Module - I: Introduction

Notes

Course Contents:

- Introduction to Interaction Design
- I/O Channels-Memory- Reasoning and problem solving
- The Computer: Devices- Memory- processing and networks
- Interaction: Models – frameworks – Ergonomics – styles- elements interactivity

Key Learning Objectives:

At the end of this module, you will be able to:

1. Understand basic of Interaction Design
2. Illustrate three components with respect to human: I/O Channels, memory and processing
3. Analyse various parts of the computer like devices, types of memory, networking and processing of information.
4. Evaluate various interaction models, frameworks, styles and elements

Structure:

Unit 1.1: Introduction to Interaction Design

1.1.1 Introduction

Unit 1.2: I/O Channels-Memory-Reasoning and problem solving

1.2.1 Introduction

1.2.2 I/O Channels

1.2.3 Human Memory

1.2.4 Reasoning and problem solving

Unit 1.3: The Computer

1.3.1 Introduction

1.3.2 Devices

1.3.3 Memory

1.3.4 Processing

1.3.5 Network

Unit 1.4: Interaction

1.4.1 Introduction

1.4.2 Models

1.4.3 Framework

1.4.4 Ergonomics

Notes

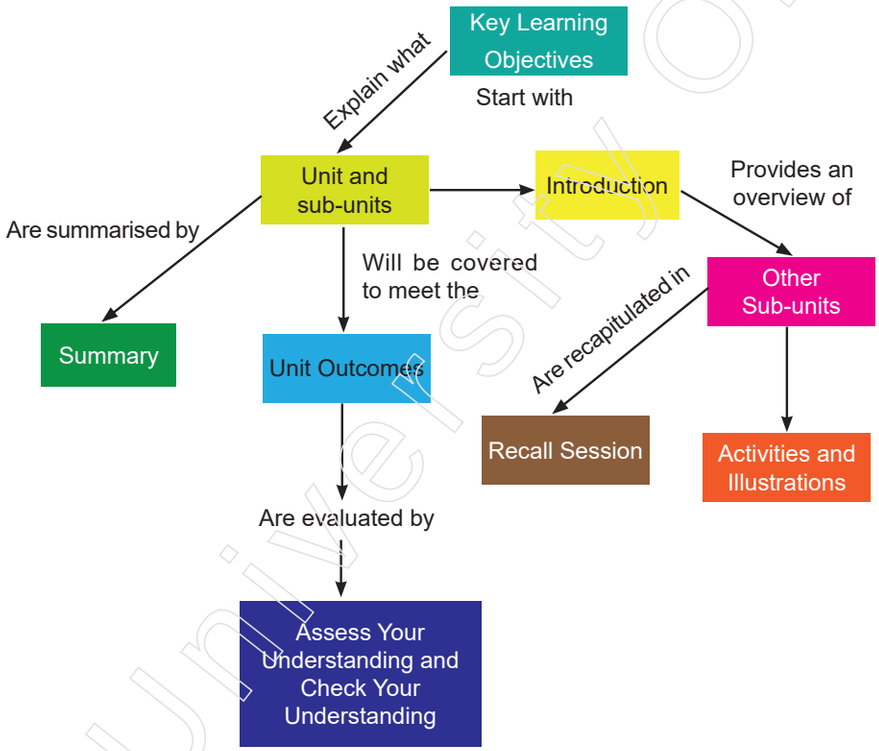
1.4.5 Styles

1.4.6 Elements

1.4.7 Interactivity

1.4.8 Paradigms

Learning Map:



Unit - 1.1: Introduction to Interaction Design

Notes

Unit Outcome:

At the end of this unit, you will be able to:

1. Understand the basic of Interaction Design

1.1.1 Introduction

Human Computer Interaction (HCI) is a multidisciplinary area. It studies design, implementation and the way users interact with computing devices. Primarily, HCI focused on only computers but later on it has expanded to all computing devices. The main goal of HCI is to maintain simple interaction between human and computing devices. Hence, we can say that human, computers and the way they work together are the three important components of HCI.

HCI is implemented in all the domains. Following are domains where HCI has implemented with great importance:

- Computer Science
- Psychology
- Sociology
- Industrial design

The purpose of HCI is to learn various approaches of designing user friendly interfaces or interactions. Interaction Design (ID) is the most important part of HCI. ID is about design of interactive system as well as interaction itself.

Let us consider one scenario of ceiling fan and table fan shared by 4 people in the same room. When table fan is shared then everyone will try to rotate it towards him or they will keep changing their seating arrangement but in case of ceiling fan, it will be fixed so no one can do anything. So here the design of fan affects the way of interaction.

Hence, we can say that design is not only about the producing any physical or computing device but it is about understanding and choosing how it is going to affect the way people work. In addition to physical & computing devices design also provides manuals, tutorials and online help system.

Summary:

- Interaction Design is a design of interactive system and interaction itself.
- This is not only about providing physical or any computing device but it is about understanding and choosing how the design is going to affect the way people work.

Activity:

1. Provide real life example of interaction design and explain how it affects the way people work.

Notes

Unit - 1.2: I/O Channels-Memory-Reasoning and Problem Solving

Recall Session:

In the previous unit, you studied about:

1. Basic of interaction design
2. The way design affects the working of people.

Unit Outcome:

At the end of this unit, you will be able to:

1. Understand three components: input-output, memory and processing.
2. Process the information using problem solving and reasoning.

1.2.1 Introduction

In this unit, you will be introduced to some of the basics of HCI. The main character in any interactive system is user who is nothing but human. Hence, it is important to know about user requirement as well as their capabilities and limitations.

Under this unit you will learn cognitive psychology which defines;

- How humans perceive surrounding world (I/O Channels).
- How human store and process the information
- How humans solve the problems.

1.2.2 I/O Channels

Person can interact with the world through received and sent information which is defined as input and output respectively. During the interaction with the computer:

- Input: response or information provided to the computer by the user.
- Output: response or information provided by the computer to the user.

Initially, the use of input/output terms may cause confusion so let us consider the term “channel”.

In the human body there are five senses as follows:

1. Sight
2. Hearing
3. Touch
4. Taste
5. Smell

Similarly, in human body there are number of effectors. Such as:

Notes

1. Limbs
2. Fingers
3. Eyes
4. Head
5. Vocal system

Out of the five senses, first three plays very important role in HCI. These 3 senses act as a channel for human to receive the information from surrounding hence you can call it as input channel of human.

In the same way, effectors act as a channel for human to provide some information to the world. Hence, effectors can define as output channel of human.

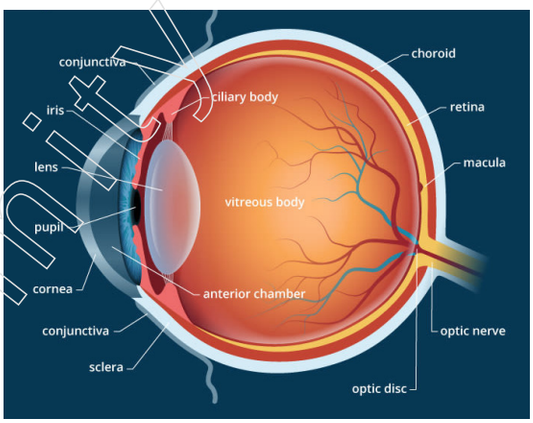
Imagine you are using personal computer (PC) with mouse, keyboard and speaker. Suppose you are creating PowerPoint on your PC. During your interaction with the system:

- You receive some information by using sight from what appears on screen.
- You also receive information by hearing some kind of beep after doing any wrong action.
- With the help of touch and fingers you can click on mouse to provide some information to your PC.

In the above example sight and hearing act as a input channel for you; at the same time fingers and touch plays the role of output channel.

Let us discuss I/O channels in detail.

1. **Vision:** For vision, eye is the main physical receptor. Vison begins with the light and with the help of eye you can receive light and transform it into electrical energy. This electrical energy is pass to the brain.



The Human Eye

As shown in above figure, the human eye has number of important components.

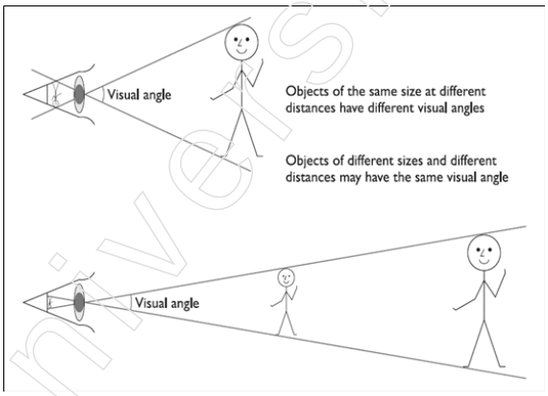
Notes

- **Cornea**: This is the entry point for light. Also important for focusing of light.
- **Lens**: With the help of lens, eye can focus on either near or distant objects.
- **Retina**: Retina is a light sensitive and consist of two types of photoreceptors: rods & cones. Rods are highly sensitive to the light hence you can see under a low level of illumination.
- **Blind spot**: this is the point through which optic nerve enter into the eye.

With the help of structure of the eye you can understand physical mechanism of eye but visual perception is more important than physical mechanism. With the help of visual perception, you will understand how you perceive size and depth as well as brightness and color.

Perceiving size and depth: To perceive size and depth you must consider how images appear on retina. In the reflected light, upside down image form on the retina. With the help of visual angle, size of the image can be specified.

Visual Angle can be calculated as shown in following figure



Visual Angle

A line from top of the object to a central point of the front of the eye and second line from the bottom of the object to the same point; the angle form between these two lines is called visual angle.

Objects of the same size at different distances have different visual angles.

Objects of different sizes and different distances may have the same visual angle.

Perceiving brightness: Brightness is the reaction to the levels of light. It is affected by luminance. Luminance is the amount of light emitted by an object. Luminance can be measured using a photometer. If brightness is high then luminance must be high and vice versa.

Imagine you are using cell phone in day light where the brightness of sun is high and luminance of cell phone is low then you won't be able to see anything in cell phone. Similarly, if you are using cell phone after sunset and luminance is also low then you can access the contents from cell phone because then brightness of daylight is low.

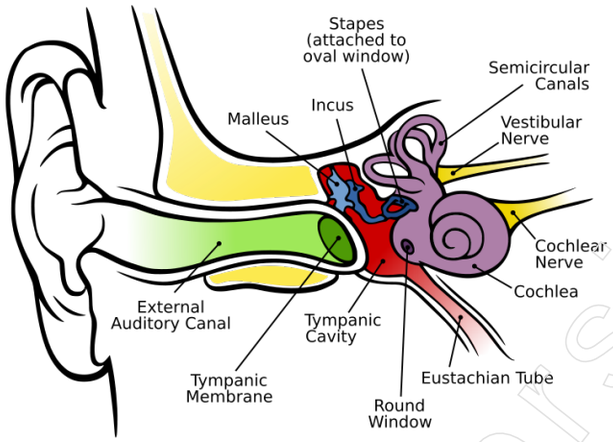
Perceiving color: Eye perceives color because of the cones. Cones are sensitive to the lights of different wavelength. There are three types of cone each sensitive to different color(red, green, blue).

Notes

2. **HEARING:** Hearing starts with the vibrations in the air or sound waves. The ear receives these vibrations and forward them to the auditory nerves through various stages.

The structure of ear has divided into three parts:

- Outer ear
- Middle ear
- Inner ear



A model of the structure of the ear

As shown in above figure, outer ear is the visible part of the ear. It consists of two parts: pinna and auditory canal. Pinna is a structure attached to the head and auditory canal is a passage through which sound waves pass to the middle ear. Pinna and auditory canal help to amplify some sounds.

Middle ear is the small cavity connected to the outer ear with the help of ear drum and inner ear using cochlea. Cavity is consisting of ossicles which is smallest bone in the body. Sound wave pass through auditory canal and vibrate the ear drum which vibrates ossicles results in transmission of vibration to the cochlea and inner ear.

Sound has number of characteristics. The frequency of a sound is called pitch.

Low frequency produces low pitch and high frequency produces high pitch. Loudness is proportional to the amplitude of the sound; the frequency is constant. Timbre is nothing but type of sound. Sound with same pitch and loudness but made using different instrument will have different timbre.

Human ear can hear frequencies from about 20 Hz to 15 kHz.

3. **Touch:** Touch provides important information from environment. When you touch something hot, you get warning signals and you take out your hand or leg back. Hence, we can say that touch also gives us feedback. Whenever you touch something, you receive stimuli from the skin. Skin is consisting of three types of sensory receptors:

- Thermoreceptors: respond to heat and cold
- Nociceptors: respond to intense pressure, heat and pain
- Mechanoreceptors: respond to pressure

Notes

Whole body consist of such a receptor with some high sensitivity and low sensitivity.

4. **MOVEMENT:** Here you will understand how movements affect our interaction with computer. Imagine you are pressing a button as a response to the question. Though this is the simple action, it consists of multiple processing stages. Sensory receptors receive question and pass it to the brain. Brain will process the question and generates appropriate response. According to the generated response, respective muscles will take the action. Each of these stages takes the time. This time is divided into reaction time and movement time. Movement time is typically depending on physical characteristics of the subject; for example, age and fitness whereas reaction time depends on input channel through which stimulus (question in case of our example) is received.

   A person can react to a

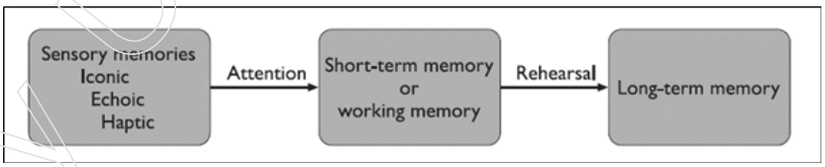
   - Auditory signal in approx. 150ms
   - Visual signal in 200ms
   - Pain in 700ms

While considering speed, the accuracy in movement is also important factor in design of interactive system.

1.2.4 Human Memory

Memory or memory function is divided into three parts as shown in following figure

- Sensory buffers
- Short term memory/ working memory
- Long term memory



A model of the structure of the memory

Sensory Memory: It exists for each sensory channel as follow:

- **Iconic memory for visual stimuli**: e.g. after watching a movie, you can recall or remember and visualize all those shots from movies. This is because of iconic memory. After movie if you went to art exhibition, then whatever you see over there that will get overwritten over previous data in iconic memory.
- **Echoic memory for aural stimuli**: e.g. imagine while reading a newspaper, if someone ask you question then you ask that person to ask it again just to make sure that whatever you heard at first was correct or not. This is the example of existence of echoic memory. It allows play back of information.
- **Haptic memory for touch**: e.g. in a party you spent time with your friends' kitten. After coming back to home, you can remember for few days how soft that kitten was. This is because of haptic memory.

Notes

- These memories can be overwritten by new information. Hence, information received by sensory memories should be quickly passed to more permanent memory to avoid overwritten or loss of information.

- **Short term Memory:** Consider you want to perform 35+55; for this first you will perform 5+5 and then 5+3. Here you have to remember the result of 5+5 while computing 5+3. Hence, to store intermediate stages which can be used later short-term memory is used.

