PANIMALAR INSTITUTE OF TECHNOLOGY

DEPARTMENT OF CSE

IV YEAR - VII SEMESTER

CS8079 – HUMAN COMPUTER INTERACTION (R 2017)

UNIT I FOUNDATIONS OF HCI

The Human: I/O channels – Memory – Reasoning and problem solving; The computer: Devices – Memory – processing and networks; Interaction: Models – frameworks – Ergonomics – styles – elements – interactivity- Paradigms. Case Studies

FOUNDATIONS OF HCI

The h*uman–computer interaction* is the systematic study of human performance. Ergonomists have been concerned primarily with the physical characteristics of machines and systems, and how these affect user performance. Human Factors incorporates these issues and more cognitive issues as well. Both of these disciplines are concerned with user performancein the context of any system, whether computer, mechanical or manual. As computer use became more widespread, an increasing number of researchers specialized in studying the interaction between people and computers, concerning themselves with the physical, psychological and theoretical aspects of this process. This research originally went under the name *man– machine interaction*, but this became human–*computer interaction in* recognition of the particular interest in computers and the composition of the user population.

HCI involves the design, implementation and evaluation of interactive systems in the context of the user's task and work. By ***user*** we may mean an individual user, a group of users working together, or a sequence of users in an organization, each dealing with some part of the task or process. The user is whoever is trying to get the job done using the technology. By ***computer*** we mean any technology ranging from the general desktop computer to a large- scale computer system, a process control system or an embedded system. The system may include non-computerized parts, including other people. By ***interaction*** we mean any communication between a user and computer, be it direct or indirect. **Direct interaction** involves a dialog with feedback and control throughout performance of the task. **Indirect interaction** may involve batch processing or intelligent sensors controlling the environment. The important thing is that the user is interacting with the computer in order to accomplish something.

WHO IS INVOLVED IN HCI?

The ideal designer of an interactive system would have expertise in a range of topics: psychology and cognitive science to give her knowledge of the user's perceptual, cognitive and problem-solving skills; ergonomics for the user's physical capabilities; sociology to help her under-stand the wider context of the interaction; computer science and engineering to be able to build the necessary technology; business to be able to market it; graphic design to produce an effective interface presentation; technical writing to produce the manuals, and so it goes on. There is obviously too much expertise here to be held by one person. So, we must be pragmatists rather than theorists: we want to know how to apply the theory to the problem rather than just acquire a deep understanding of the theory. Our goal, then, is to be multi-disciplinary but practical.

THEORY AND HCI

There is an underlying principle that forms the basis of the views on HCI, and claims that people use computers to accomplish work. This outlines the three major issues of concern: **the people, the computers and the tasks** that are performed. The system must support the user's task, which gives us a **fourth focus, usability**: if the system forces the user to adopt an unacceptable mode of work then it is not usable.

The word 'task' or the focus on accomplishing 'work' is also problematic when we think of areas such as domestic appliances, consumer electronics and e-commerce. There are three 'use' words that must all be true for a product to be successful; it must be:

Useful – accomplish what is required: play music, cook dinner, and format a document

Usable – do it easily and naturally, without danger of error, etc.

Used – make people want to use it, be attractive, engaging, fun, etc.

The other issues like motivation, enjoyment and experience are increasingly important.

The most impressive structures, the most beautiful buildings, the innovative and imaginative creations that provide aesthetic pleasure, all require inventive inspiration in design and a sense of artistry and in this sense the discipline is a craft. So it is for HCI, beautiful and/or novel interfaces are artistically pleasing *and* capable of fulfilling the tasks. Innovative ideas lead to more usable systems, but in order to maximize the potential benefit from the ideas, we need to understand not only that they work, but how and why they work. We have to pro-vide them with an understanding of the concepts involved, a scientific view of the reasons why certain things are successful whilst others are not, and then allow their creative nature to feed off this information: creative flow, underpinned with science; or maybe scientific method, accelerated by artistic insight. The truth is that HCI is required to be both a craft and a science in order to be successful.

1.1 The Human

1.1.1 Introduction

The human, the ***user***, is, after all, the one whom computer systems are designed to assist. The requirements of the user should therefore be our first priority.

The aspects of **cognitive psychology** have a bearing on the use of computer systems, how humans perceive the world around them, how they store and process information and solve problems, and how they physically manipulate objects.

The ***Model Human Processor***, which is a simplified view of the human processing involved in interacting with computer systems. The model comprises three subsystems: **the perceptual system**, handling sensory stimulus from the outside world, **the motor system**, which controls actions, and **the cognitive system**, which provides the processing needed to connect the two. Each of these subsystems has its own processor and memory, although obviously the complexity of these varies depending on the complexity of the tasks the subsystem has to perform. The model also includes a number of *principles of operation* which dictate the behavior of the systems under certain conditions.

In conventional computer system, the three components are input–output, memory and processing. In the human, dealing is with an intelligent information-processing system, and processing therefore includes problem solving, learning, and, consequently, making mistakes. The human, unlike the computer, is also influenced by external factors such as the social and organizational environment, and we need to be aware of these influences as well. These factors are ignored for now and concentration is on the human's information processing capabilities only.

1.1.2 Input– Output Channels

A person's interaction with the outside world occurs through information being received and sent: input and output. In an interaction with a computer the user receives information that is output by the computer, and responds by providing input to the computer – the user's output becomes the computer's input and vice versa.

Input in the human occurs mainly through the senses and output through the motor control of the effectors. There are five major senses: sight, hearing, touch, taste and smell. Of these, the first three are the most important to HCI. Taste and smell do not currently play a significant role in HCI.

Similarly there are a number of effectors, including the limbs, fingers, eyes, head and vocal system. In the interaction with the computer, the fingers play the primary role, through typing or mouse control, with some use of voice, and eye, head and body position.

1.1.2.1 Vision

Human vision is a highly complex activity with a range of physical and perceptual limitations, yet it is the primary source of information for the average person. We can roughly divide visual perception into two stages: the physical reception of the stimulus from the outside world, and the processing and interpretation of that stimulus. On the one hand the physical properties of the eye and the visual system mean that there are certain things that cannot be seen by the human; on the other the interpretative capabilities of visual processing allow images tobe constructed from incomplete information

The human eye

The eye is a mechanism for receiving light and transforming it into electrical energy. Light is reflected from objects in the world and their image is focused upside down on the back of the eye. The receptors in the eye transform it into electrical signals which are passed to the brain.

The eye has a number of important components (Figure 1.1). The *cornea* and *lens* at the front of the eye focus the light into a sharp image on the back of the eye, the *retina*. The retina is light sensitive and contains two types of *photoreceptor*: *rods* and *cones*.

Rods are highly sensitive to light and therefore allow us to see under a low level of illumination. However, they are unable to resolve fine detail and are subject to light saturation. This is the reason for the temporary blindness we get when moving from a darkened room into sunlight: the rods have been active and are saturated by the sudden light. The cones do not operate either as they are suppressed by the rods. We are therefore temporarily unable to see at all. There are approximately 120 million rods per eye which are mainly situated towards the edges of the retina. Rods therefore dominate peripheral vision.

Cones are the second type of receptor in the eye. They are less sensitive to light than the rods and can therefore tolerate more light. There are three types of cone, each sensitive to a different wavelength of light. This allows color vision. The eye has approximately 6 million cones, mainly concentrated on the *fovea*, a small area of the retina on which images are fixated.

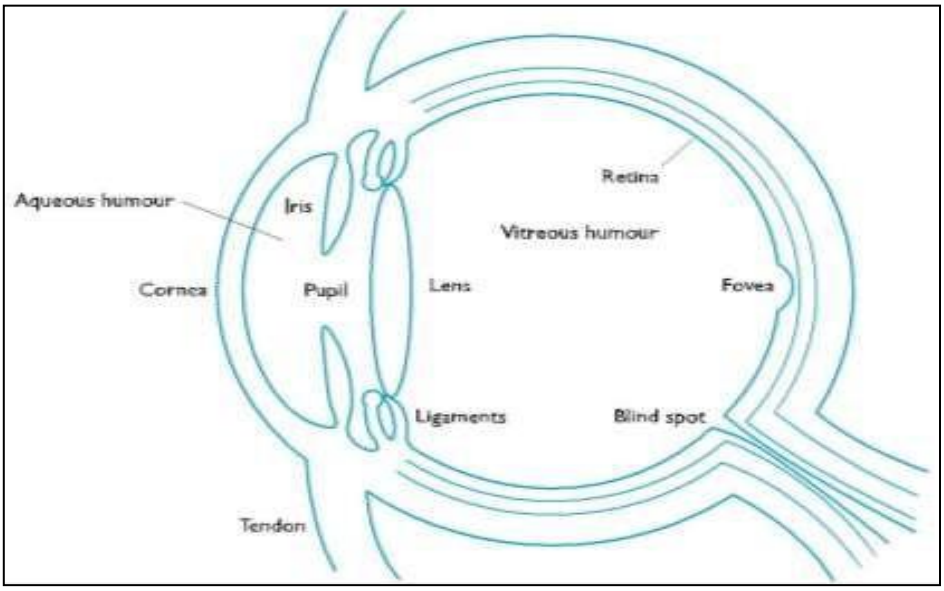


Figure 1.1: Parts of Eye

Although the retina is mainly covered with photoreceptors there is one *blind spot* where the optic nerve enters the eye. The blind spot has no rods or cones, yet our visual system compensates for this so that in normal circumstances we are unaware of it.

The retina also has specialized nerve cells called *ganglion cells*. There are two types: X-cells, which are concentrated in the fovea and are responsible for the early detection of pattern; and Y-cells which are more widely distributed in the retina and are responsible for the early detection of movement. The distribution of these cells means that, while we may not be able to detect changes in pattern in peripheral vision, we can perceive movement

Visual perception

The information received by the visual apparatus must be filtered and passed to processing elements which allow us to recognize coherent scenes, disambiguate relative distances and differentiate color. The factors that are crucial to the design of effective visual interfaces are how we perceive size and depth, brightness and color.

Perceiving size and depth Imagine you are standing on a hilltop. Describing such a scene the notions of size and distance predominate. To understand how the eye perceives size, depth and relative distances, we must consider how the image appears on the retina. Reflected light from the object forms an upside-down image on the retina. The size of that image is specified as a *visual angle*. Figure 1.2 illustrates how the visual angle is calculated.

If we were to draw a line from the top of the object to a central point on the front of the eye and a second line from the bottom of the object to the same point, the visual angle of the object is the angle between these two lines. Visual angle is affected by both the size of the object and its distance from the eye. Therefore if two objects are at the same distance, the larger one will have the larger visual angle. Similarly, if two objects of the same size are placed at different distances from the eye, the furthest one will have the smaller visual angle. The visual angle indicates how much of the field of view is taken by the object. The visual angle measurement is given in either degrees or *minutes of arc*, where 1 degree is equivalent to 60 minutes of arc, and 1 minute of arc to 60 seconds of arc.

So, an object's visual angle affect our perception of its size. First, if the visual angle of an object is too small we will be unable to perceive it at all. ***Visual acuity*** is the ability of a person to perceive fine detail. A number of measurements have been established to test visual acuity, most of which are included in standard eye tests. For example, a person with normal vision can detect a single line if it has a visual angle of 0.5 seconds of arc. Spaces between lines can be detected at 30 seconds to 1 minute of visual arc. These represent the limits of human visual acuity.

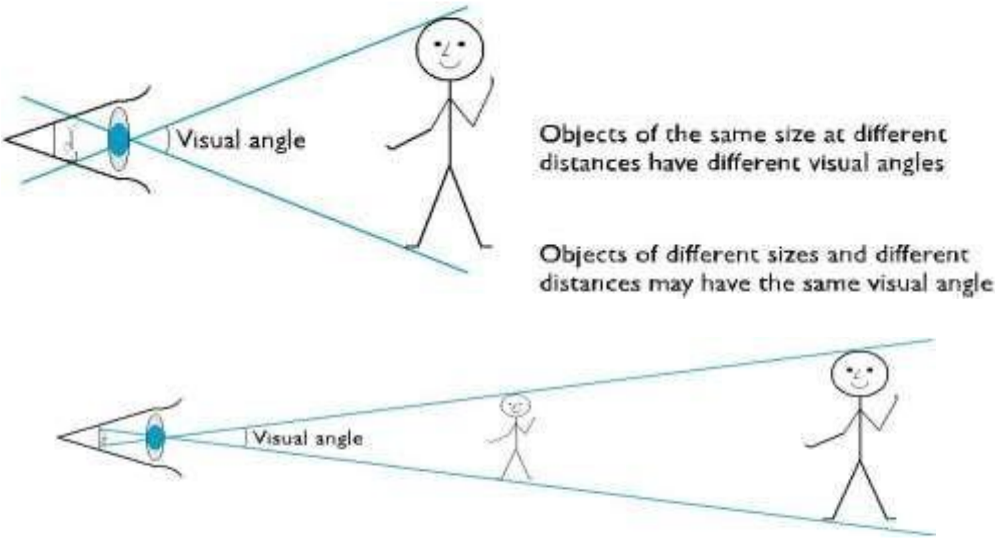


Figure 1.2: Visual Angle

Our perception of an object's size remains constant even if its visual angle changes. So a person's height is perceived as constant even if they move further from you. This is the ***law of size constancy***, and it indicates that our perception of size relies on factors other than the visual angle.

One of these factors is our perception of depth. If we return to the hilltop scene there are a number of *cues* which we can use to determine the relative positions and distances of the objects which we see. If objects overlap, the object which is partially covered is perceived to be in the background, and therefore further away. Similarly, the size and height of the object in our field of view provides a cue to its distance.

Perceiving brightness:

A second aspect of visual perception is the perception of *brightness*. Brightness is in fact a subjective reaction to levels of light. It is affected by *luminance* which is the amount of light emitted by an object. The luminance of an object is dependent on the amount of light falling on the object's surface and its reflective properties. Luminance is a physical characteristic and can be measured using a *photometer*. *Contrast* is related to luminance: it is a function of the luminance of an object and the luminance of its background

The visual system itself also compensates for changes in brightness. In dim lighting, the rods predominate vision. Since there are fewer rods on the fovea, objects in low lighting can be seen less easily when fixated upon, and are more visible in peripheral vision. In normal lighting, the cones take over.

Perceiving color*:*

A third factor that we need to consider is perception of color. Color is usually regarded as being made up of **three** components: **hue, intensity and saturation**. Hue is determined by the spectral wavelength of the light. Blues have short wavelengths, greens medium and reds long. Intensity is the brightness of the color, and saturation is the amount of whiteness in the color. By varying these two, we can perceive in the region of 7 million different colors.

The eye perceives color because the cones are sensitive to light of different wave-lengths. There are **three different types of cone**, each sensitive to a different color (blue, green and red). Color vision is best in the fovea, and worst at the periphery where rods predominate.

