feature of a graphical user interface. *(2* ***marks)***

(ii) Name, giving **one** reason, **one** application where this interface would be appropriate. *(2 marks)*

References

Applied Ergonomics. (1974). *Applied Ergonomics Handbook.* Surrey, England: IPC Science and Technology Press.

Arthur, W., Jr., Barrett, G. V., & Alexander, R. A. (1991). Prediction of vehicular accident involvement: A meta-analysis. *Human Performance, 4,* 89-105.

Avolio, B. J, .Kroeck, K. G., & Panek, P. E. (1985). Individual differences in information processing ability as a predictor in motor vehicle accidents. *Human Factors, 27,* 577-588.

Aycan, Z., & Kanungo, R. N. (2001) Cross cultural industrial and organizational psychology: A critical appraisal of the field and future directions. In N. Anderson, D.S. Ones, H. K. Sinangil, & C. Viswesvaran (Eds.). *Handbook of Industrial, Work, and Organizational Psychology: Volume 1* (pp. 385-408). London: Sage Publications.

Baltes, B., Briggs, T. E., Huff, J. W., Wright, J. A., & Neuman, G. A.(1999). Flexible and compressed workweek schedules: a meta-analysis of their effects on work-related criteria. *Journal of Applied Psychology, 84,* 496-513.

Barrett, G. V., & Thornton, C. L. (1968) The relationship between perceptual style and driver reaction to an emergency situation. *Journal of Applied Psychology, 52,* 169-176.

Beede, K. E., & Kass, S. J. (2006). Engrossed in conversation: The impact of cell phones on simulated driving performance. *Accident Analysis & Prevention, 38,* 415-421.

Bentley , R., Hughes, J. A., Randall, D., Rodden, T., Sawyer, P. (1992) Ethnographically-informed system design for air traffic control. Proceedings of CSCW 1994 Conference on Computer Supported Cooperative Work, pp. 123-129. New York: Assoc. Comput. Mach.

Buchanan, D. A., & Boddy, D. (1983). Advanced technology and the quality of working life: The effects of computerized controls on biscuit making operators. *Journal of Occupational Psychology,* 56, 109-119.

Byrnes, J. P., Miller, D. C., & Schafer, W. D. (1999). Gender differences in risk taking: A meta-analysis. *Psychological Bulletin, 125,* 367-383.

Campion, M. A. (1988). Interdisciplinary approaches to job design: a constructive replication with extensions. *Journal of Applied Psychology, 73,* 467-481.

Campion, M. A. (1989) Ability requirement implications of job design: an interdisciplinary perspective. *Personnel Psychology, 42,* 1-24.

Campion , M. A.& Thayer, P. W. (1985). Development and field evaluation of an interdisciplinary measure of job design. *Journal of Applied Psychology, 70,* 29-43.

Campion, M. A. & Thayer, P. W. (1987). Job design: Approaches, outcomes, and trade-offs. *Organizational Dynamics, 15(3),* 66-79.

Caplan , R..(1992). Disabled by design. *Interior Design*, 63, 88-91.

Carroll , J. M. (1997). Human-computer interaction: Psychology as a science of design. *Annual Review of Psychology, 48,* 61-83.

Cascio, W. F. (1998). *Managing Human Resources*. Boston: Irwin-McGraw-Hill.

Chaffin, D. B. (1997). Biomechanical Aspects of Workplace Design. In Salvendy , G. (1997) *Handbook of Human Factors and Ergonomics (2nd Ed.).* (pp. 772-789).New York: John Wiley & Sons

Clarke, S. (1999) Perceptions of organizational safety: Implications for the development of safety culture. *Journal of Organizational Behavior,* 20, 185-198.

Hoel, H., Cooper, C. L. & Faragher (2001). The experience of bullying in Great Britain: The impact of organizational status. *European Journal of Work and Organizational Psychology,* 10, 485-496.

Coyne, T. (2002). Employee kills four colleagues in Indiana. *Rocky Mountain News,* March 23, 2002, p. 11A.

Craiger , J. P. (2000). Traveling in cyberspace: Psychology of software design, Part II – usability evaluation. *The Industrial-Organizational Psychologist,* 37 (3), 101-107.

Craiger, J. P., & Weiss, R. J. (1998). Traveling in cyberspace, the final frontier: an interview with Donald Norman. *The Industrial-Organizational Psychologist*, 35(4), pp. 47-53.

Edwards, J. R., Scully, J. A., & Brtek, M. D. (2000) The nature and outcomes of work: a replication and extension of interdisciplinary work design research. *Journal of Applied Psychology*, 85, 860-868.

Edworthy, J., & Adams, A. (1996). *Warning Designs: A Research Perspective.* London: Taylor & Francis.

Frone, M. R. (1998). Predictors of work injuries among employed adolescents. *Journal of Applied Psychology, 83,* 565-576.

Goetsch, D. L. (1996). *Occupational Safety and Health (2nd Ed.).* Englewood Cliffs: Prentice Hall.

Grandjean, E. (1988). *Fitting the Task to the Man* (4th Ed.). London: Taylor & Francis.

Greenberg , L. & Barling, J. (1999). Predicting employee aggression against co-workers, subordinates, and supervisors: the roles of person behaviors and perceived workplace factors. *Journal of Organizational Behavior, 20,* 897-913.