  Let us consider you are watching series on Netflix. To understand episode 2, you have to remember the episode 1. Here short-term memory plays its role.

  You can access short term memory rapidly in 70ms. However, it can be lost rapidly. It means, short-term memory holds information temporarily for 200ms. It has limited capacity.

- **Long term Memory:** Benefits of long-term memory over other two memories:
  - It has huge capacity; not unlimited.
  - Slow access time; approx. 10th of second.
  - Forgetting occurs slowly
  - Useful for long-term storage of information.

  Long term memory is divided into two types:

  a. Episodic memory: Episodic memory stores the information in serial form.

  b. Semantic memory: Information stored in semantic memory is derived from episodic memory.

  Long-term memory is related to three main activities:

  1. Storage or remembering of information
  2. Forgetting
  3. Information retrieval

1.2.4 Reasoning and Problem Solving

Till now you studied input and output channels of human system and the way information stored in human system. Now you will understand how it is processed.

- **Reasoning:** Reasoning is the process through which we conclude with the help of knowledge we have. Reasoning is classified into three types which we use in day to day life:

1. **Deductive**

   It is the process of reasoning from one or more statements to derive logical conclusion.

   Example: all girls like black dress. Tina is a girl.

   Therefore, Tina likes black dress.

   Example:. if it is raining then the umbrella is dry.

   It is raining.

Notes

Therefore, the umbrella is dry.

Though this second example conflicts our knowledge, it is true in the world of deductive reasoning.

2. Inductive

Inductive reasoning is a logical thinking used to form generalizations with the help of incidents you have experienced, observations made by you or the facts which you know.

In inductive reasoning, user is providing set of evidence to derive the conclusion.

Example: The left-handed people I know, uses left hand to hold the scissor.

Therefore, all the left-handed people use left hand to hold the scissor.

In the above example, with the help of fact you know, you are forming generalized statement for everyone.

3. Abductive

With the help of abductive reasoning, you can derive explanation for the events you observe.

Example: suppose you know that Sanket always comes late night at home when he has been partying. If you see Sanket coming mid night at home, you may infer that he has been partying. But this unreliable since there can be another reason for coming mid night at home: urgent office work, for example.

Imagine you know baby cries whenever he is hungry. If you heard baby is crying you can derive the conclusion that baby must be hungry. But this can be a false conclusion. Baby can cry because of other reasons too like stomach pain.

Problem solving

If reasoning means deriving new information from known facts, problem solving is the process of finding a solution for unknown task using the knowledge we have. There are two ways for human problem solving:

a. Gestalt Theory

Gestalt theory is based on trial and error. However, Gestalt school considered it as insufficient explanation for human problem-solving approach. Instead, school claimed there are two types of problem solving: productive and reproductive.

Reproductive derived from previous experience but productive includes restructuring of the problem.

b. Problem space Theory

Newell and Simon proposed that problem solving centers on the problem space. The problem space consists of problem states. By generating the correct states using operator problem can be solved. Every problem has initial and goal state. User uses operators to move from one state to another state.

For example: you want to travel from Mumbai to Shimla. In this example your initial state is Mumbai and goal state is Shimla. To reach from initial to goal state you can

Notes

apply various operators like time, cost and distance (for this example only). If you want to reach in less time and cost is not an issue then you can book flight.

If you want to reach fast but can spent more money then you can book train ticket. If time is not an issue then you can travel by road, in that you can take shortest route for optimized journey.

In this way you can apply various operators to reach goal state.

Consider another example. Initially your table is present at the center of the room and you want to move it near window. In this example, initial state is at the center and goal state is near window. To reach to the goal state you can apply operators: either you can lift the table, drag it or push it. You know to lift something; it must be light but the table is heavy. So, the new sub goal is to make the table light. Hence, your new operators will be removing drawers, removing files from the tables.

Summary:

- Input output channel for the human system are sight, hearing, touch and movement.
- Human memory is divided into three parts: sensory memory, short-term memory and long-term memory.
- Reasoning is classified into three types: deductive, inductive and abductive.
- Reasoning is deriving facts from known facts whereas problem solving is forming the facts from unknown facts.
- Problem solving can done by either gestalt theory or problem space theory.

Activity:

Identify initial and goal state, operators, sub goal states for the following scenario:

1. Imagine there is one closed room. Inside the room, there is a monkey with the box on the ground. Bananas are at the center attached to the ceiling. How monkey will get the bananas?
2. 4 queen problem.

Distinguish following form are deductive or inductive?

1. All football players are tall. Rahul is a football player. Therefore, Rahul is a tall.
2. The scholar students I know use to meditate early in the morning. Therefore, all the scholar students meditate early in the morning.
3. Fatty people don’t eat junk food. John is a fatty. Therefore, John don’t eat junk food.

Differentiate short-term and long-term memory.

Notes

Unit - 1.3: The Computer

Recall Session:

In the previous unit, you studied about:

1. Types of I/O Channels.
2. Types of human memory.
3. Basic of reasoning and its types.
4. Two types of problem-solving approaches.

Unit Outcome:

At the end of this unit, you will be able to:

1. Understand I/O devices of Computer.
2. Summaries types of computer memory
3. Illustrate processing of information and how all the devices attached in the network.

1.3.1 Introduction

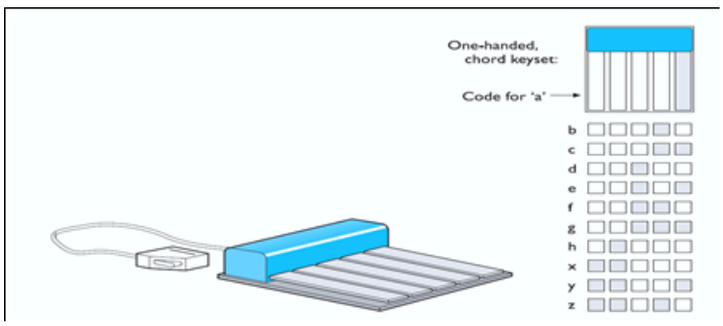
In order to understand the human computer interaction, we must have knowledge of both human as well as computer. In previous unit you studied the I/O channels, memories and problem-solving approach for human. In this unit you will learn about computer system. This unit will cover various I/O devices, memory, processing and networking of computer system.

1.3.2 Devices

Text entry devices: If you want to write letter, application or email using computer, the main task is entering words using keyboard. Keyboard is the most common input device. There are various keyboards:

a. **The alphanumeric keyboard:** The layout of digits and letters are fixed on the keyboard. Only non-alphanumeric keys vary.

b. **The chord keyboards:** These keyboards are varying from alphanumeric keyboards. The chord keyboard uses only few keys, four or five to generate single letter by pressing one or two at a time.

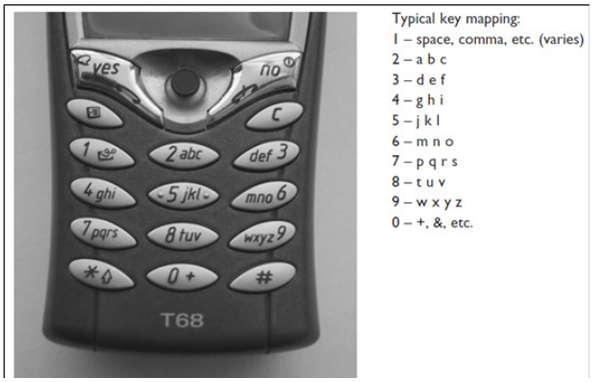


The chord keyboard

Notes

The above figure shows structure and key combination for few letters. This keyboard is very small in size but difficult to use.

c. **Phone pad:** When you want to use SMS text messaging, phone pad is important to enter the text. Following figure shows the phone pad.



Mobile pad

Above diagram also shows mapping of digits to letter on mobile key pad. If you want to enter 2, you have to press 2 for four times. When you press 2 for 1 time, it prints a. when you press 2 for 2 times, it prints b. when you press 2 for 3 times, it prints c. In this way you can print numbers and letters.

Positioning, pointing and drawing: In computing devices, positioning, pointing and drawing are the major function which are done with the help of mouse. The structure of mouse varies according to the computing device. For example, traditional mouse is replaced by touch pad in laptop but the function is same.

a. **The mouse:** Mouse is the major component of the desktop system. It is a small palm size device with weighted ball at the bottom. As the mouse moved on table top, ball also rotates on table top. With the help of this ball motion can be detected. This detected motion sent to computer through wire attached to the mouse or in case of wireless through infrared waves. Mouse moves the pointer on the screen which is called cursor.

b. **Touch pad:** Touch pad is a touch sensitive tablet of size 2-3 inches square. Touch pad can replace the mouse of desktop system. Touch pad works by stroking a finger over its surface instead of scrolling the ball.

c. **Joystick:** It is an indirect input device with very small size. It is a small palm size box with stick sticking on it. The movement of stick causes the movement in the cursor on the screen. Joysticks are not expensive and handy hence, majorly they found in computer games.

d. **Touchscreen:** Touchscreen allows user to point and select objects on the screen direct than the mouse. Touchscreen detects presence of fingers or stylus. Touchscreen is very fast and more suitable in hostile environment because it is dust proof and damage proof.

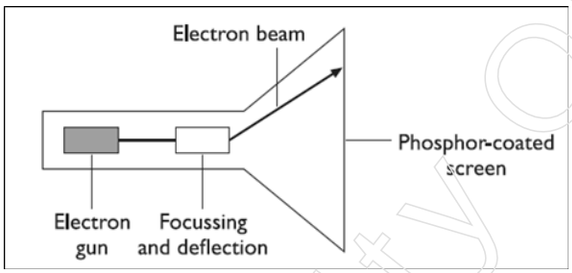
But the drawback is, finger leaves greasy marks on it. Sometimes selection of small area become very difficult.

Notes

e. **Stylus:** To avoid greasy marks on touchscreen and for more accurate positioning stylus is used. Stylus is small pen like plastic stick used to select or draw on the touchscreen.

Display devices

a. **CRT:** CRT stands for cathode ray tube.



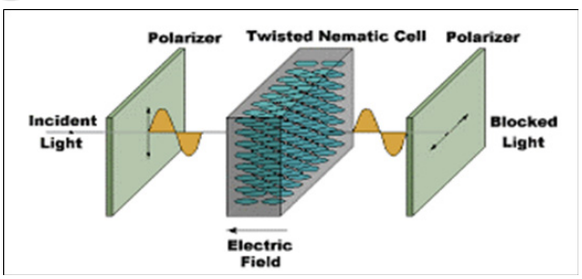
CRT Screen

Above figure indicate the structure and components of CRT screen.

The stream of electrons is emitted from electron gun and focused and directed by magnetic field. As the beam hits the phosphor-coated screen, the phosphor is excited by electrons and glows. The electron beam is scanned from left to right and come back to scan new line from top to bottom.

It is a cheap display device. It has fast enough response time for rapid animation. However, the drawback is CRT is bulky because of electron gun and focusing components behind the screen.

b. **LCD:** LCD stands for Liquid Crystal Display. The computing devices having flat plastic screens uses LCD technology. These devices are smaller, lighter and consumes less power as compare to CRTs. They are also called as flat panel display.



Working of LCD

In this, a thin layer of liquid crystal is sandwiched between two glass plates. The top plate is transparent and polarized while bottom plate is reflective. External light passes through the top plate and is polarized, which means that it only oscillates in one direction. This then passes through the crystal, reflects off the bottom plate and back to the eye, and so that cell looks white. When a voltage is applied to the crystal, via the conducting glass plates, the crystal twists. This causes it to turn the plane of polarization of the incoming light, rotating it so that it cannot return through the top plate, making the activated cell look black.

Notes

1.3.3 Memory

a. **RAM and Short-Term Memory:** The most currently active information is stored in silicon chip which is nothing but Random Access Memory (RAM). RAM can be classified on the basis of precise access time, power consumption and characteristics.

RAM is volatile means contents are lost when power is turned off. Hence, it is called as Short-Term Memory.

b. **Disks and Long-Term Memory:** Long term memory is consisting of disk, most probably of small tapes for backup purpose. Disk is divided into two types: magnetic disk and optical disk.

The commonly used storage media is floppy disk and hard disk. These two are coated with magnetic material. Capacity of floppy disk ranges between 300kbytes to 1.4 Mbytes. Floppy disks are removable hence you can number of floppy disks with you.

The capacity of hard disk ranges between 40 Mbytes to several Gbytes. Disk has two types of access time.

- Time taken to find the right track on the disc.
- Time taken to read the track.

Different types of large media are also available. Optical disk uses laser light to read and write the information on the disk. CD-ROM also uses same technology as compact audio disc. The capacity of CD-ROM is around 650 megabytes but cannot utilize all of it.

	STM Small/fast	LTM large/slower
Media	RAM	Hard disk
Capacity	256 Mbytes	100 Gbytes
Access Time	10 ns	7 ms

Comparison between STM and LTM

1.3.4 Processing

User interface can be affected by processing speed. You to consider this effect while designing the interactive system. Due to processing speed, two types of fault can take place:

- Fault when system is too slow.
- Fault when system is too fast.

In addition to above two faults you have to consider functional fault also. When system or part of the system does not work properly then it leads to functional fault.

When system is too slow, it increases waiting response time of user. And when system is too fast, user won't be able to match the speed.

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

Notes

- **Computation bound:** Computation bound is rare but possible. Imagine you are using find/replace in large document. The system should be designed so that there won't be long delays in the middle of interaction and user will get idea how job is processing.

  Sometimes system shows warning message "this may take some. Do you want to continue(Y/N)?"

- **Storage channel bound:** In storage channel bound, the memory access speed can be affected by interactive performance. To reduce this effect, it is possible to trade of memory against processing speed.

  Consider compressed data takes less space to store and faster to read. But the condition is it must be compressed before storage and decompressed during retrieval. Hence, increase in memory access speed increases processing time.

  If data is written more than it is read; one can choose technique which is expensive to compress but simple to decompress.

- **Graphics bound:** For many modern interfaces, this is the main issue. This issue can be addressed by some clever coding which can reduce time taken by graphics operations. Now a days, many computers have special purpose graphics card which handles most of the graphics operations and reduces the load of main processor.

- **Network capacity:** In many organization, computer and computing devices like printer, scanner is connected to each other by forming a network. With the help of network computers can share the files easily. The main limitation of this network sharing can be memory of system than speed of network.

1.3.5 Networks

In many organizations, instead of using standalone computer they uses computer linked in the network. Networking provides benefits over standalone like networking allows to communicate between different parties. As well as you can access resources remotely.

As per the research, most of the people buy computers to connect to the internet. Using a network, sometimes computers over large distance try to communicate with each other. In this case, transmission time may increase which results in increase in response time. There may be a huge delay in response time and if user is not aware of what is going on than user may get frustrated. Hence, it is very important to inform user what is going on over the network.

Summary:

- Keyboard is the main input device to enter the text into computing devices. There are various forms of keyboard with respect to computing device. For desktop you have alphanumeric keyboard and chord keyboard; for mobile you have mobile pad.
- In alphanumeric keyboard, position of alphabets is fixed. While in chord keyboard, there are very few keys: 4 to 5 keys. At a time 1 or 2 keys pressed to generate single alphabet.