The capabilities and limitations of visual processing:

Visual processing compensates for the movement of the image on the retina which occurs as we move around and as the object which we see moves. Although the retinal image is moving, the image that we perceive is stable. Similarly, color and brightness of objects are perceived as constant, in spite of changes in luminance.

This ability to interpret and exploit our expectations can be used to resolve ambiguity.

Figure 1.3: Ambiguous shape? **Figure 1.4: Alphabets** **Figure 1.5: Numerals**

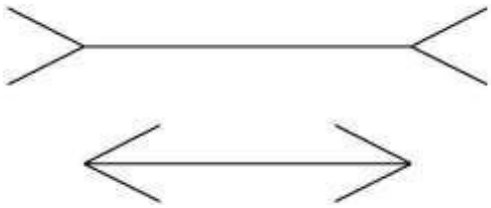
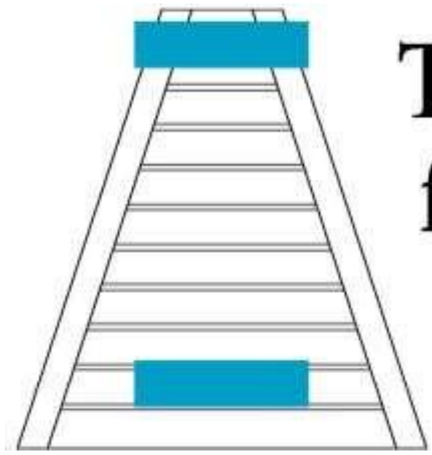


Figure 1.6: The Muller–Lyer illusion – which line is longer?

Consider Figure 1.6. Which line is longer? Most people when presented with this will say that the top line is longer than the bottom. In fact, the two lines are the same length.

A similar illusion is the Ponzo illusion (Figure 1.7). Here the top line appears longer, owing to the distance effect, although both lines are the same length. These illusions demonstrate that our perception of size is not completely reliable.

Another illusion created by our expectations compensating an image is the proof-reading illusion. Read the text in Figure 1.8 quickly. What does it say? Most people reading this rapidly will read it correctly, although closer inspection shows that the word 'the' is repeated in the second and third line.



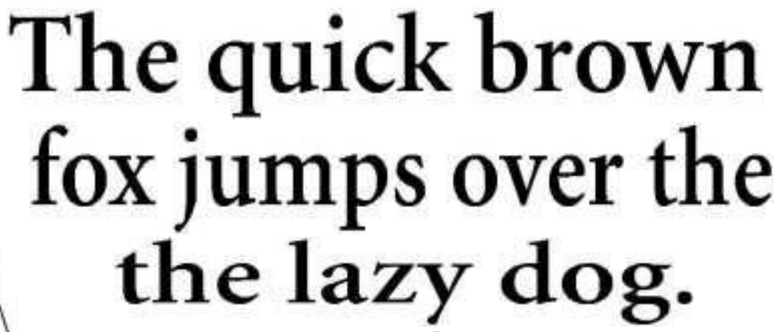


Figure 1.8 Is this text correct?

Figure 1.7:The Ponzo illusion – are these the same size?

Reading:

During reading, the eye makes jerky movements called *saccades* followed by fixations. Perception occurs during the fixation periods, which account for approximately 94% of the time elapsed. The eye moves backwards over the text as well as forwards, in what are known as *regressions*. If the text is complex there will be more regressions. Adults read approximately 250 words a minute. It is unlikely that words are scanned serially, character by character, since experiments have shown that words can be recognized as quickly as single characters. Instead, familiar words are recognized using word shape.

1.1.2.2 Hearing

The sense of hearing is often considered secondary to sight. As I sit at my desk I can hear cars passing on the road outside, machinery working on a site nearby, the drone of a plane overhead and bird song. But I can also tell *where* the sounds are coming from, and estimate how far away they are. So from the sounds I hear I can tell that a car is passing on a particular road near my house, and which direction it is traveling in.

The human ear

Hearing begins with vibrations in the air or *sound waves*. The ear receives these vibrations and transmits them, through various stages, to the auditory nerves. The ear comprises three sections, commonly known as the ***outer ear*, *middle ear* and *inner ear***

The **outer ear** is the visible part of the ear. It has two parts: the ***pinna*,** which is the structure that is attached to the sides of the head, and the ***auditory canal***, along which sound waves are passed to the middle ear. The outer ear serves two purposes. First, it protects the sensitive middle ear from damage. The auditory canal contains wax which prevents dust, dirt and over-inquisitive insects reaching the middle ear. It also maintains the middle ear at a constant temperature. Secondly, the pinna and auditory canal serve to amplify some sounds.

The **middle ear** is a small cavity connected to the outer ear by the *tympanic membrane*, or ear drum, and to the inner ear by the *cochlea*. Within the cavity are the *ossicles*, the smallest bones in the body. Sound waves pass along the auditory canal and vibrate the ear drum which in turn vibrates the ossicles, which transmit the vibrations to the cochlea, and so into the inner ear. This 'relay' is required because, unlike the air-filled outer and middle ears, the inner ear is filled with a denser cochlean liquid. If passed directly from the air to the liquid, the transmission of the sound waves would be poor. By transmitting them via the ossicles the sound waves are concentrated and amplified.

The waves are passed into the liquid-filled cochlea in the inner ear. Within the cochlea are delicate hair cells or cilia that bend because of the vibrations in the cochlean liquid and release a chemical transmitter which causes impulses in the auditory nerve.

Processing Sound

Sound is changes or vibrations in air pressure. It has a number of characteristics which we can differentiate. ***Pitch*** is the frequency of the sound. A low frequency produces a low pitch, a high frequency, a high pitch. ***Loudness*** is proportional to the amplitude of the sound; the frequency remains constant. ***Timbre*** relates to the type of the sound: sounds may have the same pitch and loudness but be made by different instruments and so vary in timbre. The human ear can hear frequencies from about 20 Hz to 15 kHz. The auditory system performs some filtering of the sounds received, allowing us to ignore background noise and concentrate on important information.

1.1.2.3 Touch

The third and last of the senses is touch or ***haptic perception***. Touch provides us with vital information about our environment. It tells us when we touch something hot or cold, and can therefore act as a warning. It also provides us with feedback when we attempt to lift an object, for example. Touch is therefore an important means of feedback, and this is no less so in using computer systems. Feeling buttons depress is an important part of the task of pressing the button. Also, we should be aware that, although for the average person, haptic perception is a secondary source of information, for those whose other senses are impaired, it may be vitally important. For such users, interfaces such as braille may be the primary source of information in the interaction. We should not therefore underestimate the importance of touch.

We receive stimuli through the skin. The skin contains three types of sensory receptor: ***thermoreceptors*** respond to heat and cold, ***nociceptors*** respond to intense pressure, heat and pain, and ***mechanoreceptors*** respond to pressure. It is the last of these that we are concerned with in relation to human–computer interaction.

There are two kinds of mechanoreceptor, which respond to different types of pressure. ***Rapidly adapting mechanoreceptors*** respond to immediate pressure as the skin is indented. These receptors also react more quickly with increased pressure. However, they stop responding if continuous pressure is applied. ***Slowly adapting mechanoreceptors*** respond to continuously applied pressure.

Although the whole of the body contains such receptors, some areas have greater sensitivity or acuity than others. It is possible to measure the acuity of different areas of the body using the ***two-point threshold test***. Take two pencils, held so their tips are about 12 mm apart. Touch the points to your thumb and see if you can feel two points. If you cannot, move the points a little further apart. When you can feel two points, measure the distance between them. The greater the distance, the lower the sensitivity. You can repeat this test on different parts of your body. You should find that the measure on the forearm is around 10 times that of the finger or thumb. The fingers and thumbs have the highest acuity.

A second aspect of haptic perception is ***kinesthesis***: awareness of the position of the body and limbs. This is due to receptors in the joints. Again there are three types: rapidly adapting, which respond when a limb is moved in a particular direction; slowly adapting, which respond to both movement and static position; and positional receptors, which only respond when a limb is in a static position. This perception affects both comfort and performance. For example, for a touch typist, awareness of the relative positions of the fingers and feedback from the keyboard are very important.

1.1.2.4 Movement

The stimulus is received through the sensory receptors and transmitted to the brain. The question is processed and a valid response generated. The brain then tells the appropriate muscles to respond. Each of these stages takes time, which can be roughly divided into reaction time and movement time. Movement time is dependent largely on the physical characteristics of the subjects: their age and fitness, for example. Reaction time varies according to the sensory channel through which the stimulus is received.

A second measure of motor skill is accuracy. One question that we should ask is whether speed of reaction results in reduced accuracy. This is dependent on the task and the user. In some cases, requiring increased reaction time reduces accuracy. This is the premise behind many arcade and video games where less skilled users fail at levels of play that require faster responses. However, for skilled operators this is not necessarily the case. Studies of keyboard operators have shown that, although the faster operators were up to twice as fast as the others, the slower ones made 10 times the errors.

Speed and accuracy of movement are important considerations in the design of interactive systems, primarily in terms of the time taken to move to a particular target on a screen. The target may be a button, a menu item or an icon, for example. The time taken to hit a target is a function of the size of the target and the distance that has to be moved. This is formalized in *Fitts' law*. There are many variations of this formula, which have varying constants, but they are all very similar. One common form is

Movement time = $a + b \log_2(\text{distance}/\text{size} + 1)$
where a and b are empirically determined constants.

This affects the type of target we design. Since users will find it more difficult to manipulate small objects, targets should generally be as large as possible and the distance to be moved as small as possible. This has led to suggestions that pie-chart-shaped menus are preferable to lists since all options are equidistant. However, the trade-off is increased use of screen estate, so the choice may not be so simple.

1.3 MEMORY

- Much of our everyday activity relies on memory. As well as storing all our factual knowledge, our memory contains our knowledge of actions or procedures. It allows us to repeat actions, to use language, and to use new information received via our senses. It also gives us our sense of identity, by preserving information from our past experiences.
- Memory is the second part of our model of the human as an information-processing system. It is generally agreed that there are three types of memory or memory function: sensory buffers, short-term memory or working memory, and long-term memory. There is some disagreement as to whether these are three separate systems or different functions of the same system as shown in figure 1.8.

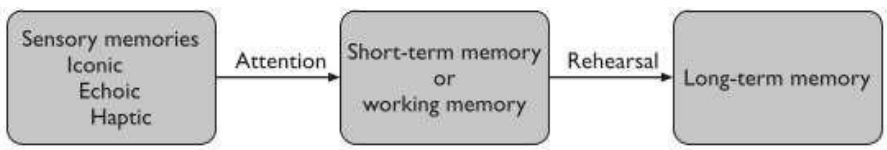


Figure 1.8: A model of the structure of memory

a. Sensory memory:

- The sensory memories act as buffers for stimuli received through the senses. A sensory memory exists for each sensory channel: iconic memory for visual stimuli, echoic memory for aural stimuli and haptic memory for touch. These memories are constantly overwritten by new information coming in on these channels.
- We can demonstrate the existence of iconic memory by moving a finger in front of the eye. Can you see it in more than one place at once? This indicates a persistence of the image after the stimulus has been removed. A similar effect is noticed most vividly at firework displays where moving sparklers leave a persistent image. Information remains in iconic memory very briefly, in the order of 0.5 seconds.
- Similarly, the existence of echoic memory is evidenced by our ability to ascertain the direction from which a sound originates. This is due to information being received by both ears. However, since this information is received at different times, we must store the stimulus in the meantime. Echoic memory allows brief 'play-back' of information.

- Information is passed from sensory memory into short-term memory by attention, thereby filtering the stimuli to only those which are of interest at a given time. Attention is the concentration of the mind on one out of a number of competing stimuli or thoughts. It is clear that we are able to focus our attention selectively, choosing to attend to one thing rather than another. This is due to the limited capacity of our sensory and mental processes. If we did not selectively attend to the stimuli coming into our senses, we would be overloaded. We can choose which stimuli to attend to, and this choice is governed to an extent by our arousal, our level of interest or need.

b. Short-term memory:

- Short-term memory or working memory acts as a 'scratch-pad' for temporary recall of information. It is used to store information which is only required fleetingly. For example, calculate the multiplication 35 × 6 in your head. The chances are that you will have done this calculation in stages, perhaps 5 × 6 and then 30 × 6 and added the results; or you may have used the fact that 6 = 2 × 3 and calculated 2 × 35 = 70 followed by 3 × 70.

- To perform calculations such as this we need to store the intermediate stages for use later. For this task use short-term memory. Short-term memory can be accessed rapidly, in the order of 70 ms. However, it also decays rapidly, meaning that information can only be held there temporarily, in the order of 200 ms. Short-term memory also has a limited capacity.

- There are two basic methods for measuring memory capacity. The first involves determining the length of a sequence which can be remembered in order. The second allows items to be freely recalled in any order. Using the first measure, the average person can remember 7 ± 2 digits. A generalization of the 7 ± 2 rule is that we can remember 7 ± 2 chunks of information.

- Therefore chunking information can increase the short-term memory capacity. The limited capacity of short-term memory produces a subconscious desire to create chunks, and so optimize the use of the memory. The successful formation of a chunk is known as closure.

- In experiments where subjects were able to recall words freely, evidence shows that recall of the last words presented is better than recall of those in the middle, this is known as the recency effect. However, if the subject is asked to perform another task between presentation and recall (for example, counting backwards) the recency effect is eliminated. The recall of the other words is unaffected. This suggests that short-term memory recall is damaged by interference of other information. (refer Fig.1.9)

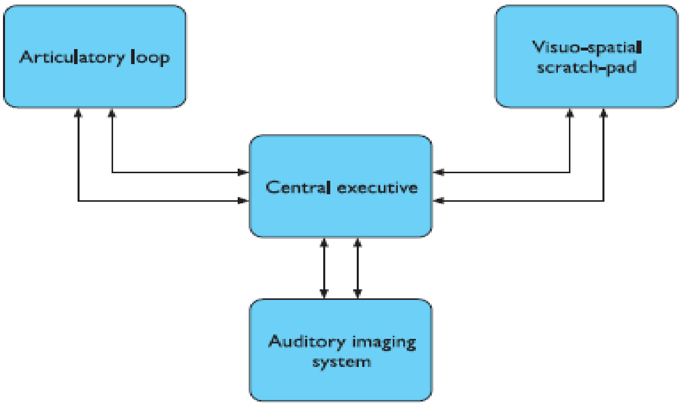


Fig 1.9: **A more detailed model of short-term memory**

c. Long-term memory:

- Short-term memory is our working memory or 'scratch-pad', long-term memory is our main resource. Here we store factual information, experiential knowledge, procedural rules of behavior – in fact, everything that we 'know'.
- It differs from short-term memory in a number of significant ways. First, it has a huge, if not unlimited, capacity. Secondly, it has a relatively slow access time of approximately a tenth of a second. Thirdly, forgetting occurs more slowly in long-term memory, if at all. These distinctions provide further evidence of a memory structure with several parts.
- Long-term memory is intended for the long-term storage of information. Information is placed there from working memory through rehearsal. Unlike working memory there is little decay: long-term recall after minutes is the same as that after hours or days.