Griffin, M. A., & Kabanoff, B. (2001) Global vision: the psychology of safety. *The Industrial and Organizational Psychologist,* 38(4), 123-127.

Lesch, M., & Hancock, P. A. (2004). Driving performance during cell-phone use. *Accident Analysis and Prevention, 36,* 471-480.

Helander , M. (1997) The Human Factors Profession. In Salvendy , G. (1997). *Handbook of Human Factors and Ergonomics (2nd Ed.).* (pp. 3-17). New York: John Wiley & Sons.

Heller, E. Chief Witness: A cellphone. *National Law Journal,* January 21, 2002, p. B10.

Hewitt Associates, LLC (1995). *Work and family benefits provided by major U.S. employers in 1994.* New York: Hewitt Associates, LLC.

Horrey, W. J., & Wickens, C. D. (2006). Examining the impact of cell phone conversations on driving using meta-analytic techniques. *Human Factors, 48*, 196-205.

Howell , W. C. (1993) Engineering psychology in a changing world. *Annual Review of Psychology, 44,* 231-263.

Hubert, A. B., & Veldhoven, M. (2001) Risk sectors for undesirable behavior and mobbing. *European Journal of Work and Organizational Psychology, 10,* 415-424.

Kohn, J. P., Friend, M. A., & Winterberger, C. A. (1996). *Fundamentals of Occupational Safety and Health.* Rockville, Md.: Government Institutes.

Landy, F. J. (1989). *The Psychology of Work Behavior.* Pacific Grove: Brooks Cole.

Laughery, K. R, Sr., & Wogalter, M. S. (1997). Warnings and Rick Perception. In Salvendy , G. (1997). *Handbook of Human Factors and Ergonomics (2nd Ed.).* (pp. 1174-1198). New York: John Wiley & Sons.

Leymann, H. (1996). The content and development of mobbing at work. *The European Journal of Work and Organizational Psychology, 5,* 165-184.

Ludwig, T. D., & Geller, E. S. (1997). Assigned versus participative goal setting and response generalization: managing injury control among professional pizza drivers. *Journal of Applied Psychology, 82,* 253-261.

McArdle, W. D., Katch, F. I., & Katch, V. L. (2001). *Exercise physiology: Energy, nutrition, and human performance (5th Edition).* Lippincott, Williams and Wilkins.

Meagher, M. (1989). Death in the delivery zone. *Inside Edition – February*. New York: American Broadcasting Company.

Mikkelsen, E. G. & Einarsem, S. (2002). *The role of victim personality in workplace bullying*. Unpublished manuscript. Psykologisk Institut, Risskov, Denmark.

Mihal, W. L., & Barrett, G. V. (1976). Individual differences in perceptual information processing and their relation to automobile accident involvement. *Journal of Applied Psychology, 61,* 229-233.

National Research Council (2001). *Musculoskeletal Disorders and the Workplace.* Washington, D.C.: National Academies press.

National Safety Council (1996). *Accident Facts.* Chicago: National Safety Council

Nell, V. (2002) Why young men drive dangerously: Implications for injury prevention. *Current Directions in Psychological Science, 11(2),* 75-78.

New York Times. (2002, Feb. 5). *Maryland: Driver in crash was using cell phone*. National Briefing, p.A21.

Noe, R. A,, Hollenbeck, J.R., Gerhart, B., & Wright, P. M. (2000). *Human Resource Management: Gaining a Competitive Advantage.* Boston: Irwin McGraw-Hill.

Perotta, T. (2002) N Y Cell phone ban passes court test. *National Law Journal*, January 21, 2002, p A4.

Performance. In Salvendy , G. *Handbook of Human Factors and Ergonomics (2nd Ed.).* (pp. 1021-1058). New York: John Wiley & Sons

U.S. Department of Transportation. (1996). *Traffic Safety CD-ROM, 1975-1994 (BTS-CD-10).* Washington, D.C.: Bureau of Transportation Statistics.

Vanderheiden, G. C. (1997). Design for people with functional limitations resulting disability, aging, and circumstance. In Salvendy , G. *Handbook of Human Factors and Ergonomics (2nd Ed.).* (pp. 1210 – 1252). New York: John Wiley & Sons

Wickens, C. D., Gordon, S. E., & Liu, Y. (1998). *An Introduction to Human Factors Engineering.* New York: Longman.

Wickens, C. D.,& Hollands, J. G. (2000). *Engineering Psychology and Human Performance (3rd Ed.).* Upper Saddle River: Prentice Hall.

Wong, E. (2002, March 20). *13-year-old girl dies after being struck by puck at NHL game*. New York Times, p. C19.

Zellars, K. J., Tepper, B. J., Duffy, M. K. (2002) Abusive supervision and subordinates' organizational citizenship behaviors. *Journal of Applied Psychology, 87,* 1068–1078.

Zohar, D. (2000). A group-level model of safey climate: Testing the effect of group climate on microaccidents in manufacturing jobs. *Journal of Applied Psychology*, 85, 587-596.

Zohar, D. (2002) Modifying supervisory practices to improve subunit safety: A leadership-based intervention model. *Journal of Applied Psychology*, 87, 156-163.