Notes

- For positioning, pointing and drawing you can use mouse. The forms of mouse vary according to its applications. For gaming you use joystick, for touchscreen you use stylus and for laptop you can use touchpad.
- Display devices are based on two technologies: CRT and LCD.
- The memory of a computer is divided into two types: RAM and Disks.
- RAM stands for random access memory which is short term memory and hence it is volatile.
- Disk is a long-term memory and hence it can be categorized as non-volatile memory.

Activity:

1. Compare CRT and LCD.

Notes

Unit - 1.4: Interaction

Recall Session:

In the previous unit, you studied about:

1. Various types' keyboards, mouse and display devices.
2. Two types of computer memory
3. Processing and networking

Unit Outcome:

At the end of this unit, you will be able to:

1. Understand different models and framework of interaction
2. Illustrate how interaction takes place between human and system
3. Analyse different styles of interaction.

1.4.1 Introduction

In previous two units, we learned about the human and the computer. In this unit, you will learn how human and system interact with each other. In order to do this user must communicate his requirements to the system. There are number of ways to communicate with the system.

In this unit, you will learn different models, frameworks and styles to interact with the system.

1.4.2 Models

Models helps to understand complex system and their complex behaviour. Here, you learn about models of interaction. For any kind of interaction, 2 participants are necessary. If both the participants are complex to understand then interface is needed to perform communication between them. This interface can fail at any time and because of any reason. With the help of interaction models, you can easily understand what is going on over interface and identify the cause of failure.

Before discussing about the models, consider following terminologies:

- Domain: defines area of expertise and knowledge in some real-world activity.
- Tasks: are operation to manipulate the concepts of domain.
- Goals: desired output from performed task.
- Intention: specific action required to meet the goal.
- Task analysis: identification of a problem space
- System language: core language describes computational attributes of domain.
- User language: task language describes psychological attributes of the domain.

Notes

Now, we will consider most popular interaction model: "Norman's execution evaluation cycle".

This model is popular because it closely understands interaction between human user and computer. The user decides the plan of action and execute on computer interface. When plan or part of plan has been executed, user observes the interface and evaluates result and plan further plan of action.

This interactive cycle is divided into two parts: execution and evaluation. These can be further divided into seven stages as follow:

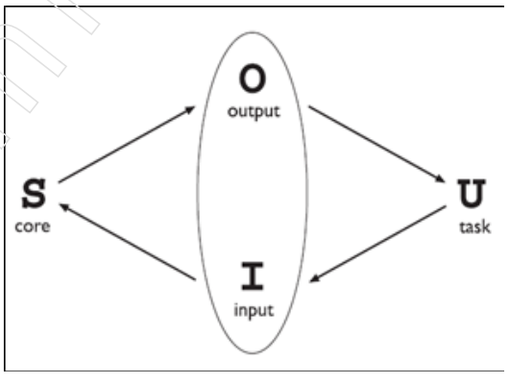
- Establishing the goal: User forms the goal and decides what needs to be done.
- Forming the intention: User forms more specific intention to reach the goal.
- Specifying the action sequence: User decides sequence of actions before execution to achieve the goal.
- Executing the action: After planning goal, intention and sequence of action, user executes the action.
- Perceiving the system state: After execution user perceives new state of system.
- Interpreting the system state: Once new state of system has perceived, it will get interpreted with respect to users' expectations.
- Evaluating the system state with respect to the goals and intentions: If system shows users' goal then user can say, computer has done what it is expected to be done. Hence, interaction is successful.

Otherwise, user has to form a new goal and repeat the cycle.

Imagine you are reading in rooms and ac is on inside the room. Suddenly you started to fill chilled. Hence, you want to turn of the ac. So here you form goal i.e. turn off the ac. From there you form an intention to turn off the ac and you specify the actions required, to reach over and press the switch. When you have executed the actions, you perceive the result either ac is on or off and you interpret this based on your knowledge of the world.

1.4.3 Frameworks

The following figure shows general interaction framework.



The General Interaction Framework

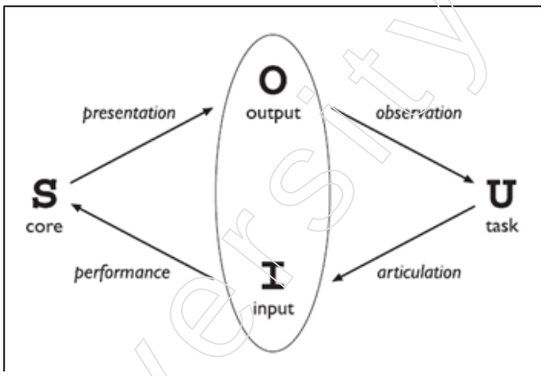
Notes

Framework is divided into four components:

- Input
- Output
- Core/System
- Task/User

Each component has its own language. In addition to system's core language and user's task language, input and output also has its language. Input and output together form an interface. An interface lies between user and system.

The cycle of framework consists of four phases. Each phase is a translation from one component to another component as shown in following figure.



Translations between components

It consists of four translations:

- Observation
- Articulation
- Performance
- Presentation

User begins the cycle by forming a goal and task to achieve that goal. Only through the input user can manipulate the system hence task must be articulated within input. Input language is translated into core language as operations to be performed by the system. The system transforms itself as per the operations and completed execution phase. After execution, evaluation phase begins.

Now system is in new state where it can communicate with the user. System presents the output to the user.

User observes the output and evaluate it with respect to the goal.

In this way, evaluation phase ends and so framework cycle.

1.4.4 Ergonomics

Ergonomics is the study of physical characteristics of the interactions which includes design of controls, physical environment, layout and physical qualities of the screen.

Notes

- **Arrangement of controls and displays:** Arrangement of controls and displays is not important if you are considering excel sheet on personal computer but it is important for safety critical applications where users work under pressure and responsibilities.

  The arrangement of controls and displays include following:

  a. Functional controls and displays: are organized those are functionally related to each other and hence placed together

  b. Sequential controls and displays: are organized to reflect the order of their use.

  c. Frequency controls and displays: are organized according to how frequently they are used

- **The physical environment of the interaction:** Ergonomics also consider design of physical environment. It thinks about where will the system be used? Who will be the user of the system? User will be standing, sitting or moving? All these things are depending on domain of the environment.

  The first basic consideration is the size of the user. Obviously, size is going to change but system should be built in such a way that smaller user must be able to reach all the controls of the system. And large person should not uncomfortable into it.

- **Health issues:** The quality of interaction and the user's performance is affected by the following factors in physical environment.

  a. Physical position: physical position should be comfortable. User should able to reach all the controls and displays easily. It sitting work is more than it should be provided with some support.

  b. Temperature: users can adjust themselves with slight change in temperature. But research has proved that performance decreases at high or low temperature as user are not able to concentrate efficiently.

  c. Lighting: lighting level depend on the work environment. Adequate lighting should be provided to the user to see computer screen without discomfort. Lighting source should be positioned in the way that it will not cause glare which can affect the display.

  d. Noise: excessive noise is harmful to health. Noise should be maintained at a comfortable level at work environment.

  e. Time: the time users spent using system should be controlled. Previously, we have seen the example of CRT. Continuous exposure to CRT is bad for health especially for pregnant women.

- **The use of color:** Color used in the system should as distinct as possible. Colors should follow common conventions. For example, red, green, yellow indicates stop, move and standby respectively at traffic signal. So, in the system red can be used to indicate danger. Green can be used to indicate safe system.

1.4.5 Styles

In interaction system, number of interface styles are there.

Notes

- **Command line interface**

```
sable.soc.staffs.ac.uk> javac HelloWorldApp
javac: invalid argument: HelloWorldApp
use: javac [-g][-O][-classpath path][-d dir] file.java…
sable.soc.staffs.ac.uk> javac HelloWorldApp.java
sable.soc.staffs.ac.uk> java HelloWorldApp
Hello world!!
sable.soc.staffs.ac.uk>
```

Command Line Interface

As shown in above figure, command line interface is most common interface. Through this interface you can provide commands to the computer directly. However, it is difficult to use as there are many commands and parameter, values and syntax of commands.

- **Menus**

```
PAYMENT DETAILS                P3-7

please select payment method:
  1. cash
  2. check
  3. credit card
  4. invoice

  9. abort transaction
```

Menu driven interface

In this interface, set of available options is display on screen as shown in above figure 1.4.4. Users can select these options using mouse, numeric key or alphabetic key. As all the options are displayed on the screen, user don't have to remember and recall the options.

Mostly options are grouped together in meaningful way or hierarchical pattern.

- **Natural language:** The most attractive way to communicate with the computer is natural language. Most of the time user finds commands difficult to remember. But natural language can create ambiguity hence its use will be difficult to provide information.
- **Question/answer and query dialog:** This is the simple mechanism to provide input to the system. The user is asked series of questions mainly with yen/no type of answers and establishes the interaction.

  Query dialog use query to retrieve the information from the system. Queries looks like natural language phrases but they require more specific syntax and knowledge of database structure. While writing the queries, user should provide attributes to the queries.
- **Form-fills and spreadsheets:** Form filling interface basically used for data entry and also useful for information retrieval. As shown in following figure 1.4.5, the form fill interface is just like a form on paper with spaces provided to enter the input value. This interface provides advantage to keep some fields blanks, user can easily move the tabs while filing this form over interface.

Notes

Typical form filling interface

Spreadsheet is group of cells, each of which contain value or formula. Formula is consisting of address of another cell. The user can enter and alter the values as well as formulae.

	Pooches Pet Emporium				
Date	*Description*	*Dog*	*Income*	*Outgoings*	*Balance*
9/2/02	Fees – Mr C. Brown	Snoopy	96.37		96.37
10/2/02	Rubber bones			36.26	60.11
10/2/02	Fees – Mrs E. R. Windsor	7 corgis	1006.45		1066.56
12/2/02	Special order: 7 red carpets			47.28	992.28
16/2/02	Fees – Master T. Tin	Snowy	32.98		1025.26
17/2/02	Beefy Bruno's Bonemeal			243.47	781.79
21/2/02	Fees – Mr F. Flintstone	Dino	21.95		803.74
21/2/02	Special order: 1 Brontosaurus bone			6.47	797.27
28/2/02	Wages – Mr S. H. Ovelit			489.46	307.81

Typical Spreadsheet

Above figure shows typical spreadsheet. Spreadsheet is more flexible and natural.

1.4.6 Elements

- **Windows:** Windows are the area of the screen which work independently. Windows contains text or graphics. It can be moved or resized.

Window

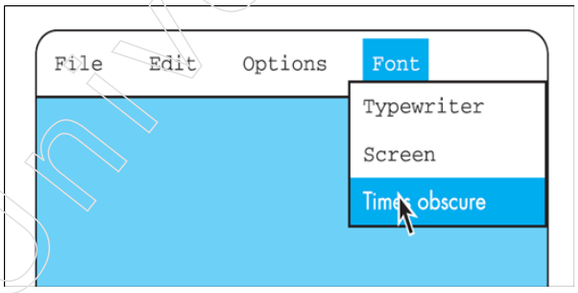
Notes

User can open multiple windows at a time and can perform work simultaneously. Windows have various things associated with them. Scrollbar is one of the things. By using scrollbar user can move the window up and down. Window also possess title bar to identify particular window. Window possess three small buttons at top right corner to minimize, maximize and cancel the window.

- **Icons:** A small picture is used to represent closed window which is called as icon. By using icons, many windows are available on the screen at the same time. After clicking on the icon, window get expanded.
- **Pointers:** Pointer is an important component of interface used to point or select things such as icon. Mouse, touchpad, joystick, stylus used to point the things on screen. Screen provided cursor to the user to point the things or position on window through input devices.

There are various shapes of cursor. You can use different shape to distinguish the mode. For example, normal cursor may be the arrow but if processing is going on you can set the cursor as sand clock.

- **Menus:** Menu represents operations or services provided by the system. Menu consist of menu items. When cursor is moved to the menu, it highlights, menu items. Some menus can be selected using short cut keys. Short cut keys are nothing but the combination of two or more keys of keyboard. By pressing those keys from keyboard, we can select the menus.



Pull down Menu

You can arrange these menus in various forms like pull down menus, fall down menus, circular way, pop up menu, pined up menus and pie menus.

The main problem with menus is deciding what items to possess together and how to group. Menu item should be ordered according to its importance and frequency of use.

- **Buttons:** Buttons are individuals and present on the screen used to initiate specific action. Buttons are divided into various types like push button, toggle buttons, radio buttons and check boxes.

Radio button allows to select any one option at a time while check box button allows to select more than one button at a time. Toggle button is used to toggle between the screens.

- **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

Notes

function of this toolbar is similar to a menu bar, but as the icons are smaller than the equivalent text.

1.4.7 Interactivity

When looking at an interface, it is possible to understand visual element but how they interact with user is difficult part. Interactivity is the feature of an interactive system. All the systems virtually look same as they have same elements like windows, icons, menus, menu bar and toolbar. However, the exact behaviour of these elements differs in different environment. For example, the behaviour of pull down and fall down menu.

Both the menus look same but their behaviour is different. In fact, menus are major difference in Mac OS and Microsoft Windows environment. In Mac OS you have to keep mouse depressed during menu selection; in Windows you click on menu bar and pull-down menu appear on the screen.

In older computer system, you cannot do anything unless computer is ready. But now a day's user can take initiative with different applications. The exceptional part to this is pre-emptive part of the interface.

In pre-emptive part, because of any error or issue or it needs information in order to continue, system take away user control.

Consider the example of dialog boxes. Whenever dialog box appears on the screen, it does not allow you to continue with your work until you close the dialog box with appropriate response. Interactivity is also crucial while dealing with errors.

1.4.8 Paradigms

History of interactive system design provides paradigms for usable designs. Paradigms can be defined as theoretical framework or views of scientific world. It is a understanding of history of HCI. Origin of new computing technologies creates a new perception of human-computer relationship. With the help of paradigms, it is possible to trace some perception in history.

The initial paradigm was batch processing. It was a computing without person. Following are the few examples of paradigms shift:

- Batch processing: impersonal computing
- Timesharing: interactive computing
- Networking: community computing
- Graphical display: direct manipulation
- Microprocessor: personal computing
- www: global information
- ubiquitous computing: a combination of physical and electronic world

We have seen the examples of computing above. These examples provide insight into how interaction between human and computer can be improved. Without replacing the input in interactive system design, we want to maximize the benefit of good idea by

Notes

repeating the benefits in other designs. The issue with these paradigms is they are not well defined. Hence, it is not clear how paradigms are supporting a user in completing the task.

Therefore, repeated use of some paradigms will not generate the design which is more usable.

Summary:

- Models helps to understand complex system and their complex behaviour.
- "Norman's execution evaluation cycle" is the most popular interaction model.
- Framework is divided into four components:
  - Input
  - Output
  - Core/System
  - Task/User
- Framework is consisting of 4 translations:
  - Observation
  - Articulation
  - Performance
  - Presentation
- Ergonomics is the study of physical characteristics of the interactions which includes design of controls, physical environment, layout and physical qualities of the screen.
- Command line interface is the most common interface. However, it is difficult to use as there are many commands and parameter, values and syntax of commands.
- In pre-emptive part of the interface, because of any error or issue or it needs information in order to continue, system take away user control.