Long-term memory structure:

- There are two types of long-term memory: episodic memory and semantic memory. Episodic memory represents our memory of events and experiences in a serial form. It is from this memory that we can reconstruct the actual events that took place at a given point in our lives.
- Semantic memory, on the other hand, is a structured record of facts, concepts and skills that we have acquired. The information in semantic memory is derived from that in our episodic memory, such that we can learn new facts or concepts from our experiences.

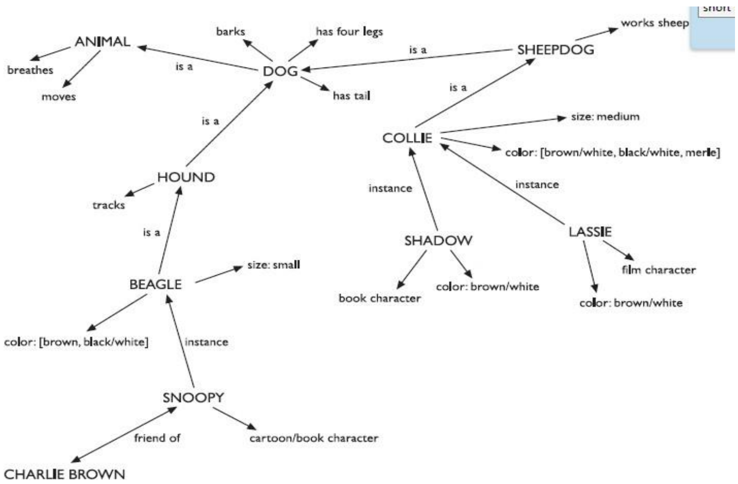


Fig 1.10: **Long-term memory may store information in a semantic network**

- Semantic memory is structured in some way to allow access to information, representation of relationships between pieces of information, and inference. One model for the way in which semantic memory is structured is as a network. Items are associated to each other in classes, and may inherit attributes from parent classes. This model is known as a semantic network. As an example, our knowledge about dogs may be stored in a network such as that shown in figure 1.11.
- Specific breed attributes may be stored with each given breed, yet general dog information is stored at a higher level. This allows us to generalize about specific cases. For instance, we may not have been told that the sheepdog Shadow has four legs and a tail, but we can infer this information from our general knowledge about sheepdogs and dogs in general. Note also that there are connections within the network which link into other domains of knowledge, for example cartoon characters.
- A number of other memory structures have been proposed to explain how we represent and store different types of knowledge. Each of these represents a different aspect of knowledge and, as such, the models can be viewed as complementary rather than mutually exclusive.

Figure1.11: A frame based representation of knowledge

- Semantic networks represent the associations and relationships between single items in memory. However, they do not allow us to model the representation of more complex objects or events, which are perhaps composed of a number of items or activities. Structured representations such as frames and scripts organize information into data structures. Slots in these structures allow attribute values to be added. Frame slots may contain default, fixed or variable information.
- A frame is instantiated when the slots are filled with appropriate values. Frames and scripts can be linked together in networks to represent hierarchical structured knowledge. Returning to the 'dog' domain, a frame-based representation of the knowledge may look something as said above.
- The fixed slots are those for which the attribute value is set, default slots represent the usual attribute value, although this may be overridden in particular instantiations and variable slots can be filled with particular values in a given instance. Slots can also contain procedural knowledge.
- Frames extend semantic nets to include structured, hierarchical information. They represent knowledge items in a way which makes explicit the relative importance of each piece of information. Scripts attempt to model the representation of stereotypical knowledge about situations. Consider the following sentence: John took his dog to the surgery.

Script for a visit to the vet

Entry conditions:	*dog ill* *vet open* *owner has money*	Roles:	*vet examines* *diagnoses* *treats* *owner brings dog in* *pays* *takes dog out*
Result:	*dog better* *owner poorer* *vet richer*	Scenes:	*arriving at reception* *waiting in room* *examination* *paying*
Props:	*examination table* *medicine* *instruments*	Tracks:	*dog needs medicine* *dog needs operation*

Figure1.12: Script Example

- After seeing the vet, he left. From our knowledge of the activities of dog owners and vets, we may fill in a substantial amount of detail. The animal was ill. The vet examined and treated the animal. John paid for the treatment before leaving. We are less likely to assume the alternative reading of the sentence, that John took an instant dislike to the vet on sight and did not stay long enough to talk to him!
- A script represents this default or stereotypical information, allowing us to interpret partial descriptions or cues fully. A script comprises a number of elements, which, like slots, can be filled with appropriate information:
  - Entry conditions: Conditions that must be satisfied for the script to be activated.
  - Result: Conditions that will be true after the script is terminated.
  - Props: Objects involved in the events described in the script.
  - Roles: Actions performed by particular participants.
  - Scenes: The sequences of events that occur.
  - Tracks: A variation on the general pattern representing an alternative scenario.

Long-term memory processes:

- There are three main activities related to long-term memory: storage or remembering of information, forgetting and information retrieval. Information from short-term memory is stored in long-term memory by rehearsal. The repeated exposure to a stimulus or the rehearsal of a piece of information transfers it into long-term memory.
- Sentences are easier still to memorize. To retell the story replacing unfamiliar words and concepts with words which were meaningful was easier.
- Stories were effectively translated into the subject's own culture. This is related to the semantic structuring of long-term memory: if information is meaningful and familiar, it can be related to existing structures and more easily incorporated into memory.
- So if structure, familiarity and concreteness help us in learning information, what causes us to lose this information, to forget? There are two main theories of forget- ting: decay and interference. The first theory suggests that the information held in long-term memory may eventually be forgotten.
- From an experiments with nonsense syllables that information in memory decayed logarithmically, that is that it was lost rapidly to begin with, and then more slowly. Jost's law, which follows from this, states that if two memory traces are equally strong at a given time the older one will be more durable.

- The second theory is that information is lost from memory through interference. If we acquire new information it causes the loss of old information. This is termed retroactive interference.
- A common example of this is the fact that if you change telephone numbers, learning your new number makes it more difficult to remember your old number. This is because the new association masks the old. However, sometimes the old memory trace breaks through and interferes with new information.
- This is called proactive inhibition. An example of this is when you find yourself driving to your old house rather than your new one. Forgetting is also affected by emotional factors.
- In experiments, subjects given emotive words and non-emotive words found the former harder to remember in the short term but easier in the long term. Indeed, this observation tallies with our experience of selective memory. We tend to remember positive information rather than negative (hence nostalgia for the 'good old days'), and highly emotive events rather than mundane.

1.4 REASONING AND PROBLEM SOLVING

- Humans, on the other hand, are able to use information to reason and solve problems, and indeed do these activities when the information is partial or unavailable. Human thought is conscious and self-aware: while we may not always be able to identify the processes we use, we can identify the products of these processes, our thoughts. Thinking can require different amounts of knowledge.

1.4.1 Reasoning: Reasoning is the process by which we use the knowledge we have to draw conclusions or infer something new about the domain of interest. There are a number of different types of reasoning: deductive, inductive and abductive. We use each of these types of reasoning in everyday life, but they differ in significant ways.

Deductive reasoning: Deductive reasoning derives the logically necessary conclusion from the given premises.
Inductive reasoning: Induction is generalizing from cases we have seen to infer information about cases we have not seen.

Abductive reasoning: The third type of reasoning is abduction. Abduction reasons from a fact to the action or state that caused it. This is the method we use to derive explanations for the events we observe.

1.4.2 Problem solving:

- If reasoning is a means of inferring new information from what is already known, problem solving is the process of finding a solution to an unfamiliar task, using the knowledge we have. Human problem solving is characterized by the ability to adapt the information we have to deal with new situations. However, often solutions seem to be original and creative.
- There are a number of different views of how people solve problems. The earliest, dating back to the first half of the twentieth century, is the Gestalt view that problem solving involves both reuse of knowledge and insight. This has been largely superseded but the questions it was trying to address remain and its influence can be seen in later research.

- A second major theory, proposed in the 1970s by Newell and Simon, was the problem space theory, which takes the view that the mind is a limited information processor. Later variations on this drew on the earlier theory and attempted to reinterpret Gestalt theory in terms of information processing theories. We will look briefly at each of these views.

Gestalt theory:

- Gestalt psychologists were answering the claim, made by behaviorists, that problem solving is a matter of reproducing known responses or trial and error. This explanation was considered by the Gestalt school to be insufficient to account for human problem-solving behavior. Instead, they claimed, problem solving is both productive and reproductive.
- Reproductive problem solving draws on previous experience as the behaviorists claimed, but productive problem solving involves insight and restructuring of the problem. Indeed, reproductive problem solving could be a hindrance to finding a solution, since a person may 'fixate' on the known aspects of the problem and so be unable to see novel interpretations that might lead to a solution.
- Gestalt psychologists backed up their claims with experimental evidence.Problem space theory Newell and Simon proposed that problem solving centers on the problem space. The problem space comprises problem states, and problem solving involves generating these states using legal state transition operators. The problem has an initial state and a goal state and people use the operators to move from the former to the latter.
- Such problem spaces may be huge, and so heuristics are employed to select appropriate operators to reach the goal. One such heuristic is means–ends analysis. In means–ends analysis the initial state is compared with the goal state and an operator chosen to reduce the difference between the two.
- For example, imagine you are reorganizing your office and you want to move your desk from the north wall of the room to the window. Your initial state is that the desk is at the north wall. The goal state is that the desk is by the window. The main difference between these two is the location of your desk.
- You have a number of operators which you can apply to moving things: you can carry them or push them or drag them, etc. However, you know that to carry something it must be light and that your desk is heavy. You therefore have a new subgoal: to make the desk light. Your operators for this may involve removing drawers, and so on.

Analogy in problem solving:

- A third element of problem solving is the use of analogy. Here we are interested in how people solve novel problems. One suggestion is that this is done by mapping knowledge relating to a similar known domain to the new problem – called analogical mapping. Similarities between the known domain and the new one are noted and operators from the known domain are transferred to the new one. This process has been investigated using analogous stories.

1.4.3 Skill acquisition:

- All of the problem solving that we have considered so far has concentrated on handling unfamiliar problems. However, for much of the time, the problems that we face are not completely new. Instead, we gradually acquire skill in a particular domain area.

1.4.4 Errors and mental models:

- Human capability for interpreting and manipulating information is quite impressive. However, we do make mistakes. Some are trivial, resulting in no more than temporary inconvenience or annoyance. Others may be more serious, requiring substantial effort to correct. Occasionally an error may have catastrophic effects, as we see when ‘human error’ results in a plane crash or nuclear plant leak.
- There are several different types of error. If a pattern of behavior has become automatic and we change some aspect of it, the more familiar pattern may break through and cause an error. Other errors result from an incorrect understanding, or model, of a situation or system.
- People build their own theories to understand the causal behavior of systems. These have been termed mental models. They have a number of characteristics. Mental models are often partial: the person does not have a full understanding of the working of the whole system. They are unstable and are subject to change.

1.5 THE COMPUTER

- There is the computer ‘box’ itself, a keyboard, a mouse and a color screen. Some of this variation is driven by different hardware configurations: desktop use, laptop computers, PDAs (personal digital assistants).
- Partly the diversity of devices reflects the fact that there are many different types of data that may have to be entered into and obtained from a system, and there are also many different types of user, each with their own unique requirements. A computer system comprises various elements, each of which affects the user of the system.
- Input devices for interactive use, allowing text entry, drawing and selection from the screen:
    a. text entry: traditional keyboard, phone text entry, speech and handwriting
    b. pointing: principally the mouse, but also touchpad, stylus, and others
    c. 3D interaction devices
- Output display devices for interactive use:
    a. different types of screen mostly using some form of bitmap display
    b. large displays and situated displays for shared and public use
    c. digital paper may be usable in the near future
- Virtual reality systems and 3D visualization have special interaction and display devices. Various devices in the physical world:
    - physical controls and dedicated displays
    - sound, smell and haptic feedback
    - sensors for nearly everything including movement, temperature, bio-signs
- Paper output and input: the paperless office and the less-paper office:
    - different types of printers and their characteristics, character styles and fonts
    - scanners and optical character recognition
- Memory:
    - short-term memory: RAM
    - long-term memory: magnetic and optical disks
    - capacity limitations related to document and video storage
    - access methods as they limit or help the user

- Processing:
  - the effects when systems run too slow or too fast, the myth of the infinitely fast machine
  - limitations on processing speed
  - networks and their impact on system performance.

1.6 DEVICES

Input devices for interactive use, allowing text entry, drawing and selection from the screen.

1.6.1 TEXT ENTRY DEVICES: The alphanumeric keyboard:

- The keyboard is still one of the most common input devices in use today. It is used for entering textual data and commands. The QWERTY keyboard: The layout of the digits and letters on a QWERTY keyboard is fixed, but non-alphanumeric keys vary between keyboards. The standard layout is also subject to variation in the placement of brackets, backslashes and such like.

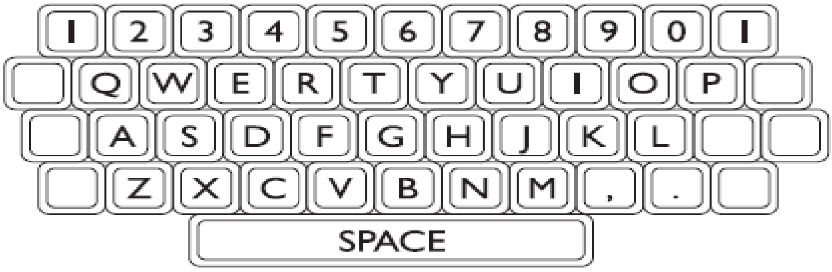


Fig.1.13 QWERTY Keyboard

- Ease of learning – alphabetic keyboard: One of the most obvious layouts to be produced is the alphabetic keyboard, in which the letters are arranged alphabetically across the keyboard. It might be expected that such a layout would make it quicker for untrained typists to use.
- Ergonomics of use – DVORAK keyboard and split designs: The DVORAK keyboard uses a similar layout of keys to the QWERTY system, but assigns the letters to different keys. Based upon an analysis of typing, the keyboard is designed to help people reach faster typing speeds. It is biased towards right-handed people, in that 56% of keystrokes are made with the right hand. The layout of the keys also attempts to ensure that the majority of keystrokes alternate between hands, thereby increasing the potential speed.

Chord keyboards:

- Chord keyboards are significantly different from normal alphanumeric keyboards. Only a few keys, four or five, are used and letters are produced by pressing one or more of the keys at once.
- For example, in the Microwriter, the pattern of multiple keypresses is chosen to reflect the actual letter shape.Such keyboards have a number of advantages. They are extremely compact: simply reducing the size of a conventional keyboard makes the keys too small and close together, with a correspondingly large increase in the difficulty of using it.