Module - II: Design and Software Process

Notes

Course Contents:

- Design and process of design
- Screen design and layout, universal design principles, iteration and prototyping
- Human Computer Interaction in the software process
- Design Rules
- HCI Patterns

Key Learning Objectives:

At the end of this module, you will be able to:

1. Understand design, process of design, layout and principles.
2. Illustrate the steps of human computer interaction in software process.
3. Analyse various design rules and HCI patterns.

Structure:

Unit 2.1: Design

2.1.1 Introduction

2.1.2 Process of design

2.1.3 Screen Design and layout

2.1.4 Iteration and prototyping

Unit 2.2: HCI in the software process

2.2.1 Introduction

2.2.2 The software life cycle

2.2.3 Usability engineering

2.2.4 Iterative design and prototyping

Unit 2.3: Design Rules

2.3.1 Introduction

2.3.2 Standards and guidelines

2.3.3 Golden rules and heuristics

Unit 2.4: HCI Patterns

2.4.1 Introduction

2.4.2 Task centered system design

2.4.3 User centered design

2.4.4 Prototyping

Notes

Unit - 2.1: Design

Unit Outcome:

At the end of this unit, you will be able to:

1. Understand process of design.
2. Understand screen design and layout.
3. Illustrate iteration and prototyping.

2.1.1 Introduction

Design can be defined as achieving goals within constraints. This definition does not explain everything about design but focuses on following points:

Goals: It covers the purpose of design, who will be the end user and why do they want it? For example, if you are designing wireless printer for hotel management system then end users will be hotel management people as well as customers in the hotels who want print their document using their own system or internet.

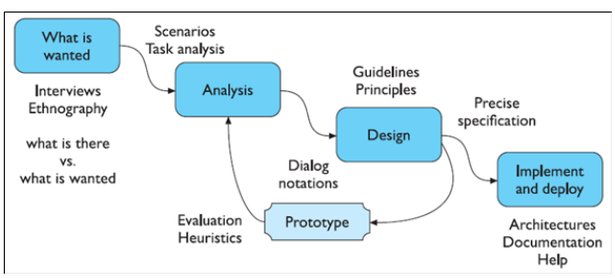
Constraints: constraints cover standards to be followed, material to be used, estimation of cost and time to develop it, health and safety issues to be consider.

Trade off: It is selection of goals and constraints to be ignore so that other goals and constraints can meet the requirement. The more common skill needed in design is to accept the changes and choose most appropriate trade off.

Other than definition, there are also more interesting concepts to understand. Though the raw materials are same, design we produce may be different. This leads to the golden rule of the design. This is obvious in case of physical design. Consider a study table and computer trolley. Both are made up of same raw material that is wood but their designs are different.

In human computer interaction, raw material is human and computer. Hence, it is necessary to understand human and computers. You can understand computers with respect to limitation, capacities, tools and platforms. To understand people, it is important to understand their psychological and social aspects. Therefore, in previous chapter you studied fundamentals of human and computer. The core of interaction design is put the user first, keep the user in the centere and remember the user at the end.

2.1.2 Process of Design



Interaction design process

Notes

Above figure shows interaction design process. It consists of four main phases and iteration loop.

Requirements – What is Wanted?

The first stage is establishing what is exactly wanted. As a prerequisite to this it is necessary to find out what is currently happening. There are number of techniques used for this in HCI such as interviewing people, videotaping them, looking documents and objects that they work with.

Analysis

The results of interview and observation need to be organized in order to bring out key issues and communicate in further stages. Task models are used to record how people carry out various tasks as a part of their work and life.

Design

There is a stage when you shift from what you want to how to do it. There are number of rules, guidelines and design principles that can be used to help with this.

Iteration and Prototyping

Humans are complex and you cannot expect correct designs in first attempt. Therefore, it is necessary to evaluate the design and check how well it is working and what are the area of improvements. In this unit, you will learn various evaluation techniques. One of the evaluation techniques is design on paper, but it is hard to get real feedback without trying it out. Hence, most user interface design involves some form of prototyping which produces early version of system to try with real users.

Implementation and Deployment

Finally, when you are done with your design part, you need to implement it and deploy it. It will include writing code, implementing hardware, writing documentations and manuals. Everything that goes with real system and given to the others will include in this step.

It is easy to think that the goal of iterative stages is to find the usability problems and fix them. In real designs, you may experience that real problem is not to find faults nor to work out how to fix them. The real problem is which usability problem is it worth fixing?

2.1.3 Screen Design and Layout

The basic principles at the screen level are as following:

Ask: What is the user doing?

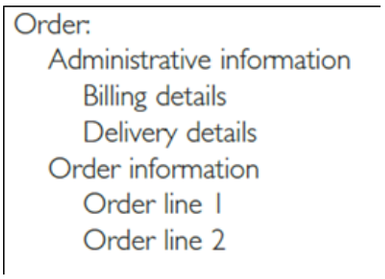
Think: What information is required?

Design: Form follows function: let the required interactions result into layout.

1. **Tools for layout:** There are number of tools available to suggest users' appropriate ways to read and interact with screen or device.

Notes

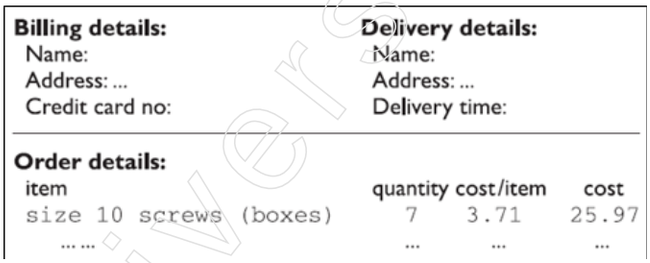
- **Grouping and structure:** Generally, if things logically belong to each other then they are grouped together. This may involve multiple levels of structure.



Ordering screen

Consider the example of ordering screen which is shown in above figure. On ordering screen, you can observe how billing and delivery are grouped together. You can also observe how billing and delivery details are separated from order information.

- **Order of groups and items**



Grouping related items in an order screen

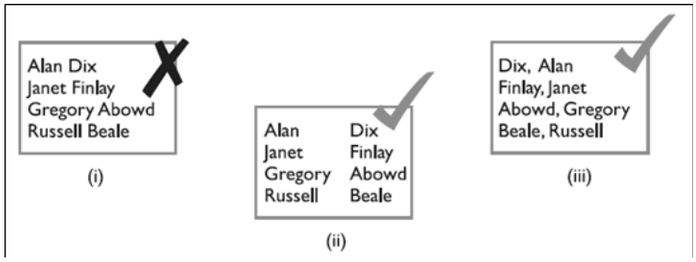
You can observe in above figure, first there is billing details which is then followed by delivery details and followed by order details.

Here, you have to think on is this the right order? What is the natural order for the user? Natural order of the user must match with the screen order shown in figure.

Data entry form must design in the order tab key moves between the fields.

Decoration: Consider figure, in which design of the screen uses boxes and separating lines for clear grouping. Other decorative features like background color and font color, font style can be used for more attractive decoration.

- **Alignment:** Alignment of list is very important.

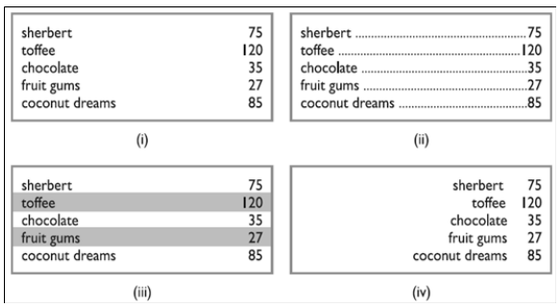


Looking up surnames

Notes

Consider list (i) shown in above figure. In that it is hard to look for someone if you know only surname. To make it easy, list should be as shown in (ii) or (iii) in figure. In figure (ii), there two different columns for name and surname so by looking into surname column you find the person or in figure 2.1.4 (iii), the sequence is surname followed by name hence it will be also easy.

Multiple column list needs to be handled more carefully. Consider following scenario.



Managing multiple columns

In above figure 2.1.5 (i), you can observe how difficult it is for your eye to scan across the rows. This problem can overcome with solution shown in figure 2.1.5 (ii), (iii) and (iv).

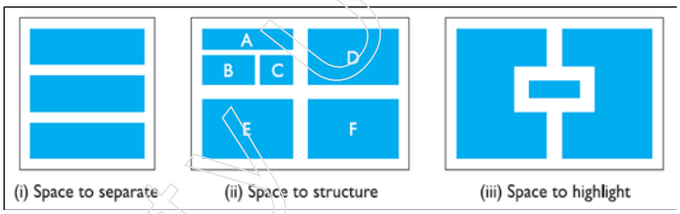
In figure 2.1.5 (ii), it uses line of dots for linking two columns.

Figure 2.1.5 (iii) uses soft tone grays or colors behind rows or columns.

In figure 2.1.5 (iv), text has shifted to right alignment.

Using these kinds of visual ways, you can arrange multiple columns effectively.

- **White space:** White space can be used in several ways.



Using white space in layout

Consider the above figure the dark area represents continuous areas of text or graphics. In figure (i), you can see space is used to separate the blocks just like the gaps between the paragraphs or spaces between the sections in report.

Space can also be used to create more complex structure. In figure (ii), there are main four areas: ABC, D, E and F. ABC is a single area which is further divided into three separate blocks A, B and C.

In figure 2.1.6, space used to highlight. This technique is used in magazines to highlight quote or graphic.

2. User action and control

- **Entering information:** Some of the form-based interfaces are complicated and have difficult screen layout. In form some fields show information to user,

Notes

some fields take input from user where user has to enter the data. Alignment is very important while filling or displaying data. Mostly you can observe text entry boxes are aligned in jagged fashion because field labels are of different size. To overcome this issue small font can be used.

For presenting and entering information logical layout is important. Task analysis techniques help in determining how to group screen items and order in which user would like to read them or fill them.

- **Knowing what to do:** Elements of the screen are either passive or active. Passive elements only give the information. Active elements expecting you to fill them or do something to them. But many times, it is not clear whether elements are active or passive. If buttons and menus designed by everyone look same, then users can easily recognize them. But this is not sufficient. It is also important that labels on menus and icons must be clear. For common actions, standards can help.
- **Affordances:** A situation can create a problem in multimedia applications if someone adopts non-standard style. Because of non-standard style, user will not understand where to click. The physiological idea of affordances says that things may suggest by their shape and other attributes what user can do to them.

  Example: handle affords pulling or pushing. Button affords pushing. These affordances can be used while designing the elements.

3. **Appropriate Appearance**

- **Presenting information:** The way of presenting information on screen depends on
  - Type of information. Information can be of type text, numbers, maps or tables.
  - Technology available to present it. Technology can be character display, line drawing, graphics, virtual reality etc.
  - The purpose for which it is being used.

Name	Date modified
Barplots	21-04-2020 01:52
Basic	20-04-2020 20:27
EDD	23-04-2020 00:32
Histogram	21-04-2020 02:20
import_csv	22-04-2020 23:00
Inbuild datasets of R_1	21-04-2020 00:22
Inbuild datasets of R_2	21-04-2020 00:22
Inbuild datasets of R_3	21-04-2020 01:06
Packages	20-04-2020 21:22

Alphabetic file listing

Notes

Consider the window displayed in above figure. In that window, file listing is alphabetic. It is suitable if you want to check details of particular file. But if the list is ordered by date then it would be difficult to find a particular file. This alphabetic listing is not appropriate to find recently modified file. Therefore, different purpose requires different presentation of information.

- **Aesthetics and utility:** Preferably, an interface with any well-designed item should be pleasing with respect to beauty. Good graphic design and attractive displays can increase users' satisfaction and improves productivity.

  However, beauty and utility may sometimes cannot go hand in hand. This conflict can see in many well-designed posters and multimedia system.

  For example, the backdrop behind the text must have low contrast in order to keep text readable. But most often this is not the case. Graphic designer includes most complex and strong backgrounds because they look good. The results are impressive but completely unusable.

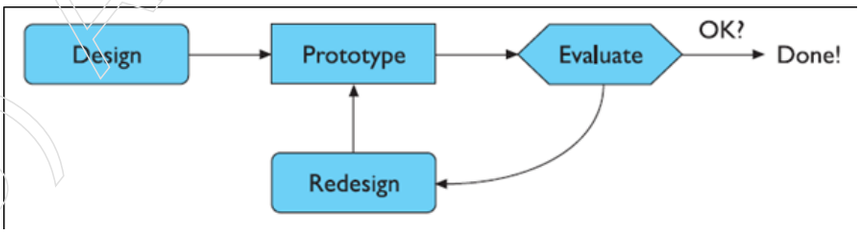
- **Making a mess of it:** One of worst feature in interface is use of color. This is not completely because many monitors only support a limited range of primary colors. Overuse of color can be distracting.

- **Localization/internationalization:** If you are working in different country then maybe you have observed though document is word processed, the text of the document and the file names are in local language. But all the menus and instructions are still in English. The process of making software suitable for different languages and cultures is called localization or internationalization.

2.1.4 Iteration and Prototyping

As humans are complex, design in first attempt will never be perfect. Hence, almost all interaction designs include some form of iteration of ideas. This starts with paper designs and story boards explained to colleagues and users. Its further proceeds with mock-up of physical devices or tools to create prototype versions of software.

Any of these prototypes either paper based or software tool, then evaluated to check whether they are acceptable and is there a need of improvement? This type of evaluation improves designs hence it is called as formative evaluation. Formative evaluation is exactly opposite to summative evaluation. Summative evaluation is performed at the end to check whether product is good enough or not. There are various approaches for evaluation including involving expert to check the guidelines have followed or not. Another approach is involving users. The result of evaluation is mostly a list of faults and problems. This result is followed by redesign of system which is again prototyped and evaluated. This process is shown in following figure.



Role of prototyping

Notes

The end point is when there are no more faults that can economically fixed. Therefore, iteration and prototyping are universally accepted best practise approach for interaction design. But there are some limitations of prototyping.

Prototyping is an example of hill climbing approach. Imagine you are standing somewhere in the open countryside. You walk uphill and keep going uphill as steeply as possible. Eventually you will find yourself at a hill top. This is exactly how iterative prototyping works: you start somewhere, evaluate it to see how to make it better, change it to make it better and then keep on doing this until it can't get any better. But hill climbing does not always work.

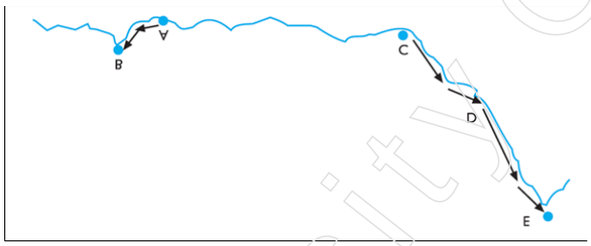


Illustration of Hill climbing approach

Above figure illustrate hill climbing approach. As shown in above figure, if you start at A, you get trapped at local maximum at B. But if you start at C, you move up through D to the global maximum at E.

The problem of getting trapped at local maximum is possible with interfaces. If you start with bad design concept, you may end up with messed version of bad design. Hence, there are two things to remember so that prototyping methods will work.

- To understand what is wrong and how to improve.
- A good start point.

Out of these two second one is most important to avoid local maximum. Using experience and judgment, a good designer guesses an initial good design.

Summary:

- Design can be defined as achieving goals within constraints.
- Though the raw materials are same, design we produce may be different. This leads to the golden rule of the design.
- For presenting and entering information logical layout is important.
- Passive elements only give the information.
- Active elements expecting you to fill them or do something to them.
- The physiological idea of affordances says that things may suggest by their shape and other attributes what user can do to them.
- The way of presenting information on screen depends on

- Type of information. Information can be of type text, numbers, maps or tables.
- Technology available to present it. Technology can be character display, line drawing, graphics, virtual reality etc.
- On purpose for which it is being used.

- Different purpose requires different presentation of information.
- Iteration and prototyping are universally accepted best practise approach for interaction design.
- Prototyping is an example of hill climbing approach.

Notes

Notes

Unit - 2.2: HCI in Software Process

Recall Session:

In the previous unit, you studied about:

1. Design and process of design
2. Screen design and layout
3. Iteration and prototyping

Unit Outcomes:

At the end of this unit, you will be able to:

1. Understand the software life cycle.
2. Understand usability engineering
3. Illustrate iterative design and prototyping.

2.2.1 Introduction

Within computer science, there is another sub-branch that addresses the management and technical issues of the development of software systems called software engineering. The important aspect of software engineering is software lifecycle. Software lifecycle describes the activities for initial concept formation for a software system until its eventual phasing out and replacement.

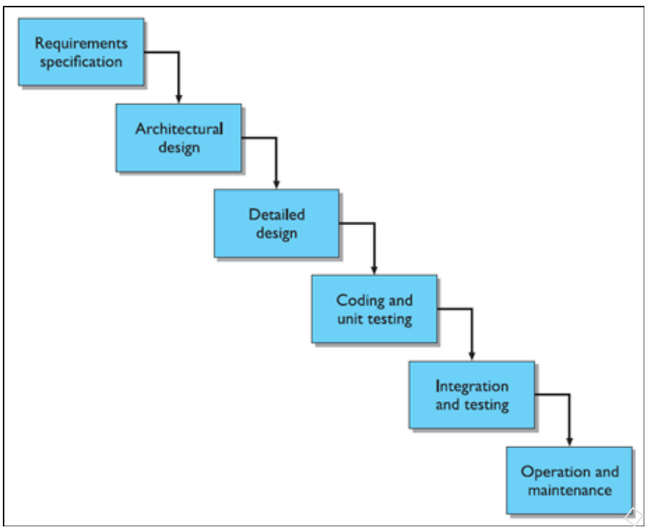
Issues from the HCI which affects the usability of interactive systems are related to all the activities of the software lifecycle. Hence, software engineering for interactive system design is not only about adding one or more activity that fits with existing activity of lifecycle. It is about involving technique that span the entire lifecycle.

2.2.2 The Software Life Cycle

The basic feature of software engineering is to provide the structure for applying techniques to develop software systems. The software lifecycle identifies the activities that occur in software development. In the development of software, there are two main parties: the customer and the designer. The customer uses the product and designer provides the product. Basically, customer and designer are group of people and some people can be both customer and designer. In this unit, the group of people who interact with the design team will refer as customer and term user or end user will refer to those who will interact with designed system.

Notes

- **Activities in life cycle**



The activities in waterfall model of the software lifecycle

Detailed description of the life cycle activities is given in above figure 2.2.1. The graphical representation of lifecycle is like a waterfall model where each activity leads to next activity.

Requirement Specification: In this stage, customer and designer try to find out what the system will be expected to provide. It also involves obtaining information from the customer about work environment or domain of final product. Domain information provides particular function that software product must perform as well as details of the environment in which it must operate.

This phase begins at the start of product development. Though the requirements are from customer point of view, if they fulfil software product, they must be written in a language suitable for implementation. The requirements are initially represented in native language of user. It is important to convert them into executable language which is less natural and more precise in terms of interpretation or semantics. This conversion from native language to executable language is the key to successful development.

Architectural Design: As discussed earlier, requirement specification concentrates on what system is supposed to do. The next activity concentrate on how system provides services expected from it. Architectural design performs decomposition of system into components. This decomposition determines which component provides which service. It also describes interdependencies between separate components.

Many structured techniques are sufficient for capturing functional requirements of the system but there is no technique to capture non-functional requirements. Functional requirements are the services that system must provide in work domain whereas non-functional requirements are the features of the system that are not directly related to actual services provided but related to the way in which services must be provided.

Detailed Design: Architectural design provides a decomposition of a system description which allows separate development of separate component which can be

Notes

integrated later. The components which are not available for immediate integration, the designer must provide detailed description of that component so that they can be implemented in some programming language. Architectural designs provide component description which is clarified by detailed design.

Coding and unit testing: As we discussed in detailed design, design for a component of the system should be in such a form that it can implement in some executable programming language. After coding components can be tested to check its performance. This coding activity can be automated directly from low-level detailed design.

Integration and testing: Once components have been implemented and tested, they must be integrated as described in architectural design. After integration, testing is done to make sure correct behaviour and use of shared resources. It is also possible to perform some acceptance testing with the customer to ensure that system meets their requirements. Only after acceptance of integrated system, the product is finally released to the customer. It is also necessary to certify the system to check whether system meets requirements imposed by some outside authority.

Maintenance: Once product has been released, all work on system considered under category maintenance until new version of product requires complete redesign or the product is phase out completely. Mostly, the lifetime of a product is spent in maintenance activity.

Maintenance involves correction of the errors in system which are discovered after release. Hence, maintenance provides feedback to all other activities in life cycle.

- **Validation and verification:** Throughout the lifecycle, the design must be checked to ensure that it satisfies the high-level requirements agreed with the customer and it is complete as well as internally consistent. These checks are called as validation and verification respectively.

According to Boehm, validation is like designing the right thing and verification is designing the thing right. Various languages are used throughout the design from informal natural language to formal mathematical language. Validation and verification are difficult when carried out using one language.

Verification of a design most often occur within a single life cycle activity or between two adjacent activities. For example, in the detailed design of component of a payroll accounting system, the designer will be concerned with the correctness of the algorithm to compute taxes deducted from an employee's gross income. The architectural design will provide a general specification of the information as an input to this component and output must be information.

The detailed design may also have to change the representations for the information and will almost certainly break up a single high-level operation into several low-level operations that can eventually be implemented. In introducing these changes to information and operations, the designer must show that the refined description is a legal one within its language and that it describes all of the specified behaviour of the high-level description in a provably correct way.

Notes

Validation of a design demonstrates that the customer's requirements are fulfilled. Validation is a much more subjective exercise than verification, mainly because the difference between the language of the requirements and the language of the design. In interactive system design, the validation against HCI requirements is often referred to as evaluation and can be performed by the designer with or without customer.

For a verification of design, proofs are used. They can frequently occur within one language or between two languages. Time constraints and economic implications dictates which proof has to be carried out.

- **Management and contractual issues:** The lifecycle described earlier focused on technical features of software development. In technical discussion, managerial issues of a design such as time constraints and economic forces are not as important. In management, wider perspective is adopted which considers marketability of a system, its training needs, the availability of a skilled personnel or subcontractor and other activities for development of system.

  In managing the development process, the temporal relationship between various activities is important and managerial perspective is described in temporally bound phases. A phase is defined in terms of documentation taken as input to the phase and documentation delivered as output from the phase. So, the requirements phase will take any marketing or conceptual development information, identifying potential customers, as input and produce a requirements specification that must be agreed upon between customer and designer.

  As the design activity proceeds, the customer and the designer must sign off on various documents, indicating their satisfaction with progress to date. These signed documents can carry a varying degree of contractual obligation between customer and designer. A signed requirements specification indicates both that the customer agrees to limit demands of the eventual product to those listed in the specification and also that the designer agrees to meet all of the requirements listed. From a management perspective, it is unacceptable to both designer and customer to delay the requirements specification that long.

2.2.3 Usability Engineering

Usability engineering is the method in the development of software and system. It includes user contribution and assures product effectiveness by using usability requirements and matrices. The focus of usability engineering is knowing exactly what criteria will be used to judge a product for its usability.

Product usability test depends on measurements of users' experience with it. In relation to the software life cycle, one of the important features of usability engineering is the inclusion of a usability specification, forming part of the requirements specification that concentrates on features of the user–system interaction which contribute to the usability of the product. Various attributes of the system are suggested for testing the usability. For each attribute, six items are defined to form usability specification of that attribute.

Notes

Attribute:	Backward recoverability
Measuring concepts:	Undo an erroneous programming sequence
Measuring method:	Number of explicit user actions to undo current program
Now level:	No current product allows such an undo
Worst case:	As many actions as it takes to program in mistake
Planned level:	A maximum of two explicit user actions
Best case:	One explicit cancel action

Sample usability specification for undo with a VCR

Above table provides an example of a usability specification for the design of a control panel for a video cassette recorder (VCR). In this example, we choose the principle of recoverability. Recoverability refers to the ability to reach a desired goal after recognition of some error in previous interaction. The recovery procedure can be in either a backward or forward. Current VCR design has resulted in interactive systems that are difficult to use; the redesign of a VCR provides a good case study for usability engineering. In designing a new VCR control panel, the designer wants to consider how a user might recover from a mistake he encountered while trying to program the VCR to record some television program in his absence. One approach that the designer decides to follow is to allow the user to undo the programming sequence. The backward recoverability attribute is defined in terms of a measuring concept, which makes the abstract attribute more concrete by describing it in terms of the actual product. So, in this case, we realize backward recoverability as the ability to undo an erroneous programming sequence. The measuring method states how the attribute will be measured, in this case by the number of explicit user actions required to perform the undo, irrespective of where the user is in the programming sequence.

The remaining four entries in the usability specification then provide the agreed criteria for judging the success of the product based on the measuring method.

Now level indicates, the value for the measurement with the existing system, whether it is computer based or not.

The worst-case value is the lowest acceptable measurement for the task, providing a clear distinction between what will be acceptable and what will be unacceptable in the final product.

The planned level is the target for the design and the best case is the level which is agreed to be the best possible measurement given the current state of development tools and technology.

In the example, the designers can check out their previous VCR products and people of their competitors to work out an appropriate now level. In this case, no current model allows an undo to return the state of the VCR.

Worst case value should not be less than now level. New product must provide some improvements. Therefore, some usability attributes should provide worst case values that are better than now level.

Notes

1	Time to complete a task
2	Time to complete a task
3	Time to complete a task
4	Ratio of successes to failures
5	Time spent in errors
6	Per cent or number of errors
7	Per cent or number of competitors better than it
8	Per cent or number of competitors better than it
9	Frequency of help and documentation use
10	Per cent of favourable/unfavourable user comments
11	Number of repetitions of failed commands
12	Number of runs of successes and of failures
13	Number of times interface misleads the user
14	Number of good and bad features recalled by users
15	Number of available commands not invoked
16	Number of regressive behaviours
17	Number of users preferring your system
18	Number of times users need to work around a problem
19	Number of times the user is disrupted from a work task
20	Number of times user loses control of the system
21	Number of times user expresses frustration or satisfaction

Criteria by which measuring methods can be determined

Set levels with respect to information on:	
1	an existing system or previous version
2	competitive systems
3	carrying out the task without use of a computer system
4	an absolute scale
5	your own prototype
6	user's own earlier performance
7	each component of a system separately
8	a successive split of the difference between best and worst values observed in user tests

Possible ways to set measurement levels in a usability specification.

Tables represents list of measurement criteria which can be used to determine the measuring method for a usability attribute and possible ways to set worst/best case and planned/now level target. Measurements using usability engineering are defined as

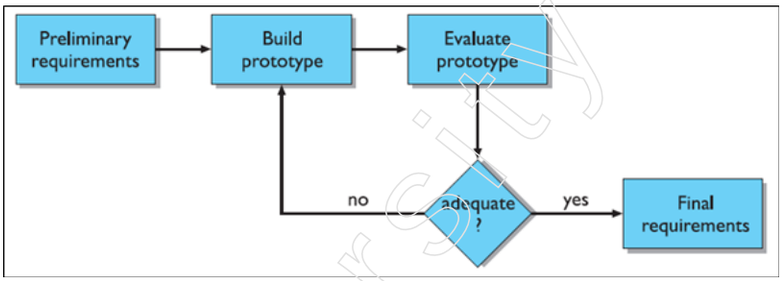
Notes

usability matrices.

2.2.4 Iterative Design and Prototyping

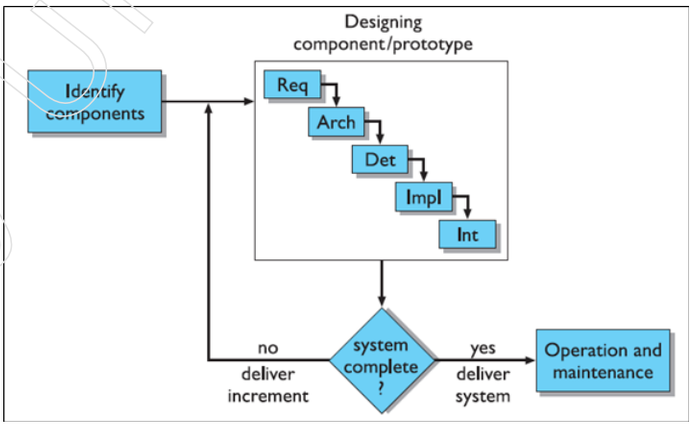
Iterative design is an intentional design process which tries to overcome the inherent problems of incomplete requirements specification by cycling through several designs, incrementally improving upon the final product with each pass. Prototyping is divided into three approaches:

Throw-away: The prototype is built and tested. The design knowledge gain from this process is used to design final product, but actual prototype is rejected. Following figure 2.2.2 gives illustration of throw-away prototyping.



Throw-away prototyping with requirement specification

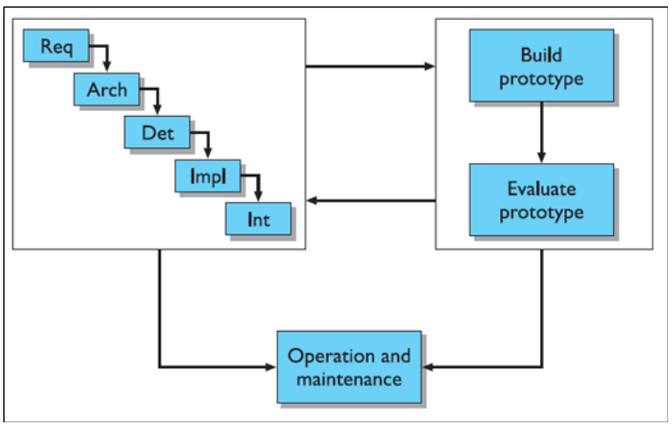
Incremental: The final product is built one at a time as a separate component. The overall design of final system is partitioned into independent and smaller components. The final product is released as a series of products. Each product is consisting of component. This is represented in following figure 2.2.3.