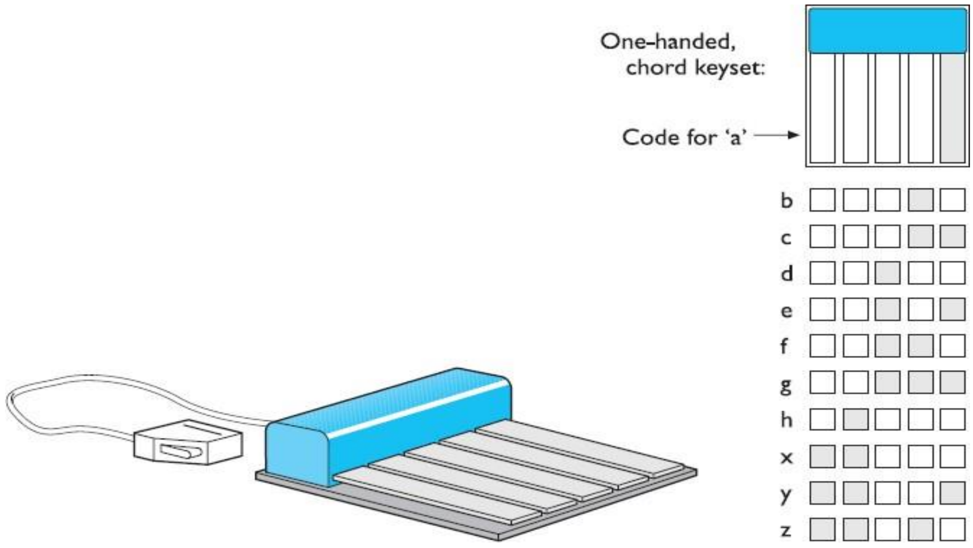


Fig.1.14 : A very early chord keyboard (left) and its lettercodes (right)

Phone pad and T9 entry:

- With mobile phones being used for SMS text messaging and WAP, the phone keypad has become an important form of text input. Unfortunately a phone only has digits 0–9, not a full alphanumeric keyboard. To overcome this for text input the numeric keys are usually pressed several times shows a typical mapping of digits to letters.
- The main number-to-letter mapping is standard, but punctuation and accented letters differ between phones. Also there needs to be a way for the phone to distinguish, say, the 'dd' from 'e'. On some phones you need to pause for a short period between successive letters using the same key, for others you press an additional key.
- Most phones have at least two modes for the numeric buttons: one where the keys mean the digits and one where they mean letters. Some have additional modes to make entering accented characters easier. Also a special mode or setting is needed for capital letters although many phones use rules to reduce this.

Handwriting recognition:

- Handwriting is a common and familiar activity, and is therefore attractive as a method of text entry. If we were able to write as we would when we use paper, but with the computer taking this form of input and converting it to text, we can see that it is an intuitive and simple way of interacting with the computer.

Speech recognition:

- Speech recognition is a promising area of text entry, but it has been promising for a number of years and is still only used in very limited situations. There is a natural enthusiasm for being able to talk to the machine and have it respond to commands, since this form of interaction is one with which we are very familiar.
- Successful recognition rates of over 97% have been reported, but since this represents one letter in error in approximately every 30, or one spelling mistake every six or so words, this is stoll unacceptible (sic)! Note also that this performance is usually quoted only for a restricted vocabulary of command words.

- Trying to extend such systems to the level of understanding natural language, with its inherent vagueness, imprecision and pauses, opens up many more problems that have not been satisfactorily solved even for keyboard-entered natural language.

1.6.2 Positioning, Pointing And Drawing

The mouse:

- The mouse has become a major component of the majority of desktop computer systems sold today, and is the little box with the tail connecting it to the machine in our basic computer system.
- It is a small, palm-sized box housing a weighted ball – as the box is moved over the tabletop, the ball is rolled by the table and so rotates inside the housing. This rotation is detected by small rollers that are in contact with the ball, and these adjust the values of potentiometers.
- If you remove the ball occasionally to clear dust you may be able to see these rollers. The changing values of these potentiometers can be directly related to changes in position of the ball. The potentiometers are aligned in different directions so that they can detect both horizontal and vertical motion.
- The relative motion information is passed to the computer via a wire attached to the box, or in some cases using wireless or infrared, and moves a pointer on the screen, called the cursor. The whole arrangement tends to look rodent-like, with the box acting as the body and the wire as the tail; hence the term 'mouse'.

Touchpad:

- Touchpads are touch-sensitive tablets usually around 2–3 inches (50–75 mm) square. They were first used extensively in Apple Powerbook portable computers but are now used in many other notebook computers and can be obtained separately to replace the mouse on the desktop. They are operated by stroking a finger over their surface, rather like using a simulated trackball.

Trackball and thumbwheel:

- The trackball is really just an upside-down mouse! A weighted ball faces upwards and is rotated inside a static housing, the motion being detected in the same way as for a mechanical mouse, and the relative motion of the ball moves the cursor.
- Because of this, the trackball requires no additional space in which to operate, and is therefore a very compact device. It is an indirect device, and requires separate buttons for selection. It is fairly accurate, but is hard to draw with, as long movements are difficult.

Joystick and keyboard nipple:

- The joystick is an indirect input device, taking up very little space. Consisting of a small palm-sized box with a stick or shaped grip sticking up from it, the joystick is a simple device with which movements of the stick cause a corresponding movement of the screen cursor. There are two types of joystick: the absolute and the isometric.

Touch-sensitive screens (touchscreens):

- Touchscreens are another method of allowing the user to point and select objects on the screen, but they are much more direct than the mouse, as they detect the presence of the user's finger, or a stylus, on the screen itself.

- They work in one of a number of different ways: by the finger (or stylus) interrupting a matrix of light beams, or by capacitance changes on a grid overlaying the screen, or by ultrasonic reflections. Because the user indicates exactly which item is required by pointing to it, no mapping is required and therefore this is a direct device.

Stylus and light pen:

- For more accurate positioning (and to avoid greasy screens), systems with touch- sensitive surfaces often employ a stylus. Instead of pointing at the screen directly a small pen-like plastic stick is used to point and draw on the screen. This is particularly popular in PDAs, but they are also being used in some laptop computers.

Digitizing tablet:

- The digitizing tablet is a more specialized device typically used for freehand drawing, but may also be used as a mouse substitute. Some highly accurate tablets, usually using a puck (a mouse-like device), are used in special applications such as digitizing information for maps.
- The tablet provides positional information by measuring the position of some device on a special pad, or tablet, and can work in a number of ways. The resistive tablet detects point contact between two separated conducting sheets.
- It has advantages in that it can be operated without a specialized stylus – a pen or the user's finger is sufficient. The magnetic tablet detects current pulses in a magnetic field using a small loop coil housed in a special pen. There are also capacitative and electrostatic tablets that work in a similar way.
- The sonic tablet is similar to the above but requires no special surface. An ultrasonic pulse is emitted by a special pen which is detected by two or more microphones which then triangulate the pen position. This device can be adapted to provide 3D input, if required.

Eyegaze:

- Eyegaze systems allow you to control the computer by simply looking at it! Some systems require you to wear special glasses or a small head-mounted box, others are built into the screen or sit as a small box below the screen.
- A low-power laser is shone into the eye and is reflected off the retina. The reflection changes as the angle of the eye alters, and by tracking the reflected beam the eyegaze system can determine the direction in which the eye is looking.
- The system needs to be calibrated, typically by staring at a series of dots on the screen, but thereafter can be used to move the screen cursor or for other more specialized uses.
- Eyegaze is a very fast and accurate device, but the more accurate versions can be expensive. It is fine for selection but not for drawing since the eye does not move in smooth lines. Also in real applications it can be difficult to distinguish deliberately gazing at something and accidentally glancing at it.

Cursor keys and discrete positioning:

- All of the devices we have discussed are capable of giving near continuous 2D positioning, with varying degrees of accuracy. For many applications we are only interested in positioning within a sequential list such as a menu or amongst 2D cells as in a spreadsheet.

- Even for moving within text discrete up/down left/right keys can sometimes be preferable to using a mouse. Cursor keys are available on most keyboards. Four keys on the keyboard are used to control the cursor, one each for up, down, left and right.

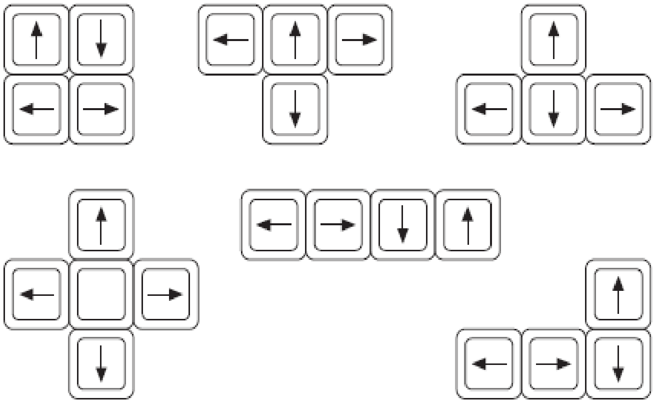


Fig.1.15: Various cursor key layouts

1.6.3 Display Devices:

- The vast majority of interactive computer systems would be unthinkable without some sort of display screen, but many such systems do exist, though usually in specialized applications only.

- Bitmap displays – resolution and color: Virtually all computer displays are based on some sort of bitmap. That is the display is made of vast numbers of colored dots or pixels in a rectangular grid. These pixels may be limited to black and white (for example, the small display on many TV remote controls), in grayscale, or full color.

- The color or, for monochrome screens, the intensity at each pixel is held by the computer's video card. One bit per pixel can store on/off information, and hence only black and white (the term 'bitmap' dates from such displays).

- More bits per pixel give rise to more color or intensity possibilities. As well as the number of colors that can be displayed at each pixel, the other measure that is important is the resolution of the screen. Actually the word 'resolution' is used in a confused (and confusing!) way for screens. There are two numbers to consider:

- The total number of pixels: in standard computer displays this is always in a 4:3 ratio, perhaps 1024 pixels across by 768 down, or 1600 × 1200; for PDAs this will be more in the order of a few hundred pixels in each direction. The density of pixels: this is measured in pixels per inch. Unlike printers (see Section 2.7 below) this density varies little between 72 and 96 pixels per inch. To add to the confusion, a monitor, liquid crystal display (LCD) screen or other display device will quote its maximum resolution, but the computer may actually give it less than this.

Cathode ray tube:

- The cathode ray tube is the television-like computer screen still most common as we write this, but rapidly being displaced by flat LCD screens. It works in a similar way to a standard television screen. A stream of electrons is emitted from an electron gun, which is then focused and directed by magnetic fields.
- As the beam hits the phosphor-coated screen, the phosphor is excited by the electrons and glows The electron beam is scanned from left to right, and then flicked back to rescan the next line, from top to bottom. This is repeated, at about 30 Hz (that is, 30 times a second), per frame, although higher scan rates are sometimes used to reduce the flicker on the screen.
- Another way of reducing flicker is to use interlacing, in which the odd lines on the screen are all scanned first, followed by the even lines. Using a high-persistence phosphor, which glows for a longer time when excited, also reduces flicker, but causes image smearing especially if there is significant animation.

Liquid crystal display:

- If you have used a personal organizer or notebook computer, you will have seen the light, flat plastic screens. These displays utilize liquid crystal technology and are smaller, lighter and consume far less power than traditional CRTs.
- These are also commonly referred to as flat-panel displays. They have no radiation problems associated with them, and are matrix addressable, which means that individual pixels can be accessed without the need for scanning.

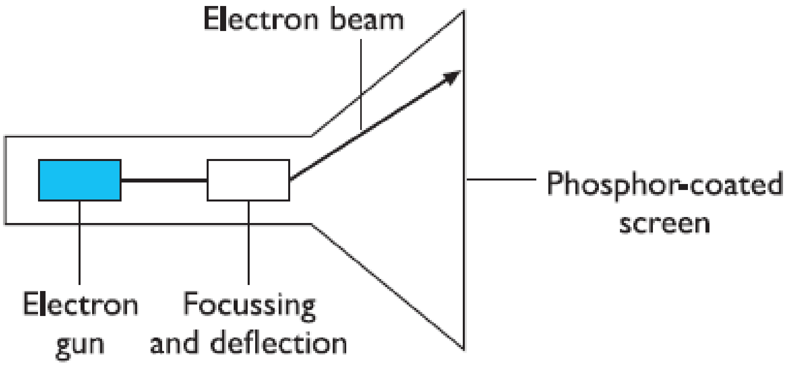


Fig.1.16: CRT screen

Special displays:

- There are a number of other display technologies used in niche markets. The one you are most likely to see is the gas plasma display, which is used in large screens. The random scan display, also known as the directed beam refresh, or vector display, works differently from the bitmap display, also known as raster scan.
- Instead of scanning the whole screen sequentially and horizontally, the random scan draws the lines to be displayed directly. By updating the screen at least 30 Hz to reduce flicker, the direct drawing of lines at any angle means that jaggies are not created, and higher resolutions are possible, up to 4096 × 4096 pixels.

- Color on such displays is achieved using beam penetration technology, and is generally of a poorer quality. The direct view storage tube is used extensively as the display for an analog storage oscilloscope, which is probably the only place that these displays are used in any great numbers.
- They are similar in operation to the random scan CRT but the image is maintained by flood guns which have the advantage of producing a stable display with no flicker. The screen image can be incrementally updated but not selectively erased; removing items has to be done by redrawing the new image on a completely erased screen. The screens have a high resolution, typically about 4096 × 3120 pixels, but suffer from low contrast, low brightness and a difficulty in displaying color.

Large displays and situated displays:

- Displays are no longer just things you have on your desktop or laptop. There are several types of large screen display. Some use gas plasma technology to create large flat bitmap displays. These behave just like a normal screen except they are big and usually have the HDTV (high definition television) wide screen format which has an aspect ratio of 16:9 instead of the 4:3 on traditional TV and monitors.
- Where very large screen areas are required, several smaller screens, either LCD or CRT, can be placed together in a video wall. These can display separate images, or a single TV or computer image can be split up by software or hardware so that each screen displays a portion of the whole and the result is an enormous image. This is the technique often used in large concerts to display the artists or video images during the performance.
- Possibly the large display you are most likely to have encountered is some sort of projector. There are two variants of these. In very large lecture theatres, especially older ones, you see projectors with large red, green and blue lenses.
- These each scan light across the screen to build a full color image. In smaller lecture theatres and in small meetings you are likely to see LCD projectors. Usually the size of a large book, these are like ordinary slide projectors except that where the slide would be there is a small LCD screen instead. The light from the projector passes through the tiny screen and is then focused by the lens onto the screen.

Digital paper:

- A new form of 'display' that is still in its infancy is the various forms of digital paper. These are thin flexible materials that can be written to electronically, just like a computer screen, but which keep their contents even when removed from any electrical supply.

1.6.4 Devices for Virtual Reality And 3D Interaction:

These require you to navigate and interact in a three-dimensional space.