Incremental prototyping within the life cycle

Evolutionary: Here the prototype is not rejected, it provides input for next iteration of design. In this case, it is observed that the actual system is improving from a very limited initial version to its final release. It is shown in following figure 2.2.4.

Notes



Evolutionary prototyping throughout the lifecycle

Evolutionary prototyping also fits in well with the modifications which must be made to the system that arise during the operation and maintenance activity in the life cycles.

Following are the techniques to generate rapid prototypes:

- **Storyboards:** Storyboard is the simple concept of prototype. It is a graphical representation of outside appearance of designed system. Storyboards do not require much computing power to construct. The inceptions of storyboards are in the entertainment world, where a progression of boards generally delineate previews from a planned film grouping so as to get the thought across about the inevitable scene. Similarly, the storyboards provide snapshots of the interface for interactive system design. Evaluating customer or user impressions of the storyboards can determine relatively quickly if the design is heading in the right direction.

  Using modern graphical drawing packages, it is possible to create storyboards with the help of computer not by hand.

- **Limited functionality simulations:** The prototype must have more functionality to demonstrate the work done by application. Storyboards and animation are not sufficient techniques for this purpose. Hence, some portion of the functionality must be simulated by design team. With the help of programming support for simulation, designer can rapidly build graphical and textual interaction objects and provide some function to those objects.

  There are various prototyping tools available for rapid development of simulation prototypes. These simulation tools provide a fast development process for number of highly interactive applications. A popular and successful prototyping tool is HyperCard.

  HyperCard is a simulation environment for the Macintosh line of Apple computers. It is like an animation tool. Also, it has ability to provide more interactive behaviour by attaching script. The script is written in HyperTalk programming language.

  Consider the technique: Wizard of Oz. With this technique, the designers can develop a limited functionality prototype and enhance its functionality in evaluation by providing the missing functionality through human intervention.

Notes

- **High level programming support:** HyperTalk is the example of special purpose high-level programming language. With the help of this programmers can easily program certain features of interactive system. HyperTalk and many similar languages allow the programmer to attach functional behaviour to the specific interactions that the user will be able to do, such as position and click on the mouse over a button on the screen.

  Low-level programming was depending on implementation. It means, programmer supposed to know some details of hardware system in order to control interactive behaviour. On the other hand, these high-level programming languages are not depending on implementation. In order to provide high-level programming support, you need UIMS (user interface management system). The job of a UIMS, then, is to allow the programmer to connect the behaviour at the interface with the underlying functionality.

Summary:

- Within computer science, there is another sub-branch that addresses the management and technical issues of the development of software systems called software engineering.
- Software lifecycle describes the activities for initial concept formation for a software system until its eventual phasing out and replacement.
- The basic feature of software engineering is to provide the structure for applying techniques to develop software systems. The software lifecycle identifies the activities that occur in software development.
- In requirement specification, customer and designer try to find out what the system will be expected to provide.
- Architectural design performs decomposition of system into components. This decomposition determines which component provides which service.
- The components which are not available for immediate integration, the designer must provide detailed description of that component so that they can be implemented in some programming language.
- Only after acceptance of integrated system, the product is finally released to the customer.
- Maintenance involves correction of the errors in system which are discovered after release.
- According to Boehm, validation is like designing the right thing and verification is designing the thing right.
- In management, wider perspective is adopted which considers marketability of a system, its training needs, the availability of a skilled personnel or subcontractor and other activities for development of system.
- The focus of usability engineering is knowing exactly what criteria will be used to judge a product for its usability.
- HyperTalk is the example of special purpose high-level programming language.

Unit - 2.3: Design Rules

Notes

Recall Session:

In the previous unit, you studied about:

1. The software life cycle
2. Usability engineering
3. Iterative design and prototyping

Unit Outcome:

At the end of this unit, you will be able to:

1. Understand the standards, guidelines of design rules.
2. Evaluate golden rules and heuristic.
3. Analyse HCI patterns.

2.3.1 Introduction

Design rules are the rules a designer can follow in order to increase the usability of software product. Rules can be classified along two dimensions based on rule's authority and generality. By authority means, whether rules are followed in design or not or whether it is only suggested. By generality means, whether rules can be applied to many design situations or whether it is focused on limited application situation. Rules can vary in level of abstraction. Some rules hide details of design and some rules are specific. It is also important to determine origin of design rules.

Principles are abstract design rules which are highly general and lower in authority.

Standards are specific design rules with high authority and limited in application.

Guidelines are lower in authority and more general with respect to application.

Principles are derived from the knowledge of psychological, computational and sociological aspects of the problem domain. They are not depending on technology. Instead, they depend on deeper understanding of human in interaction. Hence, they can apply everywhere but not to specific design.

Guidelines are less abstract and more based on technology. As they are very general, designer must know what evidences are there to support them.

While applying standards, designer does not need to know underlying theory in depth. But it is important that underlying theory must be correct because standards are higher in authority.

Design rules would be effective if they adopted in early stages of lifecycle.

2.3.2 Standards

Standards for interactive system design are set by national or international bodies. Standards can apply to either hardware or software used to build interactive system.

Notes

There are different characteristics between hardware and software, which affect utility design standards. Those characteristics are as following:

- **Underlying theory:** Standards for hardware are based on understanding of physiology or ergonomics. The results are well known, fixed and adaptable to design of hardware. On the other hand, software standards based on theories from psychology or cognitive science. They less well formed, still evolving and not easy to interpret in software design language.
- **Change:** Hardware is more crucial and costly to change than the software. Software are designed very flexible. However, in hardware requirement changes do not occur as frequent as in software. Even standards are also more suitable for hardware than software because standards are relatively stable. Hence, any standard institution like BSI (British Standard Institution) and ISO (International Organization for Standardization) first set the standards for hardware and then for software.

  For example, the UK Ministry of Defence has published an Interim Defence Standard 00–25 on Human Factors for Designers of Equipment, produced in 12 parts:

  Part 1 Introduction

  Part 2 Body Size

  Part 3 Body Strength and Stamina

  Part 4 Workplace Design

  Part 5 Stresses and Hazards

  Part 6 Vision and Lighting

  Part 7 Visual Displays

  Part 8 Auditory information

  Part 9 Voice Communication

  Part 10 Controls

  Part 11 Design for Maintainability

  Part 12 Systems

  In above example only last part is related to software design process.

  Consider another example: The international standard ISO 9241, entitled Ergonomic Requirements for Office Work with Visual Display Terminals (VDT)s, has 17 parts. Seven of these are concerned with hardware issues – requirements for visual display, keyboard layout, workstation layout, environment, display with reflections, display colors and non-keyboard input devices. Seven parts are devoted to software issues – general dialog principles, menu dialogs, presentation of information, user guidance, command dialogs, direct manipulation dialogs and form-filling dialogs. One component of ISO 9241, pertaining to usability specification applies to hardware as well as to software.

Notes

2.3.3 Guidelines

Because of incomplete theories, which are base of design of interactive software, it is difficult to generate authoritative and specific standards. Hence, many design rules suggestive and more general guidelines.

The more abstract guideline, the more it is suitable to requirement specification. The more specific guidelines, they are more suitable to detailed design. Up to some extent, guidelines can be automated. Automated guidelines provide direct way of translating detail design specification to actual implementation. Number of various guidelines are published for interactive system design.

Some books and technical report contain catalogues of guidelines. A classic example was a very general list compiled by Smith and Mosier in 1986 at the Mitre Corporation and sponsored by the Electronic Systems Division of the US Air Force. The basic categories of the Smith and Mosier guidelines are:

- Data Entry
- Data Display
- Sequence Control
- User Guidance
- Data Transmission
- Data Protection

Each of these categories is further divided into subcategory which contain particular guideline.

1. **Data Entry**

1.1 *Position Designation*

1.1–1 **Distinctive Cursor**

For position designation on an electronic display, provide a movable cursor with distinctive visual features (shape, blink, etc.).

Exception When position designation involves only selection among displayed alternatives, highlighting selected items might be used instead of a separately displayed cursor.

Comment When choosing a cursor shape, consider the general content of the display. For instance, an underscore cursor would be difficult to see on a display of underscored text, or on a graphical display containing many other lines.

Comment If the cursor is changed to denote different functions (e.g. to signal deletion rather than entry), then each different cursor should be distinguishable from the others.

Comment If multiple cursors are used on the same display (e.g. one for alphanumeric entry and one for line drawing), then each cursor should be distinguishable from the others.

Reference Whitfield, Ball and Bird, 1983

See also 1.1–17 Distinctive multiple cursors
4.0–9 Distinctive cursor

Sample guideline from Smith and Mosier

Above figure represents the guidelines from Smith and Mosier. The Mitre Corporation has taken advantage of this structure and implemented the Smith and Mosier guidelines on a hypertext system, which provides rapid traversal of the network of guidelines to investigate the cross-references and citations.

Notes

2.3.4 Golden Rules and Heuristics

Golden rules and heuristics are useful checklist for good design. There are various golden rules and heuristics such as Nielsen's 10 heuristics, Shneiderman's eight golden rules and Norman's seven principles.

1. **Shneiderman's Eight Golden Rules of Interface Design:** Ben Shneiderman is an American computer scientist. He reveals his eight golden rules of interface design:

   a. **Strive for consistency:** Implementing consistent interfaces means using same design patterns and same sequence of actions for same situations. It includes, right use of color, typography, terminology, commands and menus.

   b. **Enable frequent user to use shortcuts:** Shortcuts are useful for users if they are performing same task many times. May be following features are useful for expert users:

      - Abbreviations
      - Function keys
      - Hidden commands
      - Macro facilities

   c. **Offer informative feedback:** As a designer, you have to tell your users what is happening at every stage of the process. This feedback must be meaningful, relevant and clear.

   d. **Design dialogs to yield closure:** It is important so that users will come to know when they have completed the task.

   e. **Offer error prevention and simple error handling:** It is important to prevent users from making mistakes and if they do then designer must offer them a clear guidance to recover the errors.

   f. **Permit easy reversal of actions:** It is a great relief to find undo option after a mistake is made. Users will feel less anxious and they will explore more options because they know there is an easy way of reversal.

      This rule can be applied to action, group of actions or data entry.

   g. **Support internal locus of control:** This rule gives control to the user so they can feel in charge of the system. Here, user should be initiator instead of responder. According to this rule avoid surprises and interruptions.

   h. **Reduce short-term memory load:** Recognition of information is always simple and better than recalling it. In order to recognize easily, it is necessary to keep the interface simple and consistent; following patterns and standard.

      These eight golden rules are very useful and their application will help many design projects.

2. **Norman's Seven Principles for Transforming Difficult task into simple one:** Norman's execution evaluation cycle has explained seven stages of action as following:

   a. **Use knowledge in the world as well as knowledge in the head:** People work efficiently when knowledge required to complete the task is available externally.

Notes

But experts want to incorporate regular tasks to increase their efficiency. Hence, system should provide necessary knowledge within the environment and operations should be transparent.

b. **Simplify the structure of task:** Tasks must be simple in order to avoid complex problem solving and excessive memory load. There are number of ways to simplify the tasks.

- Provide mental support to users to keep track of stages in task.
- Use of technology to provide more information of task and feedback to user.
- Automate the task or part of it.
- Change the nature of the task so it will be simpler.

Above all of this, very important thing not to take control away from user.

c. **Make things visible:** The interface should clear the things like what system can do and how it is achieved. It also shows user the effect of their actions on the system.

d. **Get the mappings right:** Mapping between user intentions and system control and between user actions and system events must be clear. Hence, small movement will small effects and large movements will have large effects.

e. **Exploit the power of constraints:** Because of constraints it is impossible to do anything in the world but only correct action in correct way is possible.

For example, jigsaw puzzle. In this puzzle, pieces only fit together in one way.

Users are guided by physical constraints to complete the task.

f. **Design for error:** Most errors are caused by humans. Therefore, predict the mistakes that could be done by human and design recovery of the system.

g. **When all else fails, standardize:** If there are no natural mappings then arbitrary mapping should be standardized.

3. **Nielsen's ten heuristics:** Heuristics evaluation is a procedure to check user interface for usability problems. Once a usability problem is detected in design, they are solved as an integral part of constant design processes. Heuristic evaluation method is consist of some usability principles such as Nielsen's ten Usability principles which are as following:

Nielsen's Ten Heuristic Principles

- Visibility of system status.
- Match between system and real world.
- User control and freedom.
- Consistency and standards.
- Error prevention.
- Recognition rather than Recall.
- Flexibility and efficiency of use.
- Aesthetic and minimalist design.

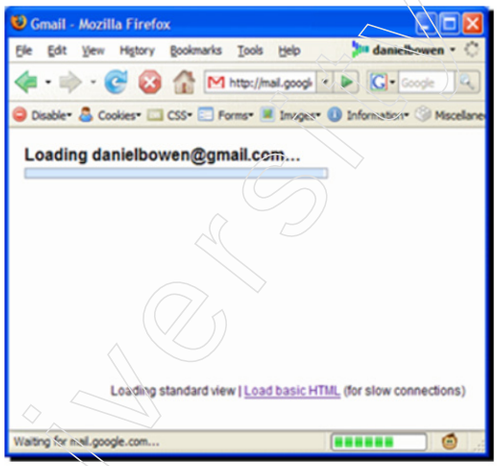
Notes

- Help, diagnosis and recovery from errors.
- Documentation and Help

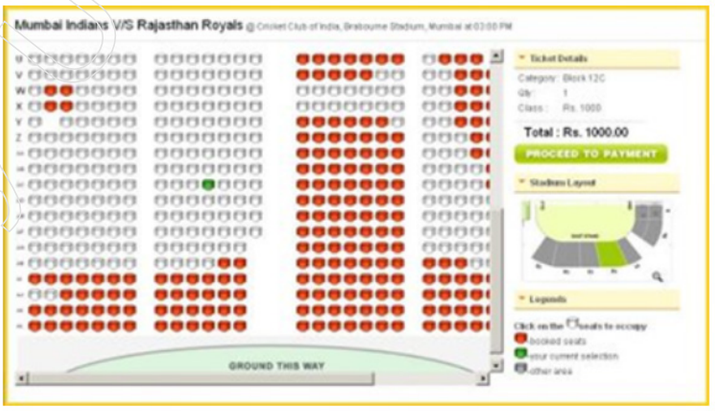
The above mentioned ten principles of Nielsen represents checklist which is used for evaluating and explaining problems for the heuristic evaluator. Let us discuss them one by one as following:

- **Visibility of System Status:** It is necessary that system always inform users about current state and actions using appropriate visual presentation like color change, loader, time-left graphics etc. it is also important to give feedback within reasonable time.

Consider the example of Gmail. While Gmail is loading the user's mailbox, it informs user to wait and indicates the status of what is going on.



Conider one more example of presentation of available and booked seats with price and other details while booking a movie ticket online.



- **Match between system and real world:** System should talk user's language, with words, phrases and concepts which are familiar to the users than system-oriented terms. If there is something in system which user cannot understand then it can create a problem over long period of time.

Using real world conventions, information can appear in natural and logical order.

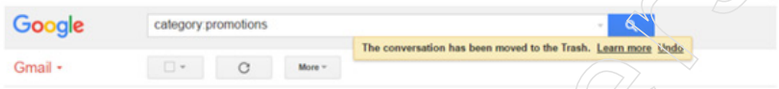
For trash following symbol is used for easy understanding of it.

Notes

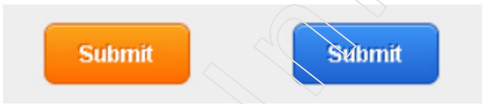
In online shopping for cart, designers used following symbol which is same as a cart in day to day life.

- **User control and freedom:** This is about freedom of users to navigate and perform actions. Sometimes users choose wrong function by mistakly, at that time freedom to undo the accidental action is necessary.

Again consider the example of Gmail. Gmail indicates flash message with undo action when you accidently delete the mail.



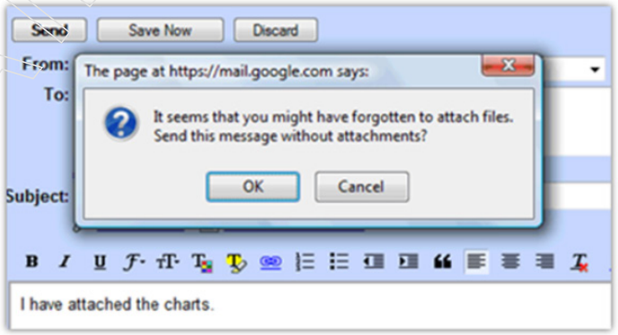
- **Consistency and standards:** Consistency is very important. If you consider the design of Submit button on one page then it should look same on any page across the website.



If you are representing the data in table format, then it should look same every time.

- **Error prevention:** A careful design to prevent the problem from occurring is better than good error message. To avoid this either you can eliminate error-prone conditions or check such conditions. After checking conditions, you can present them to users for confirmation option before they commit the action.

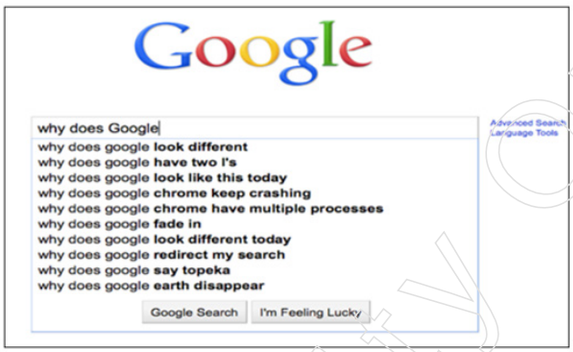
If you have mentioned something is attached in mail and there is no attachment then Gmail scans the email for such a keywords and alerts the user before sending the mail.



Notes

- **Recognition rather than Recall:** It is always better to recall than remember. To recall, you can provide set of options to user. The aim of this technique is to minimize the use of user's memory.

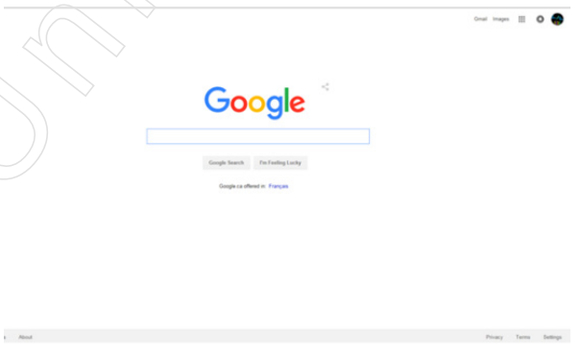
  Below is an example of the Google suggesting possible questions based on what you are trying to type.



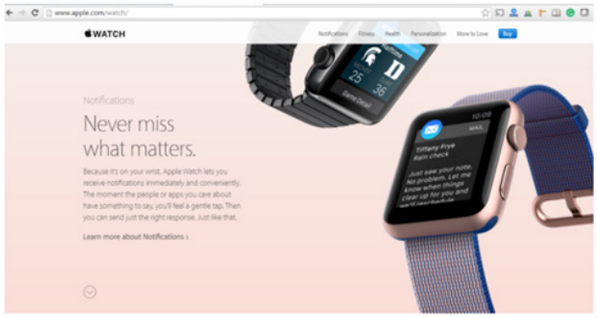
- **Flexibility and efficiency of use:** Interface should be flexible enough so that it can transform itself between a new user and an advance user.

  For example, while installing new software it gives you two options either select default installation or custom installation. In this case, advance user may select custom installation to reduce unnecessary steps.

- **Aesthetic and minimalist design:** Interface must have necessary and relevant information. For example, Google page shows very precise and only important information on their page. The following google page shows nothing more, nothing less.

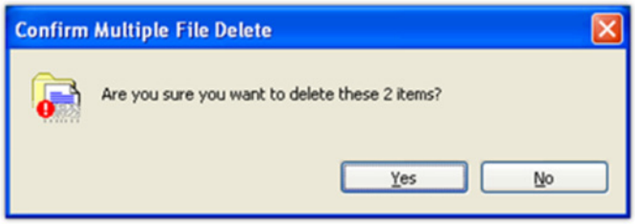


  Apple provides only basic information of features on their page. Extra information is hidden under "Learn More".



Notes

- **Help, diagnosis and recovery from errors:** Designer must represent error message in plain language (no codes) which will easily represent the problem and suggest the solution.



- **Documentation and Help:** Even if user can use the system without documentation, it is necessary to provide help and documentation. This kind of information must be easy to search, focused on user's task and list out the steps to be carried out. Last but not the least, it should not be too large.

2.3.5 HCI Patterns

Patterns are an approach of capturing and reusing the knowledge of extracting essential details if successful design so that it can be applied again and again on new situation.

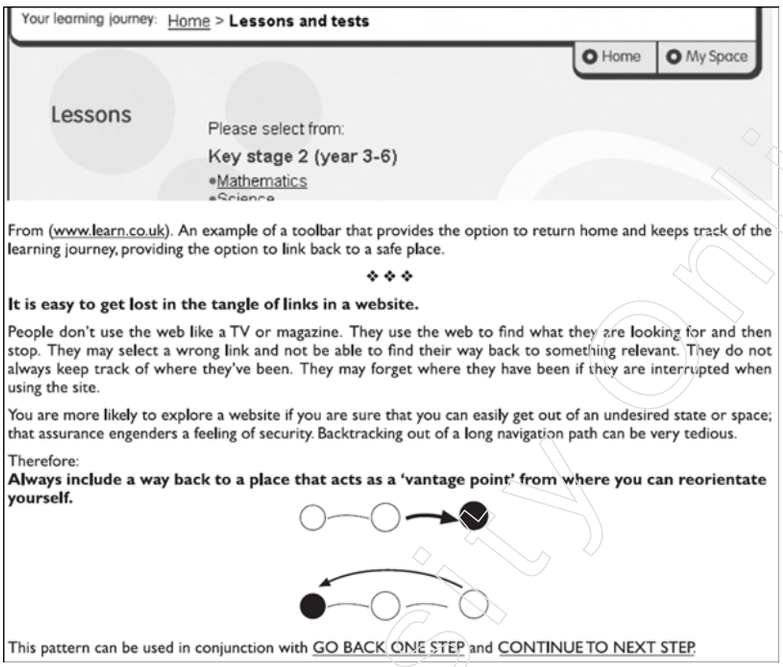
Patterns are used in software development to form solution to common programming issues. Recently, patterns are used in interface and web design. Pattern is a never changing solution to the problems which occurred again and again in specific context. Patterns solve the issues that designers face by providing a solution statement.

Pattern captures only never changing properties of good design. These never changing properties are common elements that hold between all instances of solution. On the basis of circumstances and designer's activity, specific implementation of the pattern is possible.

Patterns and pattern languages are characterized by number of features as following:

- They capture design practice and integrate knowledge of successful solutions. Design practice and knowledge come from practice than psychological theory.
- They capture the important common properties of good design. They do not inform the designer about how to do something but informs what needs to be done and why.
- They represent design knowledge at different levels, from social and organizational issues through conceptual design.
- The concept of a pattern language is generative. Hence, it can assist in the development of complete designs.
- They are generally intuitive and readable. Therefore, it can be used for communication between all stakeholders.

Notes



An example pattern "go back to safe place" adapted from Tidwell's common ground collection.

Summary:

- Design rules are the rules a designer can follow in order to increase the usability of software product.
- Principles are abstract design rules with high in generality and lower in authority.
- Standards are specific design rules with high authority and limited in application.
- Guidelines are lower in authority and high in generality with respect to applications.
- Shneiderman's Eight Golden Rules
  - Strive for Consistency.
  - Cater to Universal Usability.
  - Offer Informative feedback.
  - Design Dialogs to yield closure.
  - Prevent Errors.
  - Permit easy reversal of actions.
  - Support internal locus of control.
  - Reduce short term memory load.
- Norman's seven principles:
  - Use both knowledge in world & knowledge in the head.
  - Simplify task structures.
  - Make things visible.

Notes

- Get the mapping right (User mental model = Conceptual model = Designed model).
- Convert constrains into advantages (The constrains can be either Physical constraints or Cultural constraints or Technological constraints).
- Design for Error.
- When all else fails – Standardize

- Nielsen's Ten Heuristic Principles
  - Visibility of system status
  - Match between system and real world.
  - User control and freedom.
  - Consistency and standards.
  - Error prevention.
  - Recognition rather than Recall.
  - Flexibility and efficiency of use.
  - Aesthetic and minimalist design.
  - Help, diagnosis and recovery from errors.
  - Documentation and Help
- Patterns are an approach of capturing and reusing the knowledge of extracting essential details if successful design so that it can be applied again and again on new situation.

Activity

1. Select golden rule for design interface from following.
   a. Place the user in control
   b. Reduce the user's memory load
   c. Make the interface consistent
   d. All of the above
2. When users are involved in complex tasks, the demand on ……. can be significant.
   a. Short-term memory
   b. Shortcuts
   c. Objects that appear on the screen
   d. All of them
3. _____ and _____ are lower in authority.
   a. principles and standards
   b. standards and guidelines
   c. principles and guidelines
   d. none of them

Notes

4. _____can apply to either hardware or software used to build interactive system.
   a. principles
   b. standards
   c. guidelines
   d. all of them
5. _____is more crucial and costly to change than_____.
   a. hardware, software
   b. software, hardware
   c. interface, pattern
   d. pattern, interface
6. ISO stands for _____
   a. international organization for security
   b. international standards for organization
   c. international standardization for organization
   d. none of them
7. HyperTalk is the example of _____
   a. high level programming language
   b. low-level programming language
   c. both of them
   d. none of them
8. _____ is a simulation environment for the Macintosh line of Apple computers.
   a. Wizard of Oz
   b. HyperTalk
   c. HyperCard
   d. none of them
9. HyperCard is _____
   a. animation tool.
   b. prototyping tool
   c. both of them
   d. none of them
10. in _____ actual prototype is rejected.
    a. incremental
    b. throw-away

c. evolutionary

d. all of them

11 _____ is the target for the design.

a. now level

b. worst-case

c. planned level

d. best case

12. Now-level indicates the value for the _____

a. measurement with existing system

b. lowest acceptable measurement

c. best possible measurement

d. none of them

13. _____ is the designing the thing right.

a. validation

b. verification

c. both of them

d. none of them

Notes

Answer Keys (Exercise):

Question	Answer	Question	Answer	Question	Answer
1	d	2	a	3	c
4	b	5	a	6	d
7	a	8	c	9	c
10	b	11	c	12	a
13	b				

Notes

Module - III: Evaluation

Course Contents:

- Goals of Evaluation
- Evaluation through Experts
- Evaluation through Users
- Choosing an Evaluation method

Key Learning Objectives:

At the end of this block, you will be able to:

1. Evaluate intended goal of system
2. Understand evaluation through experts
3. Understand evaluation through users
4. Analyse various evaluation methods

Structure:

Unit - 3.1: Goals of Evaluation

Notes

Unit Outcomes:

At the end of this unit, you will be able to:

1. Understand basics to evaluate the goals.

3.1.1 Introduction

In previous unit, you studied to design the process which can support design of interactive system. After designing such a process, you have to evaluate your designs and system to make sure it is working as per your expectations. This evaluation must be done before any implementation work.

Evaluation cannot be carried out in single phase; it should be carried out throughout the design life cycle. The result of evaluation must work as feedback to modify the design. Evaluation is divided into three main goals:

- To assess the extent and accessibility of the system's functionality
- To assess users' experience of the interaction
- To identify any specific problems with the system

The system functionality is important and it must be as per user requirement. User must be able to perform their task more easily with that design of the system. According to system functionality, use of system should match to the user's expectations.

In addition to system's functionality, it is also important to assess users' experience of the interaction. It includes, how easily system can be learnt, its usability and satisfaction. The final goal is to identify specific problem with system design. This can be used when system creates confusion among users or if system gives unexpected output. Under this goal, point of failure has to be find to solve the problem.

Evaluation techniques are divided into two parts:

- Expert analysis
- User participation

In the next unit, we will discuss evaluation techniques.

Summary:

- Evaluation must be carried out throughout design life cycle.
- After designing the process, it is necessary to evaluate it.
- Evaluation is divided into three goals:
  - To assess the extent and accessibility of the system's functionality
  - To assess users' experience of the interaction
  - To identify any specific problems with the system

Notes

Unit - 3.2: Evaluation through Experts

Recall Session:

In the previous unit, you studied about:

1. The need of evaluation
2. Types of evaluation goals

Unit Outcome:

At the end of this unit, you will be able to:

1. Understand cognitive approach
2. Learn to evaluate using heuristic method
3. Learn to evaluate using model-based method

3.2.1 Introduction

Already we have discussed that evaluation must be carry out throughout the process. First evaluation must be performed before any implementation has started. If design can evaluate itself, then it can avoid expensive mistakes because design can be changed before any implementation or commitment.

If error is discovered after implementation, then to make the system work properly costs more. Hence, it is difficult to rectify the problem. However, it can be expensive to test the design at every stage and may you will not get the intended output with incomplete design.

There are various methods which can evaluate system using expert analysis. These methods depend upon designer or human expert. The basic purpose of these methods is to identify the area because of which problem has caused. These methods are cheap because they don't require user involvement.

Now you will learn approaches to expert analysis. There are four approaches to expert analysis as following:

- Cognitive walkthrough
- Heuristic evaluation
- The use of models
- Use of previous work

Let us discuss all the approaches one by one.

3.2.2 Cognitive Walkthrough

Cognitive walkthrough was proposed by Polson and colleagues as an attempt to introduce psychological theory into the informal and subjective walkthrough technique. The code walk through familiar in software engineering is the origin of cognitive walkthrough approach. Walkthrough needs detailed review of a series of actions. These

Notes

series of actions are nothing but the steps that interface will require a user to perform in order to perform some known task.

The main focus of cognitive walkthrough is to build how easy a system is to learn. Precisely, focus is on learning through exploration. Research says that many users prefer to learn the system by exploring its functionality hands on rather than training or manual.