Positioning in 3D space:

- Virtual reality systems present a 3D virtual world. Users need to navigate through these spaces and manipulate the virtual objects they find there.

  Navigation is not simply a matter of moving to a particular location, but also of choosing a particular orientation.
- In addition, when you grab an object in real space, you don't simply move it around, but also twist and turn it, for example when opening a door. Thus the move from mice to 3D devices usually involves a change from two degrees of freedom to six degrees of freedom, not just three.

Cockpit and virtual controls:

- Helicopter and aircraft pilots already have to navigate in real space. Many arcade games and also more serious applications use controls modeled on an aircraft cockpit to 'fly' through virtual space. In many PC games and desktop virtual reality, the controls are themselves virtual. This may be a simulated form of the cockpit controls or more prosaic up/down left/right buttons. The user manipulates these virtual controls using an ordinary mouse

The 3D mouse:

- There are a variety of devices that act as 3D versions of a mouse. Rather than just moving the mouse on a tabletop, you can pick it up, move it in three dimensions, rotate the mouse and tip it forward and backward.
- The 3D mouse has a full six degrees of freedom as its position can be tracked (three degrees), and also its up/down angle (called pitch), its left/right orientation (called yaw) and the amount it is twisted about its own axis (called roll). Various sensors are used to track the mouse position and orientation: magnetic coils, ultrasound or even mechanical joints where the mouse is mounted rather like an angle-poise lamp.

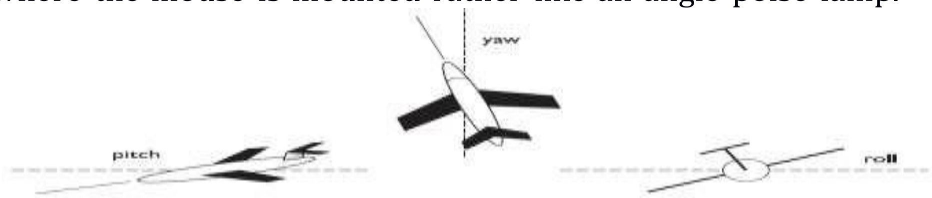


Fig.1.17 : Pitch, yaw and roll

Dataglove:

- One of the mainstays of high-end VR systems, the dataglove is a 3D input device. Consisting of a lycra glove with optical fibers laid along the fingers, it detects the joint angles of the fingers and thumb. As the fingers are bent, the fiber optic cable bends too; increasing bend causes more light to leak from the fiber, and the reduction in intensity is detected by the glove and related to the degree of bend in the joint.
- Attached to the top of the glove are two sensors that use ultrasound to determine 3D positional information as well as the angle of roll, that is the degree of wrist rotation. Such rich multi-dimensional input is currently a solution in search of a problem, in that most of the applications in use do not require such a comprehensive form of data input, whilst those that do cannot afford it.
- However, the availability of cheaper versions of the dataglove will encourage the development of more complex systems that are able to utilize the full power of the dataglove as an input device. There are a number of potential uses for this technology to assist disabled people, but cost remains the limiting factor at present.

Virtual reality helmets:

- The helmets or goggles worn in some VR systems have two purposes: (i) they display the 3D world to each eye and (ii) they allow the user's head position to be tracked. The head tracking is used primarily to feed into the output side. As the user's head moves around the user ought to see different parts of the scene.

Whole-body tracking:

- Some VR systems aim to be immersive, that is to make the users feel as if they are really in the virtual world. In the real world it is possible (although not usually wise) to walk without looking in the direction you are going.

3D displays:

- Just as the 3D images used in VR have led to new forms of input device, they also require more sophisticated outputs. Desktop VR is delivered using a standard computer screen and a 3D impression is produced by using effects such as shadows, occlusion (where one object covers another) and perspective.

Seeing in 3D:

- Our eyes use many cues to perceive depth in the real world. It is in fact quite remarkable as each eye sees only a flattened form of the world, like a photograph. One important effect is stereoscopic vision (or simply stereo vision).
- Because each eye is looking at an object from a slightly different angle each sees a different image and our brain is able to use this to assess the relative distance of different objects. In desktop VR this stereoscopic effect is absent. However, various devices exist to deliver true stereoscopic images.

VR motion sickness:

- We all get annoyed when computers take a long time to change the screen, pop up a window, or play a digital movie. However, with VR the effects of poor display performance can be more serious. In real life when we move our head the image our eyes see changes accordingly.
- VR systems produce the same effect by using sensors in the goggles or helmet and then using the position of the head to determine the right image to show. If the system is slow in producing these images a lag develops between the user moving his head and the scene changing. If this delay is more than a hundred milliseconds or so the feeling becomes disorienting.

Simulators and VR caves:

- Because of the problems of delivering a full 3D environment via head-mounted displays, some virtual reality systems work by putting the user within an environment where the virtual world is displayed upon it.
- The most obvious examples of this are large flight simulators – you go inside a mock-up of an aircraft cockpit and the scenes you would see through the windows are projected onto the virtual windows. In motorbike or skiing simulators in video arcades large screens are positioned to fill the main part of your visual field. You can still look over your shoulder and see your friends, but while you are engaged in the game it surrounds you.

1.6.5 Physical Controls, Sensors And Special Devices

Special displays:

- Apart from the CRT screen there are a number of visual outputs utilized in complex systems, especially in embedded systems. These can take the form of analog representations of numerical values, such as dials, gauges or lights to signify a certain system state. Flashing light-emitting diodes (LEDs) are used on the back of some computers to signify the processor state, whilst gauges and dials are found in process control systems.

Sound output:

- Another mode of output that we should consider is that of auditory signals. Often designed to be used in conjunction with screen displays, auditory outputs are poorly understood: we do not yet know how to utilize sound in a sensible way to achieve maximum effect and information transference. We have discussed speech previously, but other sounds such as beeps, bongs, clanks, whistles and whirrs are all used to varying effect. As well as conveying system

Touch, feel and smell:

- Our other senses are used less in normal computer applications, but you may have played computer games where the joystick or artificial steering wheel vibrated, perhaps when a car was about to go off the track. In some VR applications, such as the use in medical domains to 'practice' surgical procedures, the feel of an instrument moving through different tissue types is very important.
- The devices used to emulate these procedures have force feedback, giving different amounts of resistance depending on the state of the virtual operation. These various forms of force, resistance and texture that influence our physical senses are called haptic devices.
- Haptic devices are not limited to virtual environments, but are used in specialist interfaces in the real world too.

Physical controls:

- A desktop computer system has to serve many functions and so has generic keys and controls that can be used for a variety of purposes. In contrast, these dedicated control panels have been designed for a particular device and for a single use.
- Environment and bio-sensing: In a public washroom there are often no controls for the wash basins, you simply put your hands underneath and (hope that) the water flows. Similarly when you open the door of a car, the courtesy light turns on.
- The washbasin is controlled by a small infrared sensor that is triggered when your hands are in the basin (although it is sometimes hard to find the 'sweet spot' where this happens!). The courtesy lights are triggered by a small switch in the car door.

1.7 MEMORY

- Like human memory, we can think of the computer's memory as operating at different levels, with those that have the faster access typically having less capacity. By analogy with the human memory, we can group these into short-term and long-term memories (STM and LTM).
- RAM and short-term memory (STM): At the lowest level of computer memory are the registers on the computer chip, but these have little impact on the user except in so far as they affect the general speed of the computer. Most currently active information is held in silicon-chip random access memory (RAM).
- Different forms of RAM differ as to their precise access times, power consumption and characteristics. Typical access times are of the order of 10 nanoseconds, that is a hundred-millionth of a second, and information can be accessed at a rate of around 100 Mbytes (million bytes) per second.
- Most RAM is volatile, that is its contents are lost when the power is turned off. However, many computers have small amount of non-volatile RAM, which retains its contents, perhaps with the aid of a small battery. This may be used to store setup information in a large computer, but in a pocket organizer will be the whole memory.

a. Disks and long-term memory (LTM):

- For most computer users the LTM consists of disks, possibly with small tapes for backup. The existence of backups, and appropriate software to generate and retrieve them, is an important area for user security. There are two main kinds of technology used in disks: magnetic disks and optical disks.

- The most common storage media, floppy disks and hard (or fixed) disks, are coated with magnetic material, like that found on an audio tape, on which the information is stored. Typical capacities of floppy disks lie between 300 kbytes and 1.4 Mbytes, but as they are removable, you can have as many as you have room for on your desk.
- Hard disks may store from under 40 Mbytes to several gigabytes (Gbytes), that is several thousand million bytes. With disks there are two access times to consider, the time taken to find the right track on the disk, and the time to read the track.
- Optical disks use laser light to read and (sometimes) write the information on the disk. There are various high capacity specialist optical devices, but the most common is the CD-ROM, using the same technology as audio compact discs.
- CD-ROMs have a capacity of around 650 megabytes, but cannot be written to at all. They are useful for published material such as online reference books, multimedia and software distribution. Recordable CDs are a form of WORM device (write-once read-many) and are more flexible in that information can be written, but (as the name suggests) only once at any location – more like a piece of paper than a blackboard. They are obviously very useful for backups and for producing very secure audit information.

1.8 PROCESSING AND NETWORKS

- Computers that run interactive programs will process in the order of 100 million instructions per second. It sounds a lot and yet, like memory, it can soon be used up. Indeed, the first program written by one of the authors (some while ago) 'hung' and all attempts to debug it failed.
- Effects of finite processor speed: As we can see, speed of processing can seriously affect the user interface. These effects must be taken into account when designing an interactive system. There are two sorts of faults due to processing speed: those when it is too slow, and those when it is too fast.
- The first type of fault is functional fault in which the program did the wrong thing. A second fault due to slow processing is where, in a sense, the program does the right thing, but the feedback is too slow, leading to strange effects at the interface. In order to avoid faults of the first kind, the system buffers the user input; that is, it remembers keypresses and mouse buttons and movement. Unfortunately, this leads to problems of its own.
- One example of this sort of problem is cursor tracking, which happens in character-based text editors. A similar problem, icon wars, occurs on window systems. The user clicks the mouse on a menu or icon, and nothing happens; for some reason the machine is busy or slow. So the user clicks again, tries something else – then, suddenly, all the buffered mouse clicks are interpreted and the screen becomes a blur of flashing windows and menus.

Limitations on interactive performance

There are several factors that can limit the speed of an interactive system:

Computation bound:

- This is rare for an interactive program, but possible, for example when using find/replace in a large document. The system should be designed so that long delays are not in the middle of interaction and so that the user gets some idea of how the job is progressing. For a very long process try to give an indication of duration before it starts; and during processing an indication of the stage that the process has reached is helpful.

Storage channel bound:

- The speed of memory access can interfere with interactive performance. We discussed one technique, laziness, for reducing this effect. In addition, if there is plenty of raw computation power and the system is held up solely by memory, it is possible to trade off memory against processing speed.
- Thus faster memory access leads to increased processing time. If data is written more often than it is read, one can choose a technique that is expensive to compress but fairly simple to decompress. For many interactive systems the ability to browse quickly is very important, but users will accept delays when saving updated information.

Graphics bound:

- For many modern interfaces, this is the most common bottleneck. It is easy to underestimate the time taken to perform what appear to be simple interface operations.
- Sometimes clever coding can reduce the time taken by common graphics operations, and there is tremendous variability in performance between programs running on the same hardware. Most computers include a special-purpose graphics card to handle many of the most common graphics operations.
- This is optimized for graphics operations and allows the main processor to do other work such as manipulating documents and other user data.

Network capacity:

- Most computers are linked by networks. At the simplest this can mean using shared files on a remote machine. When accessing such files it can be the speed of the network rather than that of the memory which limits performance.

1.9 INTERACTION

- There are a number of ways in which the user can communicate with the system. At one extreme is batch input, in which the user provides all the information to the computer at once and leaves the machine to perform the task.
- This approach does involve an interaction between the user and computer but does not support many tasks well. At the other extreme are highly interactive input devices and paradigms, such as direct manipulation and the applications of virtual reality.
- Here the user is constantly providing instruction and receiving feedback. These are the types of interactive system we are considering.

1.10 MODELS

- Both are complex, as we have seen, and are very different from each other in the way that they communicate and view the domain and the task. The interface must therefore effectively translate between them to allow the interaction to be successful.
- This translation can fail at a number of points and for a number of reasons. The use of models of interaction can help us to understand exactly what is going on in the interaction and identify the likely root of difficulties. They also provide us with a framework to compare different interaction styles and to consider interaction problems.

The terms of interaction:

- The purpose of an interactive system is to aid a user in accomplishing goals from some application domain. A domain defines an area of expertise and knowledge in some real-world activity. Tasks are operations to manipulate the concepts of a domain. A goal is the desired output from a performed task.

- An intention is a specific action required to meet the goal. Task analysis involves the identification of the problem space for the user of an interactive system in terms of the domain, goals, intentions and tasks. The concepts used in the design of the system and the description of the user are separate, and so we can refer to them as distinct components, called the System and the User, respectively.
- The System and User are each described by means of a language that can express concepts relevant in the domain of the application.The System's language we will refer to as the core language and the User's language we will refer to as the task language.
- The core language describes computational attributes of the domain relevant to the System state, whereas the task language describes psychological attributes of the domain relevant to the User state. The system is assumed to be some computerized application, in the context of this book, but the models apply equally to non-computer applications. It is also a common assumption that by distinguishing between user and system we are restricted to single-user applications.

The execution–evaluation cycle:

- Norman's model of interaction is perhaps the most influential in Human–Computer Interaction, possibly because of its closeness to our intuitive understanding of the interaction between human user and computer. The user formulates a plan of action, which is then executed at the computer interface.
- When the plan, or part of the plan, has been executed, the user observes the computer interface to evaluate the result of the executed plan, and to determine further actions. The interactive cycle can be divided into two major phases: execution and evaluation. These can then be subdivided into further stages, seven in all. The stages in Norman's model of interaction are as follows:
    a. Establishing the goal.
    b. Forming the intention.
    c. Specifying the action sequence.
    d. Executing the action.
    e. Perceiving the system state.
    f. Interpreting the system state.
- Evaluating the system state with respect to the goals and intentions. Each stage is, of course, an activity of the user. First the user forms a goal. This is the user's notion of what needs to be done and is framed in terms of the domain, in the task language.
- It is liable to be imprecise and therefore needs to be translated into the more specific intention, and the actual actions that will reach the goal, before it can be executed by the user. The user perceives the new state of the system, after execution of the action sequence, and interprets it in terms of his expectations. If the system state reflects the user's goal then the computer has done what he wanted and the interaction has been successful; otherwise the user must formulate a new goal and repeat the cycle.

- The interaction framework attempts a more realistic description of interaction by including the system explicitly, and breaks it into four main components, as shown in figure below. The nodes represent the four major components in an interactive system – the System, the User, the Input and the Output.

- Each component has its own language. In addition to the User's task language and the System's core language, which we have already introduced, there are languages for both the Input and Output components. Input and Output together form the Interface. As the interface sits between the User and the System, there are four steps in the interactive cycle, each corresponding to a translation from one component to another, as shown by the labeled arcs in figure below.

- The User begins the interactive cycle with the formulation of a goal and a task to achieve that goal. The only way the user can manipulate the machine is through the Input, and so the task must be articulated within the input language. The input language is translated into the core language as operations to be performed by the System.

- The System then transforms itself as described by the operations; the execution phase of the cycle is complete and the evaluation phase now begins. The System is in a new state, which must now be communicated to the User.

- The current values of system attributes are rendered as concepts or features of the Output. It is then up to the User to observe the Output and assess the results of the interaction relative to the original goal, ending the evaluation phase and, hence, the interactive cycle. There are four main translations involved in the interaction: articulation, performance, presentation and observation

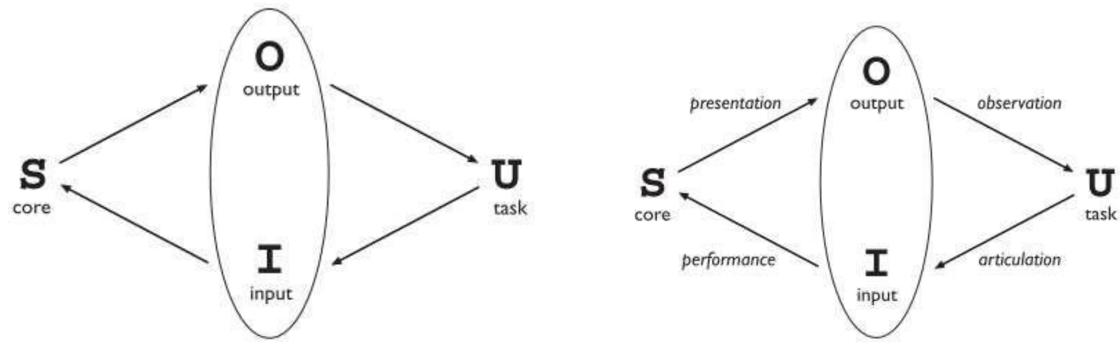


Fig1.18:a.General Interaction Framework b.Translations between components

- The general interaction framework Translations between components is shown in figure 1.18. The User's formulation of the desired task to achieve some goal needs to be articulated in the input language. The tasks are responses of the User and they need to be translated to stimuli for the Input.

- As pointed out above, this articulation is judged in terms of the coverage from tasks to input and the relative ease with which the translation can be accomplished. The task is phrased in terms of certain psychological attributes that highlight the important features of the domain for the User. If these psychological attributes map clearly onto the input language, then articulation of the task will be made much simpler.
- At the next stage, the responses of the Input are translated to stimuli for the System. Of interest in assessing this translation is whether the translated input language can reach as many states of the System as is possible using the System stimuli directly. The ease with which this translation from Input to System takes place is of less importance because the effort is not expended by the user.
- However, there can be a real effort expended by the designer and programmer. In this case, the ease of the translation is viewed in terms of the cost of implementation. Once a state transition has occurred within the System, the execution phase of the interaction is complete and the evaluation phase begins.
- The new state of the System must be communicated to the User, and this begins by translating the System responses to the transition into stimuli for the Output component. This presentation translation must preserve the relevant system attributes from the domain in the limited expressiveness of the output devices.
- The ability to capture the domain concepts of the System within the Output is a question of expressiveness for this translation. The response from the Output is translated to stimuli for the User which trigger assessment. The observation translation will address the ease and coverage of this final translation.
- For example, it is difficult to tell the time accurately on an unmarked analog clock, especially if it is not oriented properly. It is difficult in a command line interface to determine the result of copying and moving files in a hierarchical file system. Developing a website using a markup language like HTML would be virtually impossible without being able to preview the output through a browser.

1.11 FRAMEWORKS

- The ACM SIGCHI Curriculum Development Group presents a framework similar to that presented here, and uses it to place different areas that relate to HCI. The Figure these aspects are shown as they relate to the interaction framework. In particular, the field of ergonomics addresses issues on the user side of the interface, covering both input and output, as well as the user's immediate context as in figure 1.19.
- Dialog design and interface styles can be placed particularly along the input branch of the framework, addressing both articulation and performance. However, dialog is most usually associated with the computer and so is biased to that side of the framework.

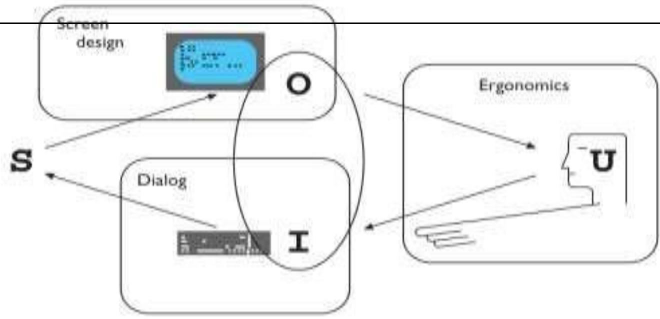


Figure1.19: A framework for human-computer interaction

- Presentation and screen design relates to the output branch of the framework. The entire framework can be placed within a social and organizational context that also affects the interaction. Each of these areas has important implications for the design of interactive systems and the performance of the user.
- A framework for human–computer interaction. Adapted from ACM SIGCHI Curriculum Development Group

1.12 ERGONOMICS

- Ergonomics (or human factors) is traditionally the study of the physical characteristics of the interaction: how the controls are designed, the physical environment in which the interaction takes place, and the layout and physical qualities of the screen. A primary focus is on user performance and how the interface enhances or detracts from this.

Arrangement of controls and displays:

- Considered perceptual and cognitive issues that affect the way we present information on a screen and provide control mechanisms to the user. In addition to these cognitive aspects of design, physical aspects are also important. Sets of controls and parts of the display should be grouped logically to allow rapid access by the user.
- This may not seem so important when we are considering a single user of a spreadsheet on a PC, but it becomes vital when we turn to safety-critical applications such as plant control, aviation and air traffic control. In each of these contexts, users are under pressure and are faced with a huge range of displays and controls. Indeed, returning to the less critical PC application, inappropriate placement of controls and displays can lead to inefficiency and frustration.
- The exact organization that this will suggest will depend on the domain and the application, but possible organizations include the following: functional controls and displays are organized so that those that are functionally related are placed together; sequential controls and displays are organized to reflect the order of their use in a typical interaction (this may be especially appropriate in domains where a particular task sequence is enforced, such as aviation); frequency controls and displays are organized according to how frequently they are used, with the most commonly used controls being the most easily accessible.

The physical environment of the interaction:

- As well as addressing physical issues in the layout and arrangement of the machine interface, ergonomics is concerned with the design of the work environment itself.
- The first consideration here is the size of the users. Obviously this is going to vary considerably. However, in any system the smallest user should be able to reach all the controls (this may include a user in a wheelchair), and the largest user should not be cramped in the environment.

- All users should be comfortably able to see critical displays. For long periods of use, the user should be seated for comfort and stability. Seating should provide back support. If required to stand, the user should have room to move around in order to reach all the controls.

Health issues:

- Leaving aside the obvious safety risks of poorly designed safety-critical systems (aircraft crashing, nuclear plant leaks and worse), there are a number of factors that may affect the use of more general computers. Again these are factors in the physical environment that directly affect the quality of the interaction and the user's performance:

Physical position:

- As we noted in the previous section, users should be able to reach all controls comfortably and see all displays. Users should not be expected to stand for long periods and, if sitting, should be provided with back support. If a particular position for a part of the body is to be adopted for long periods (for example, in typing) support should be provided to allow rest.

Temperature:

- Although most users can adapt to slight changes in temperature without adverse effect, extremes of hot or cold will affect performance and, in excessive cases, health. Experimental studies show that performance deteriorates at high or low temperatures, with users being unable to concentrate efficiently.

Lighting:

- The lighting level will again depend on the work environment. However, adequate lighting should be provided to allow users to see the computer screen without discomfort or eyestrain. The light source should also be positioned to avoid glare affecting the display.

Noise:

- Excessive noise can be harmful to health, causing the user pain, and in acute cases, loss of hearing. Noise levels should be maintained at a comfortable level in the work environment. This does not necessarily mean no noise at all. Noise can be a stimulus to users and can provide needed confirmation of system activity.

Time:

- The time users spend using the system should also be controlled. It has been suggested that excessive use of CRT displays can be harmful to users, particularly pregnant women.

The use of color:

- Ergonomics has a close relationship to human psychology in that it is also concerned with the perceptual limitations of humans. For example, the use of color in displays is an ergonomics issue.
- The visual system has some limitations with regard to color, including the number of colors that are distinguishable and the relatively low blue acuity. We also saw that a relatively high proportion of the population suffers from a deficiency in color vision.
- Colors used in the display should be as distinct as possible and the distinction should not be affected by changes in contrast. The colors used should also correspond to common conventions and user expectations.

1.13 STYLES

- The choice of interface style can have a profound effect on the nature. There are a number of common interface styles including
    - command line interface
    - menus
    - natural language
    - question/answer and query dialog
    - form-fills and spreadsheets
    - WIMP
    - point and click
    - three-dimensional interfaces.

As the WIMP interface is the most common and complex.

Command line interface:

- Command line interfaces are powerful in that they offer direct access to system functionality (as opposed to the hierarchical nature of menus), and can be combined to apply a number of tools to the same data. They are also flexible: the command often has a number of options or parameters that will vary its behavior in some way, and it can be applied to many objects at once, making it useful for repetitive tasks.
- However, this flexibility and power brings with it difficulty in use and learning. Commands must be remembered, as no cue is provided in the command line to indicate which command is needed. They are therefore better for expert users than for novices.
- This problem can be alleviated a little by using consistent and meaningful commands and abbreviations. The commands used should be terms within the vocabulary of the user rather than the technician. Unfortunately, commands are often obscure and vary across systems, causing confusion to the user and increasing the overhead of learning.Typical example: the Unix system

Menus:

- In a menu-driven interface, the set of options available to the user is displayed on the screen, and selected using the mouse, or numeric or alphabetic keys. Since the options are visible they are less demanding of the user, relying on recognition rather than recall. However, menu options still need to be meaningful and logically grouped to aid recognition.
- Often menus are hierarchically ordered and the option required is not available at the top layer of the hierarchy. The grouping and naming of menu options then provides the only cue for the user to find the required option.
- Such systems either can be purely text based, with the menu options being presented as numbered choices, or may have a graphical component in which the menu appears within a rectangular box and choices are made, perhaps by typing the initial letter of the desired selection, or by entering the associated number, or by moving around the menu with the arrow keys.

Fig.1.20 : Menu-driven interface

Natural language:

- Natural language understanding, both of speech and written input, is the subject of much interest and research. Unfortunately, however, the ambiguity of natural language makes it very difficult for a machine to understand. Language is ambiguous at a number of levels. It is important in interfaces which use natural language in this restricted form that the user is aware of the limitations of the system and does not expect too much understanding.
- The use of natural language in restricted domains is relatively successful, but it is debatable whether this can really be called natural language. The user still has to learn which phrases the computer understands and may become frustrated if too much is expected. However, it is also not clear how useful a general natural language interface would be. Language is by nature vague and imprecise: this gives it its flexibility and allows creativity in expression. Computers, on the other hand, require precise instructions.

Question/answer and query dialog:

- Question and answer dialog is a simple mechanism for providing input to an application in a specific domain. The user is asked a series of questions (mainly with yes/no responses, multiple choice, or codes) and so is led through the interaction step by step.
- These interfaces are easy to learn and use, but are limited in functionality and power. As such, they are appropriate for restricted domains (particularly information systems) and for novice or casual users. Query languages, on the other hand, are used to construct queries to retrieve information from a database.
- They use natural-language-style phrases, but in fact require specific syntax, as well as knowledge of the database structure. Queries usually require the user to specify an attribute or attributes for which to search the database, as well as the attributes of interest to be displayed. This is straight-forward where there is a single attribute, but becomes complex when multiple attributes are involved.

Form-fills and spreadsheets:

- Form-filling interfaces are used primarily for data entry but can also be useful in data retrieval applications. The user is presented with a display resembling a paper form, with slots to fill in. Often the form display is based upon an actual form with which the user is familiar, which makes the interface easier to use.
- The user works through the form, filling in appropriate values. The data are then entered into the application in the correct place. Most form-filling interfaces allow easy movement around the form and allow some fields to be left blank.
- They also require correction facilities, as users may change their minds or make a mistake about the value that belongs in each field. The dialog style is useful primarily for data entry applications and, as it is easy to learn and use, for novice users. However, assuming a design that allows flexible entry, form filling is also appropriate for expert users.

- Spreadsheets are a sophisticated variation of form filling. The spreadsheet comprises a grid of cells, each of which can contain a value or a formula. The formula can involve the values of other cells (for example, the total of all cells in this column).
- The user can enter and alter values and formulae in any order and the system will maintain consistency amongst the values displayed, ensuring that all formulae are obeyed. The user can therefore manipulate values to see the effects of changing different parameters.
- Spreadsheets are an attractive medium for interaction: the user is free to manipulate values at will and the distinction between input and output is blurred, making the interface more flexible and natural.

WIMP:

- WIMP stands for windows, icons, menus and pointers (sometimes windows, icons, mice and pull-down menus), and is the default interface style for the majority of interactive computer systems in use today, especially in the PC and desktop workstation arena.
- Examples of WIMP interfaces include Microsoft Windows for IBM PC compatibles, MacOS for Apple Macintosh compatibles and various X Windows-based systems for UNIX.

Point and click:

- In most multimedia systems and in web browsers, virtually all actions take only a single click of the mouse button. You may point at a city on a map and when you click a window opens, showing you tourist information about the city. This point-and-click interface style is obviously closely related to the WIMP style.
- It clearly overlaps in the use of buttons, but may also include other WIMP elements. However, the philosophy is simpler and more closely tied to ideas of hypertext. In addition, the point-and-click style is not tied to mouse-based interfaces, and is also extensively used in touchscreen information systems. In this case, it is often combined with a menu-driven interface.

Three-dimensional interfaces:

- The simplest technique is where ordinary WIMP elements, buttons, scroll bars, etc., are given a 3D appearance using shading, giving the appearance of being sculpted out of stone. By unstated convention, such interfaces have a light source at their top right.
- Where used judiciously, the raised areas are easily identifiable and can be used to highlight active areas. Unfortunately, some interfaces make indiscriminate use of sculptural effects, on every text area, border and menu, so all sense of differentiation is lost.
- A more complex technique uses interfaces with 3D workspaces. The objects displayed in such systems are usually flat, but are displayed in perspective when at an angle to the viewer and shrink when they are ‘further away’.

1.14 ELEMENTS OF WIMP INTERFACE

- WIMP stands for windows, icons, menus and pointers (sometimes windows, icons, mice and pull-down menus).There are also many additional interaction objects and techniques commonly used in WIMP interfaces, some designed for specific purposes and others more general. Together, these elements of the WIMP interfaces are called widgets, and they comprise the toolkit for interaction between user and system.

Windows:

- Windows are areas of the screen that behave as if they were independent terminals in their own right. A window can usually contain text or graphics, and can be moved or resized. More than one window can be on a screen at once, allowing separate tasks to be visible at the same time.
- Users can direct their attention to the different windows as they switch from one thread of work to another. If one window overlaps the other, the back window is partially obscured, and then refreshed when exposed again.
- Overlapping windows can cause problems by obscuring vital information, so windows may also be tiled, when they adjoin but do not overlap each other. Alternatively, windows may be placed in a cascading fashion, where each new window is placed slightly to the left and below the previous window.
- In some systems this layout policy is fixed, in others it can be selected by the user. Usually, windows have various things associated with them that increase their usefulness.
- Scrollbars are one such attachment, allowing the user to move the contents of the window up and down, or from side to side. This makes the window behave as if it were a real window onto a much larger world, where new information is brought into view by manipulating the scrollbars.

Icons:

- A small picture is used to represent a closed window, and this representation is known as an icon. By allowing icons, many windows can be available on the screen at the same time, ready to be expanded to their full size by clicking on the icon. Shrinking a window to its icon is known as iconifying the window.
- When a user temporarily does not want to follow a particular thread of dialog, he can suspend that dialog by iconifying the window containing the dialog. The icon saves space on the screen and serves as a reminder to the user that he can subsequently resume the dialog by opening up the window.
- Icons can also be used to represent other aspects of the system, such as a waste-basket for throwing unwanted files into, or various disks, programs or functions that are accessible to the user. Icons can take many forms: they can be realistic representations of the objects that they stand for, or they can be highly stylized. They can even be arbitrary symbols, but these can be difficult for users to interpret.

Pointers:

- The pointer is an important component of the WIMP interface, since the interaction style required by WIMP relies very much on pointing and selecting things such as icons. The mouse provides an input device capable of such tasks, although joysticks and trackballs are other alternatives.

- The different shapes of cursor are often used to distinguish modes, for example the normal pointer cursor may be an arrow, but change to cross-hairs when drawing a line. Cursors are also used to tell the user about system activity, for example a watch or hour-glass cursor may be displayed when the system is busy reading a file. Pointer cursors are like icons, being small bitmap images, but in addition all cursors have a hot-spot, the location to which they point.

Menus:

- A menu is an interaction technique that is common across many non-windowing systems as well. A menu presents a choice of operations or services that can be performed by the system at a given time. The pointing device is used to indicate the desired option.
- As the pointer moves to the position of a menu item, the item is usually highlighted (by inverse video, or some similar strategy) to indicate that it is the potential candidate for selection.
- Selection usually requires some additional user action, such as pressing a button on the mouse that controls the pointer cursor on the screen or pressing some special key on the keyboard.
- Menus are inefficient when they have too many items, and so cascading menus are utilized, in which item selection opens up another menu adjacent to the item, allowing refinement of the selection. Several layers of cascading menus can be used. The main menu can be visible to the user all the time, as a menu bar and submenus can be pulled down or across from it upon request.
- Menu bars are often placed at the top of the screen or at the top of each window. Alternatives include menu bars along one side of the screen, or even placed amongst the windows in the main 'desktop' area. Websites use a variety of menu bar locations, including top, bottom and either side of the screen. Alternatively, the main menu can be hidden and upon request it will pop up onto the screen.
- Pull-down menus are dragged down from the title at the top of the screen, by moving the mouse pointer into the title bar area and pressing the button. Fall-down menus are similar, except that the menu automatically appears when the mouse pointer enters the title bar, without the user having to press the button.
- Some menus are pin-up menus, in that they can be 'pinned' to the screen, staying in place until explicitly asked to go away. Pop-up menus appear when a particular region of the screen, maybe designated by an icon, is selected, but they only stay as long as the mouse button is depressed.
- Another approach to menu selection is to arrange the options in a circular fashion. The pointer appears in the center of the circle, and so there is the same distance to travel to any of the selections. This has the advantages that it is easier to select items, since they can each have a larger target area, and that the selection time for each item is the same, since the pointer is equidistant from them all.

Fig.1.21 : Elements of the WIMP interface

Buttons:

- Buttons are individual and isolated regions within a display that can be selected by the user to invoke specific operations. These regions are referred to as buttons because they are purposely made to resemble the push buttons you would find on a control panel.
- ‘Pushing’ the button invokes a command, the meaning of which is usually indicated by a textual label or a small icon. Buttons can also be used to toggle between two states, displaying status information such as whether the current font is italicized or not in a word processor, or selecting options on a web form.
- Such toggle buttons can be grouped together to allow a user to select one feature from a set of mutually exclusive options, such as the size in points of the current font. These are called radio buttons, since the collection functions much like the old-fashioned mechanical control buttons on car radios.
- If a set of options is not mutually exclusive, such as font characteristics like bold, italics and underlining, then a set of toggle buttons can be used to indicate the on/off status of the options. This type of collection of buttons is sometimes referred to as check boxes.

Toolbars:

- Many systems have a collection of small buttons, each with icons, placed at the top or side of the window and offering commonly used functions. The function of this toolbar is similar to a menu bar, but as the icons are smaller than the equivalent text more functions can be simultaneously displayed.

- Sometimes the content of the toolbar is fixed, but often users can customize it, either changing which functions are made available, or choosing which of several predefined toolbars is displayed.

Palettes:

- In many application programs, interaction can enter one of several modes. The defining characteristic of modes is that the interpretation of actions, such as keystrokes or gestures with the mouse, changes as the mode changes. Problems occur if the user is not aware of the current mode.
- Palettes are a mechanism for making the set of possible modes and the active mode visible to the user. A palette is usually a collection of icons that are reminiscent of the purpose of the various modes. An example in a drawing package would be a collection of icons to indicate the pixel color or pattern that is used to fill in objects, much like an artist's palette for paint.

Dialog boxes:

- Dialog boxes are information windows used by the system to bring the user's attention to some important information, possibly an error or a warning used to prevent a possible error. Alternatively, they are used to invoke a subdialog between user and system for a very specific task that will normally be embedded within some larger task.
- When the user or system wants to save the file, a dialog box can be used to allow the user to name the file and indicate where it is to be located within the filing system. When the save subdialog is complete, the dialog box will disappear.
- Just as windows are used to separate the different threads of user–system dialog, so too are dialog boxes used to factor out auxiliary task threads from the main task dialog.

1.15 INTERACTIVITY

- Interactivity is the defining feature of an interactive system. This can be seen in many areas of HCI. For example, the recognition rate for speech recognition is too low to allow transcription from tape, but in an airline reservation system, so long as the system can reliably recognize yes and no it can reflect back its understanding of what you said and seek confirmation.
- Speech-based input is difficult, speech-based interaction easier. Also, in the area of information visualization the most exciting developments are all where users can interact with a visualization in real time, changing parameters and seeing the effect.
- Interactivity is also crucial in determining the 'feel' of a WIMP environment. All WIMP systems appear to have virtually the same elements: windows, icons, menus, pointers, dialog boxes, buttons, etc.
- The precise behavior of these elements differs both within a single environment and between environments. For example, we have already discussed the different behavior of pull-down and fall-down menus. These look the same, but fall-down menus are more easily invoked by accident (and not surprisingly the windowing environments that use them have largely fallen into disuse!).
- Menus are a major difference between the MacOS and Microsoft Windows environments: in MacOS you have to keep the mouse depressed through- out menu selection; in Windows you can click on the menu bar and a pull-down menu appears and remains there until an item is selected or it is cancelled.
- Older computer systems, the order of interaction was largely determined by the machine. You did things when the computer was ready. In WIMP environments, the user takes the initiative, with many options and often many applications simultaneously available.

- The exceptions to this are pre-emptive parts of the interface, where the system for various reasons wrests the initiative away from the user, perhaps because of a problem or because it needs information in order to continue.
- The major example of this is modal dialog boxes. It is often the case that when a dialog box appears the application will not allow you to do anything else until the dialog box has been completed or cancelled. In some cases this may simply block the application, but you can perform tasks in other applications. In other cases you can do nothing at all until the dialog box has been completed.
- There are occasions when modal dialog boxes are necessary, for example when a major fault has been detected, or for certain kinds of instructional software. However, the general philosophy of modern systems suggests that one should minimize the use of pre-emptive elements, allowing the user maximum flexibility.
- Interactivity is also critical in dealing with errors. Slips and mistakes is a way to try to prevent these types of errors. The other way to deal with errors is to make sure that the user or the system is able to tell when errors have occurred. If users can detect errors then they can correct them. So, even if errors occur, the interaction as a whole succeeds. This ability to detect and correct is important both at the small scale of button presses and keystrokes and also at the large scale.

1.16 PARADIGMS

- Paradigms is
    - Predominant theoretical frameworks or scientific world views
        - e.g., Aristotelian, Newtonian, Einsteinian (relativistic) paradigms in physics
    - Understanding HCI history is largely about understanding a series of paradigm shifts
        - Not all listed here are necessarily “paradigm” shifts, but are at least candidates
        - History will judge which are true shifts

Paradigms of interaction

- New computing technologies arrive, creating a new perception of the human computer relationship. We can trace some of these shifts in the history of interactive technologies.The initial paradigm
    - Batch processing
    - Time-sharing
    - Networking
    - Graphical displays
    - Microprocessor
    - WWW
    - Ubiquitous Computing

Time-sharing

- 1940s and 1950s – explosive technological growth
- 1960s – need to channel the power
- J.C.R. Licklider at ARPA
- single computer supporting multiple users

Video Display Units

- more suitable medium than paper
- 1962 – Sutherland's Sketchpad

- computers for visualizing and manipulating data
- one person's contribution could drastically change the history of computing

Programming toolkits
- Engelbart at Stanford Research Institute
- 1963 – augmenting man's intellect
- 1968 NLS/Augment system demonstration
- the right programming toolkit provides building blocks to producing complex interactive systems

Personal computing
- 1970s – Papert's LOGO language for simple graphics programming by children
- A system is more powerful as it becomes easier to user
- Future of computing in small, powerful machines dedicated to the individual
- Kay at Xerox PARC – the Dynabook as the ultimate personal computer

Window systems and the WIMP interface
- humans can pursue more than one task at a time
- windows used for dialogue partitioning, to “change the topic”
- 1981 – Xerox Star first commercial windowing system
- windows, icons, menus and pointers now familiar interaction mechanisms

Metaphor
- relating computing to other real-world activity is effective teaching technique
  - LOGO's turtle dragging its tail
  - file management on an office desktop
  - word processing as typing
  - financial analysis on spreadsheets
  - virtual reality – user inside the metaphor
- Problems
  - some tasks do not fit into a given metaphor
  - cultural bias

Direct manipulation
- 1982 – Shneiderman describes appeal of graphically-based interaction
  - visibility of objects
  - incremental action and rapid feedback
  - reversibility encourages exploration
  - syntactic correctness of all actions
  - replace language with action
  - 1984 – Apple Macintosh
- the model-world metaphor
- What You See Is What You Get (WYSIWYG)

Language versus Action
- actions do not always speak louder than words!
- DM – interface replaces underlying system
- language paradigm
- interface as mediator
- interface acts as intelligent agent
- programming by example is both action and language

Hypertext
- 1945 – Vannevar Bush and the memex
- key to success in managing explosion of information

- mid 1960s – Nelson describes hypertext as non-linear browsing structure
- hypermedia and multimedia
- Nelson's Xanadu project still a dream today

Multimodality
- a mode is a human communication channel
- emphasis on simultaneous use of multiple channels for input and output

Computer Supported Cooperative Work (CSCW)
- CSCW removes bias of single user / single computer system
- Can no longer neglect the social aspects
- Electronic mail is most prominent success

The World Wide Web
- Hypertext, as originally realized, was a closed system
- Simple, universal protocols (e.g. HTTP) and mark-up languages (e.g. HTML) made publishing and accessing easy
- Critical mass of users lead to a complete transformation of our information economy.

Agent-based Interfaces
- Original interfaces
  - Commands given to computer
  - Language-based
- Direct Manipulation/WIMP
  - Commands performed on “world” representation
  - Action based
- Agents - return to language by instilling proactivity and “intelligence” in command processor
  - Avatars, natural language processing

Ubiquitous Computing
“The most profound technologies are those that disappear.”
Mark Weiser, 1991
Late 1980’s: computer was very apparent
How to make it disappear?
  - Shrink and embed/distribute it in the physical world
  - Design interactions that don’t demand our intention

Sensor-based and Context-aware Interaction
- Humans are good at recognizing the “context” of a situation and reacting appropriately
- Automatically sensing physical phenomena (e.g., light, temp, location, identity) becoming easier

Case Studies

1. Automatic Syringe:

The eventual users should be involved in the design process. They have vital knowledge and will soon find flaws. A mechanical syringe was once being developed and a prototype was demonstrated to hospital staff. Happily they quickly noticed the potentially fatal flaw in its interface.

The doses were entered via a numeric keypad: **an accidental keypress** and the **dose could be out by a factor of 10!** The production version had individual increment/decrement buttons for each digit.

People are complicated, so you won't get it right first time. Programming an interface can be a very difficult and time-consuming Process. So, the result becomes precious and the builder will want to defend it and minimize changes. Making early prototypes less precious and easier to throw away is crucial.

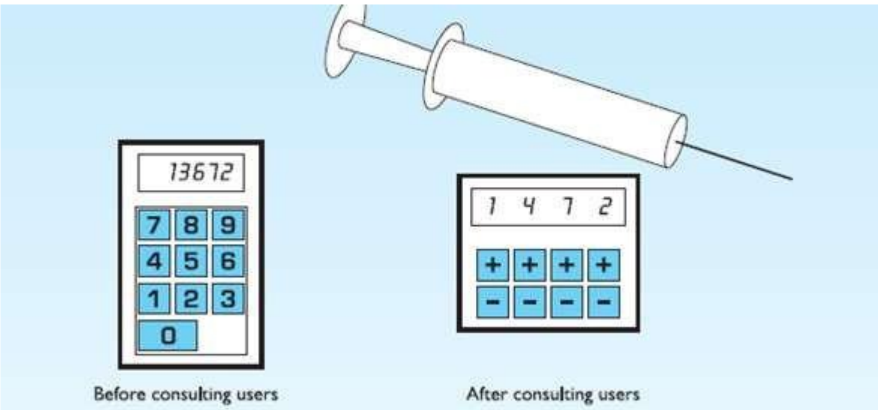


Figure 0.1 Automatic syringe: setting the dose to 1372. The effect of one key slip before and after user involvement

2. **Handling the goods**

E-commerce has become very successful in some areas of sales, such as travel services, books and CDs, and food. However, in some retail areas, such as clothes **shopping**, e-commerce has been less successful. Why?

When buying train and airline tickets and, to some extent, books and food, the experience of shopping is less important than the convenience. So, as long as we know what we want, we are happy to shop online. With clothes, the experience of shopping is far more important. We need to be able to **handle the goods, feel the texture of the material, check the weight to test quality**. Even if we know that something will fit us we still want to be able to handle it before buying.

Research into **haptic interaction** is looking at ways of solving this problem. By using special force feedback and tactile hardware, users are able to feel surfaces and shape. For example, a demonstration environment called "**TouchCity**" allows people to walk around a **virtual shopping mall**, pick up products and feel their texture and weight. A key problem with the commercial use of such an application, however, is that the haptic experience requires expensive hardware not yet available to the average e-shopper.

3. Cashing in

Early automatic teller machines (**ATMs**) gave the **customer money before returning their bank card.** On receiving the money the customer would reach closure and hence often forget to take the card. when we complete some part of a task , our minds have a tendency to flush short-term memory in order to get on with the next job.

Modern ATMs **design changed** and it **returns the card first**! .

4. 7 ± 2 revisited

When we looked at short-term memory, we noted the general rule that people can hold 7 ± 2 items or chunks of information in short-term memory. It is a principle that people tend to remember but it can be misapplied. For example, it is often suggested that this means that **lists, menus and other groups of items** should be designed to be **no more than 7 items long.** But use of menus and lists of course has little to do with short-term memory – they are available in the environment as cues and so do not need to be remembered.

On the other hand the 7 ± 2 rule would apply in command line interfaces. Imagine a scenario where a **UNIX** user looks up a command in the manual. Perhaps the **command has a number of parameters of options**, to be applied in a particular order, and it is going to be applied to several files that have long path names. The user then has to hold the command, its parameters and the file path names in short term memory while he types them in. Here we could say that the task may cause problems if the number of items or chunks in the command line string is more than 7.

5. Memorable or secure?

As online activities become more widespread, people are having to remember more and more access information, such as **passwords and security checks.** The average active internet user may have separate passwords and user names for several email accounts, mailing lists, e-shopping sites, e-banking, online auctions and more! Remembering these passwords is not easy. From a security perspective it is important that passwords are random. Words and names are very easy to crack, hence the recommendation that passwords are frequently changed and constructed from random strings of letters and numbers. But in reality these are the hardest things for people to commit to memory. Hence many people will use the **same password for all their online activities** and will choose a word or a name that is easy for them to remember, in spite of the obviously increased security risks. Security here is in conflict with memorability!

A solution to this is to **construct a nonsense password** out of letters or numbers that will have meaning to you but will not make up a word in a dictionary (e.g. initials of names, numbers from significant dates or postcodes, and so on). Then what is remembered is the meaningful rule for constructing the password, and not a meaningless string of alphanumeric characters.

6. Improve your memory

Many people can perform excellence in memory, where we have exercises like recalling the sequence of cards in a pack , or recounting π to 1000 decimal places.

There are exercises to improve memory abilities , one example discussed below,

Look at the list below of numbers and associated words:

1 bun	6 sticks
2 shoe	7 heaven
3 tree	8 gate
4 door	9 wine
5 hive	10 hen

Notice that the words sound similar to the numbers. Now think about the words one at a time and visualize them, in as much detail as possible. For example, for '1', think of a large, sticky iced bun, the base spiralling round and round, with raisins in it, covered in sweet, white, gooey icing. Now do the rest, using as much visualization as you can muster: imagine how things would look, smell, taste, sound, and so on.

This is your reference list, and you need to know it off by heart.

Having learnt it, look at a pile of at least a dozen odd items collected together by a colleague. The task is to look at the collection of objects for only 30 seconds, and then list as many as possible without making a mistake or viewing the collection again. Most people can manage between five and eight items, if they do not know any memory-enhancing techniques.

7. Chess: of human and artificial intelligence

A few years ago, Deep Blue, a **chess-playing computer**, beat Gary Kasparov, the world's top Grand Master, in a full tournament. This was the long-awaited breakthrough for the **artificial intelligence (AI)** community, who have traditionally seen chess as the ultimate test of their art.

However, despite the fact that computer chess programs can play at Grand Master level against human players, this does not mean they play in the same way. For each move played, Deep Blue investigated many millions of alternative moves and counter-moves. In contrast, a human chess player will only consider a few dozen. But, if the human player is good, these will usually be the right few dozen.

The ability to spot patterns allows a human to address a problem with far less effort than a **brute force approach**. In chess, the number of moves is such that finally brute force, applied fast enough, has **overcome human pattern-matching skill.** Many models of the mental processes have been heavily influenced by computation. It is worth remembering that although there are similarities, computer 'intelligence' is very different from that of humans.

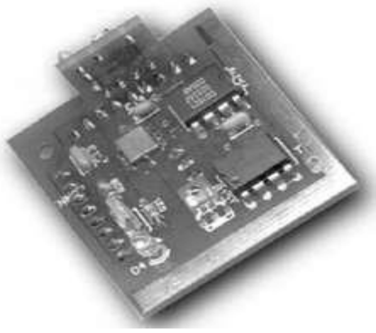
8. Feeling the road

In the **BMW 7 Series** you will find a single haptic feedback control for many of the functions that would normally have dedicated controls. It uses technology developed by Immersion Corporation who are also behind the force feedback found in many medical and entertainment haptic devices. The iDrive control slides backwards and forwards and rotates to give access to various menus and lists of options. The **haptic feedback** allows the user to feel 'clicks' appropriate to the number of items in the various menu lists

9. Smart-Its – making using sensors easy

Building systems with physical sensors is no easy task. You need a soldering iron, plenty of experience in electronics, and even more patience. Although some issues are unique to each sensor or project, many of the basic building blocks are similar – connecting simple microprocessors to memory and networks, connecting various standard sensors such as temperature, tilt, etc.

The **Smart-Its project** has made this job easier by creating a collection of components and an architecture for adding new sensors. There are a number of basic Smart-It boards – the photo on the left shows a microprocessor with wireless connectivity. Onto these boards are plugged a variety of modules – in the center is a sensor board including temperature and light, and on the right is a power controller.


10. Video recorder

A simple example of programming a **VCR from a remote control** shows that all **four translations** in the interaction cycle can affect the overall interaction.

Articulatory problem → Ineffective interaction is indicated by the user not being sure the VCR is set to record properly. This could be because the user has pressed the keys on the remote control unit in the wrong order.

performance translation → the VCR is able to record on any channel but the remote control lacks the ability to select channels, indicating a coverage problem
presentation problem → It may be the case that the VCR display panel does not indicate that the program has been set,
observational error → maybe the user does not interpret the feedback properly.

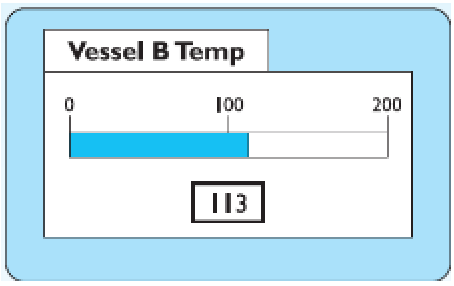
Any one or more of these deficiencies would give rise to ineffective interaction.

11. Industrial interfaces

Industrial interfaces raise some additional design issues rarely encountered in the office.

Glass interfaces vs. dials and knobs

The traditional machine interface consists of dials and knobs directly wired or piped to the equipment. Increasingly, some or all of the controls are replaced with a glass interface, a computer screen through which the equipment is monitored and controlled. Many of the issues are similar for the two kinds of interface, but glass interfaces do have some special advantages and problems. For a complex system, a glass interface can be both cheaper and more flexible, and it is easy to show the same information in multiple forms as shown below



For example, a data value might be given both in a precise numeric field and also in a quick to assimilate graphical form.

Indirect manipulation

The phrase 'direct manipulation' dominates office system design as in fig 3.5.

In a direct manipulation system, the user interacts with an artificial world inside the computer (for example, the electronic desktop). In contrast, an industrial interface is merely an intermediary between the operator and the real world. One implication of this indirectness is that the interface must provide feedback at two levels

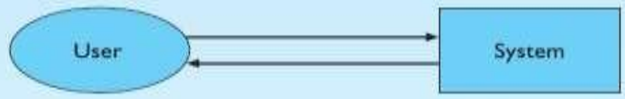


Figure 3.5 Office system – direct manipulation

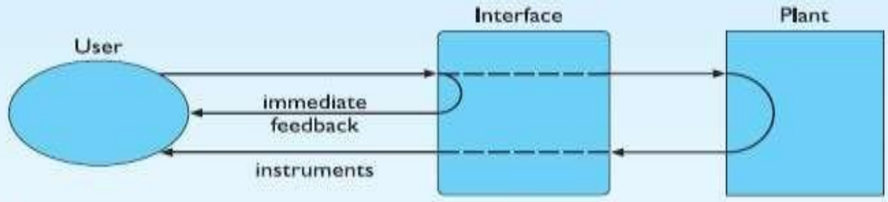


Figure 3.6 Indirect manipulation – two kinds of feedback

As in Figure 3.6 , At one level, the user must receive immediate feedback, generated by the interface, that keystrokes and other actions have been received. In addition, the user's actions will have some effect on the equipment controlled by the interface and adequate monitoring must be provided for this.

The indirectness also causes problems with simple monitoring tasks. Delays due to periodic sampling, slow communication and digital processing often mean that the data displayed are somewhat out of date. If the operator is not aware of these delays, diagnoses of system state may be wrong. These problems are compounded if the interface produces summary information displays. If the data comprising such a display are of different timeliness the result may be misleading.

12. Mixing styles

The UNIX windowing environments are interesting as the contents of many of the windows are often themselves simply command line or character-based programs . In fact, this mixing of interface styles in the same system is quite common, especially where older legacy systems are used at the same time as more modern applications.

It can be a problem if users attempt to **use commands and methods suitable for one environment** in another.

On the Apple Macintosh, HyperCard uses a point-and-click style. However, HyperCard stack buttons look very like Macintosh folders. If you double click on them, as you would to open a folder, your two mouse clicks are treated as separate actions. The first click opens the stack , but the second is then interpreted in the context of the newly opened stack, behaving in an apparently arbitrary fashion! This is an example of the **importance of consistency in the interface.**

13. Bank managers don't type . . .

The safe in most banks is operated by at least two keys, held by different employees of the bank. This makes it difficult for a bank robber to obtain both keys, and also protects the bank against light-fingered managers! ATMs contain a lot of cash and so need to be protected by similar

measures. In one bank, which shall remain nameless, the **ATM had an electronic locking device**. The machine could not be opened to replenish or remove cash until a long key sequence had been entered. In order to preserve security, the bank gave half the sequence to one manager and half to another, so both managers had to be present in order to open the ATM. However, these were traditional bank managers who were not used to typing – that was a job for a secretary! So they each gave their part of the key sequence to a secretary to type in when they wanted to gain entry to the ATM. In fact, they both gave their respective parts of the key sequence to the same secretary. Happily the secretary was honest.

but the **moral** is **you cannot ignore social expectations and relationships when designing any sort of computer system, however simple it may be.**

14. Half the picture?

When systems are not designed to match the way people actually work, then users end up having to do 'work arounds'. **Integrated student records** systems are becoming popular in **universities in the UK**. They bring the benefits of integrating examination systems with enrolment and finance systems so all data can be maintained together and cross-checked. All very useful and time saving – in theory.

However, one commonly used system only holds a single overall mark per module for each student, whereas many modules on UK courses have multiple elements of assessment. Knowing a student's mark on each part of the assessment is often useful to academics making decisions in examination boards as it provides a more detailed picture of performance.

In many cases **staff** are therefore supplementing the official records system with their own **unofficial spreadsheets to provide** this information – making **additional work** for staff and **increased opportunity for error**.

15. Worked exercise (Unit –II)

Discuss the ways in which a full-page word processor is or is not a direct manipulation interface for editing a document using **Shneiderman's** criteria. What features of a modern word processor break the metaphor of composition with pen (or typewriter) and paper?

Answer We will answer the first point by evaluating the word processor relative to the criteria for direct manipulation given by Shneiderman.

Visibility of the objects of interest

The most important objects of interest in a word processor are the words themselves. Indeed, the visibility of the text on a continual basis was one of the major usability advances in moving from line-oriented to display-oriented editors.

Depending on the user's application, there may be other objects of interest in word processing that may or may not be visible. For example, are the margins for the text on screen similar to the ones which would eventually be printed? Is the spacing within a line and the line breaks similar? Are the different fonts and formatting characteristics of the text visible (without altering the spacing)? Expressed in this way, we can see the visibility criterion for direct manipulation as very similar to the criteria for a WYSIWYG interface.

Incremental action at the interface with rapid feedback on all actions

We expect from a word processor that characters appear in the text as we type them in at the keyboard, with little delay. If we are inserting text on a page, we might also expect that the format of the page adjust immediately to accommodate the new changes.

Various word processors do this reformatting immediately, whereas with others changes in page breaks may take some time to be reflected. One of the other important actions which requires incremental and rapid feedback is movement of the window using the scroll button. If there is a significant delay between the input command to move the window down and the actual movement of the window on screen, it is quite possible that the user will 'overshoot' the target when using the scrollbar button.

Reversibility of all actions, so that users are encouraged to explore without severe penalties

Single-step undo commands in most word processors allow the user to recover from the last action performed. One problem with this is that the user must recognize the error before doing any other action. More sophisticated undo facilities allow the user to retrace back more than one command at a time. The kind of exploration this reversibility provides in a word processor is best evidenced with the ease of experimentation that is now available for formatting changes in a document (font types and sizes and margin changes).

One problem with the ease of exploration is that emphasis may move to the look of a document rather than what the text actually says (style over content).

Syntactic correctness of all actions, so that every user action is a legal operation

WYSIWYG word processors usually provide menus and buttons which the user uses to articulate many commands. These interaction mechanisms serve to constrain the input language to allow only legal input from the user. Document markup systems, such as HTML and LaTeX, force the user to insert textual commands (which may be erroneously entered by the user) to achieve desired formatting effects.

Replacement of complex command languages with actions to manipulate directly the visible objects

The case for word processors is similar to that described above for syntactic correctness.

In addition, operations on portions of text are achieved many times by allowing the user to highlight the text directly with a mouse (or arrow keys). Subsequent action on that text, such as moving it or copying it to somewhere else, can then be achieved more directly by allowing the user to 'drag' the selected text via the mouse to its new location.

To answer the second question concerning the drawback of the pen (or typewriter) metaphor for word processing, compares the functionality of the space key in typewriting versus word processing. For a typewriter, the space key is passive; it merely moves the insertion point one space to the right. In a word processor, the space key is active, as it inserts a character (the space character) into the document. The functionality of the typewriter space key is produced by the movement keys for the word processor (typically an arrow key pointing right to move forward within one line). In fact, much of the functionality that we have come to expect of a word processor is radically different from that expected of a typewriter, so much so that the typewriter as a metaphor for word processing is not all that instructive. In practice, modern typewriters have begun to borrow from word processors when defining their functionality