For a cognitive walkthrough, you need following four things:

- A specification or prototype of the system; A specification doesn't have to be complete but it should be detail enough. It includes details such as location and wording for a menu which is very important.
- A description of the task which user has to perform on the system: This is representative task which most users' wants to do.
- A complete, written list of actions: These actions are needed in order to complete the task.
- An indication of who the users are: With the help of this, evaluators can assume what kind of experience and knowledge users have.

For each action, the evaluator tries to answer following questions:

- Is the effect of the action the same as the user's goal at that point?

Every users' action will result in effect on the system. With the help of this question evaluator will try to check is this the action user is trying to achieve. For example, if the effect of the action is to close a document, is 'closing a document' what the user wants to do?

- Will users see that the action is available?

This question not regarding whether user is able to see menu or button, for example to produce the action? This is about whether it is visible or not when they will need to use it. Answers to these kinds of questions are of type yes/no.

- Once users have found the correct action, will they know it is the one they need?

This complements previous question. Here experts will ask the question: will the user recognize that it is the one he is looking for to complete his task? Where the previous question was about the visibility of the action.

- After the action is taken, will users understand the feedback they get?

Imagine user managed to perform correct action, how he will understand so? Hence, it is necessary to provide feedback with enough information of what has actually happened. In order to understand if user have achieved the goal, user need proper feedback.

It is important to document cognitive walkthrough. Document can record what is good and what needs improvement in design. To document the record, you can generate standard evaluation form. Form includes:

1. Date and time of walkthrough
2. Names of evaluators

Notes

Maintain separate standard form for actions. Any negative answer to the action should be maintain on separate usability problem report sheet. This report sheet must include following points:

1. Version of the system
2. Date
3. Evaluators information
4. Description of the problem
5. Severity of the problem

By using this information designers sets priorities for correcting the designs because this is not always possible to fix all the problems.

3.2.3 Heuristic Evaluation

Heuristic is a guideline or general principal that can help to take design decision. That has already been made. Heuristic evaluation, developed by Jakob Nielsen and Rolf Molich, is a method for structuring the critique of a system using a set of relatively simple and general heuristics.

Heuristic evaluation can be performed on design specifications which will help in early design evaluation however it can be applied on fully functioning systems. It is cheap and flexible so considered a discount usability technique.

To help evaluators in discovering usability problems, a set of 10 heuristics are provided. The heuristics are related to principles and guidelines. These 10 heuristics effectively covers most common usability problems.

Each evaluator notes the violations of any of these heuristics by assessing the system. These violations indicate potential usability problem. The evaluator also assesses the severity of the problem using following four factors:

- How common is the problem?
- How easy is it for the user to overcome?
- Will it be a one-off problem or persistent one?
- How seriously will the problem be perceived?

Evaluator can provide severity rating on a scale of 0-4 as following:

0 = I don't agree that this is the usability problem.

1 = Cosmetic problem only: need not to fix unless extra time is available on project.

2 = Minor usability problem: fixing this should be given low priority.

3 = Major usability problem: important to fix, so should be given high priority.

4 = Usability catastrophe: imperative to fix this before product can be released.

Nielsen's ten heuristics are:

- Visibility of system status: Always inform users time to time what is going on through appropriate feedback.

Notes

Example: if system is taking time to update antivirus, it should give indication how long it will take and how much update is completed.

- Match between system and the real world: The system must speak user's natural language with word, phrases and concepts familiar to the user instead of system-oriented terms. It tends to appear information in natural and logical order.
- User control and freedom: This heuristic provides user a freedom to navigate and perform actions. Users accidently selects wrong functions or action. These wrong actions need to be correct which is done by this heuristic. This heuristic supports undo and redo.
- Consistency and standards: As per this, actions, menus, situations or words must be same with respect to their meaning or size or structure in different context. This can be achieved by following platform conventions and accepted standards.

Example: submit button on one page should look same across the site on any page.

- Error prevention: Make it difficult to generate the error. Careful design of the system is always better because it prevents a problem from occurring at first place.

Example: in Gmail, while composing the mail if you have mentioned word attachment and if you are sending that mail without attachment then system will show you warning message before sending it.

- Recognition rather than recall: System should suggest user set of options than to let him remember and type everything.
- Flexibility and efficiency of use: Interface should be flexible. It must adjust itself according to the user.

Example: while installing software, sometimes it shows two options: default installation and custom installation. Advance user will select custom installation and beginner will select default installation.

- Aesthetic and minimalist design: From designer point of view entire information displayed on page is important but it is necessary to ask user is entire information displayed on page is important. System should not contain irrelevant or rarely used information.

Example: in google search, it does not show entire information on their search page.

- Help users recognize, diagnose and recover from errors: Error message should be in simple language rather than in code format. Error message must indicate the problem and suggest the solution.
- Help and documentation: Some system doesn't provide instructions. In that case, it is necessary to provide help and documentation. This documentation must be easy to use, must focus on users' actions and clearly provides steps carried out by user and not too large.

After completing assessment by each evaluator, all the problems gathered together and mean severity ratings calculated. By using severity ratings design team decides which one is the most important and provides attention first.

Notes

3.2.4 Model Based Evaluation

This evaluation is based on use of models. Design specifications and evaluation can be combined into same framework with the help of cognitive and design models.

GOAMS (goals, operators, actions, methods and selection) model predicts user performance with particular interface and used to filter design options.

Key stroke model is a lower level modelling technique. It predicts the time taken by the user to perform low level physical activities.

Design rationale is a design methodology. It provides framework to evaluate design options.

Overall, you can say that models help in reducing the role of user in testing. It also reduces the role of expert evaluator.

Summary:

- There are four approaches to expert analysis:
  - Cognitive walkthrough
  - Heuristic evaluation
  - The use of models
  - Use of previous work
- For a cognitive walkthrough, you need four things:
  - A specification or prototype of the system.
  - A description of the task which user has to perform on the system.
  - A complete written list of actions.
  - An indication of who the users are.
- Cognitive walkthrough was proposed by Polson and colleagues.
- Heuristic evaluation, developed by Jakob Nielsen and Rolf Molich, is a method for structuring the critique of a system using a set of relatively simple and general heuristics.
- Heuristic evaluation can be performed on design specifications which will help in early design evaluation.
- To help evaluators in discovering usability problems, a set of 10 heuristics are provided. The heuristics are related to principles and guidelines.
- Design specifications and evaluation can be combined into same framework with the help of cognitive and design models.
- Models help in reducing the role of user in testing. It also reduces the role of expert evaluator.

Activity:

1. Illustrate real life example to apply heuristic evaluation. Using 10 heuristics evaluate the problem and provide severity rating.

Unit - 3.3: Evaluation through Users

Notes

Recall Session:

In the previous unit, you studied about:

1. Three types evaluation: cognitive walkthrough, heuristic evaluation and model-based evaluation

Unit Outcome:

At the end of this unit, you will be able to:

1. Understand styles of evaluation.
2. Learn empirical methods of evaluation.
3. Learn to evaluate using model-based method.
4. Analyse observational and query techniques as well as physiological monitoring.

3.3.1 Introduction

Till now we studied the techniques focus on evaluating design or system using designer or evaluator. But it is important to test using user of the system. In this unit you will learn different approaches to test the system using users.

These approaches include:

- empirical or experimental methods
- observational methods
- query techniques
- methods that use psychological monitoring

Users can participate in evaluation in the later stage of development.

3.3.2 Styles of Evaluation

There are two distinct evaluation styles:

- Those performed under laboratory conditions
- Those conducted in the work environment or in the field.

Laboratory Studies

In this type of studies, users are asked to come out of their working environment and take part in controlled tests. These controlled tests are carried out in specialist usability laboratory. Laboratory studies have various benefits and limitations.

The usability laboratory can be a well-equipped like it consists of audio/video recordings for guidance, two-way mirrors, computers which cannot be possible to install in work environment.

Notes

In usability laboratory, participants operate without any interruption which is not possible in work environment.

But there are some situations when there is only one option: laboratory studies. If the system has to be located in dangerous locations then laboratory studies is the good option.

Field Studies

In this type of studies, designers or evaluators are asked to come out into user's work environment to observe system in action.

Again, this studies also have advantages and disadvantages. Field study is difficult to observe because of high level of noise, movement and interruptions such as phone calls, surrounding people etc.

Field studies are open in nature. In this study you will observe interaction between individuals and between systems. You will observe users in his natural environment instead of controlled environment. Again, there are some activities which requires days or months to complete are impossible to study in laboratory environment.

However, field study is preferable than lab studies because it allows to study interaction which occurs in real time. But in field study also participants can be influenced by designer.

After studying both the types you can say that for certain situations laboratory testing is necessary and desirable. To evaluate specific interface feature, controlled experiments can be useful. These controlled experiments can be conducted under laboratory conditions only. From economic point of view, you have to consider the cost of installing equipment in the field. This cost might be affected on actual work situation i.e. it may cost one or two participants losing their job.

3.3.3 Empirical Methods

Previously, you have studied control experiment is the most powerful method for evaluating designs. It provides experimental proofs to support particular claim. Any experiment has same basic form but within this basic form there are number of factors that are important for experiment. These factors have to be considered carefully during experimental design. Let us discuss those factors.

- **Participants:** The choice of participants is important for the success of any experiment. Participants has to be chosen in such a way that it will match expected user population. It relates experimental testing with actual users which is not always possible. If participants are not actual users then they must have similar age group and intended level of education. It is also necessary to test their computer knowledge in general and with system in order to be similar. It is not reliable if we are using general public for testing an interface design because they are not intended user population.

  Another issue is sample size chosen. Sample size must be large enough because it represents population. However, the availability of participants is limited.

Notes

- **Variables:** Experiments change and manipulate the variables under controlled conditions. Variables are divided into two types:
  - Independent variables
  - Dependent variables

  Independent variables are those which can be changed where as dependent variables are those that can be measured.

  Independent variables are changed to produce different conditions for comparison. E.g. interface style, level of help, number of menu items and icons. These variables are given number of different values which is called as level of the variable. Complex experiments may have more than one independent variable.

  The value of dependent variable is depending on changes made in independent variables. E.g. time taken to complete the task, the number of errors made, quality of user performance.

- **Hypotheses:** Hypotheses is the prediction of outcome of experiment. It is framed with respect to dependent and independent variables. It will start with change in independent variable which will create difference in dependent variable. The aim of experiment is to prove that prediction is correct. This will be achieved by disproving null hypothesis. Null hypothesis states that there is no difference between dependent variables and levels of independent variables.

- **Experimental design:** In order to produce generalized results, it is necessary to carefully design the experiment. The first phase in experimental design is to choose the hypothesis. Here, you will decide exactly what you are trying to demonstrate. In the first phase you will also decide dependent and independent variables, what you are going to change and what change you are expecting. You will also think about participants: how many participants are available and are they representing users?

  In next phase you will decide experimental method. There are two types of methods: between subjects and within subjects. In between subjects, different conditions are assigned to different participants. There are two conditions: experimental and control. In experimental condition, variables have been changed. In control, variables don't change.

3.3.4 Observational Techniques

By observing users interacting with each other, you can gather information about actual use of system. Users are asked to complete predetermined task and then observation is carried out. In this section we will consider the techniques use to evaluate the system by user behaviour observation.

Think aloud and cooperative evaluation:

In this technique, user is asked to talk about what he is doing as he is observed. This is the simple technique and require little expertise. It provides useful information about the problem. It is also used to observe actual working of the system.

A change on think aloud is known as cooperative evaluation. In cooperative evaluation user is encouraged to consider himself as a collaborator in evaluation rather

Notes

than participant in experiment. Evaluator can ask users a question if his behaviour is not clear. If problem arises, user can ask evaluator for clarification.

Think aloud process has following advantages:

- The process is easy to learn and use by evaluator.
- The user is encouraged to criticize the system.
- Evaluator can clarify the points where confusion occurred.

Think aloud and cooperative evaluations are useful if recording method and further analysis if effective. There are number of methods to record evaluation session. Methods for recording user actions include following:

Paper and pencil: it is cheap but hard to provide detail information because it will be limited by writing speed. Variation to this is use of a notebook computer for entry but this is also limited to typing speed. If this is the only method for recording then separate note taker than evaluator is recommended.

Audio recording: it is useful in thinking aloud process. But it may be difficult to record sufficient information to identify the actions and it will be also difficult to match audio recording with some other form of protocol.

Video recording: the main benefit is you can see what participant is doing. But sometimes it is difficult to keep participants in view though you have selected suitable camera and viewing angle.

User notebooks: using notebooks, participants will keep logs of their activities.

Analysing methods for recording user actions is time consuming hence, it is the need of automatic analysis tool. EVA (Experimental Video Annotator) is a system that runs on a multimedia workstation with a direct link to a video recorder. The evaluator can devise a set of buttons indicating different events. These may include timestamps and snapshots, as well as notes of expected events and errors. The buttons are used within a recording session by the evaluator to annotate the video with notes. During the session the user works at a workstation and is recorded, using video and audio and system logging as well. On the screen is the live video record and a view of the user's screen. The evaluator can use the buttons to tag interesting events as they occur and can record additional notes using a text editor.

3.3.5 Query Techniques

This is useful for detailing user's view about the system. It is based on philosophy which states that "ask the user" is the best way to find out how system meets the user requirements. Here you can get direct user's viewpoint and users reveal issues which are not considered by designer. Query techniques are simple and cheap. Query techniques are divided into two types:

- Interviews
- Questionnaires

Users can share their experience with an interactive system in interview. This is the structured way of gathering direct information. The questions can be varying in interview and evaluator can ask more questions to user for detail information. Interview

Notes

follows top down approach. Top down approach means starting with general questions and progressing to more leading questions which can explain user responses. These questions can be explored later with respect to interaction. Interview may reveal the issues which are not observed by designer.

In order to make interview effective, it must be planned in advance with ready set of questions. It helps to focus on purpose of the interview. Interview is not a controlled experimental technique.

Questionnaires technique is less flexible than interview technique. Because exploration of questions is not allowed and questions are fixed in advance. It can be used to reach to the larger group of participants and it takes less time to analyse. Questions can be asked at various points to collect the information of user's needs, preferences and experience. It is important to design the questionnaire very well mannered. First important thing is to establish the purpose of questionnaire. It is also important to decide how to analyse the questionnaire responses.

Questions can be asked in various styles as following:

- General: these kinds of questions provide background of the user

  Example: age, sex, occupation, place of residence, experience with computer, etc.

- Open ended: the response of these kind of questions is the opinion of the user.

  Example: "can you suggest improvement in design interface?"

Open ended questions used to collect general information of particular topic. But they are difficult to analyse and compare. These questions can provide suggestions and point out the error which are not observed by designer.

- Scalar: scalar questions ask user to examine the statement using numeric scale. Usually, it is with respect to agreement or dis-agreement.

  e.g. It is easy to recover from damage.

  Disagree 1 2 3 4 5 Agree

A coarse scale (scale from 1 to 3) provides clear meaning of the numbers. 1 for disagree, 2 for neutral and 3 for agree. But consider a very fine scale (scale from 1 to 10). These numbers are more difficult to interpret. In very fine scale different user will interpret different meaning of each number. Hence, a middle ground scale is preferred. Scale of 1 to 5 or 1 to 7 have been used effectively.

- Multi-choice: here respondent is provided with predefined responses. From these responses' user can select either one or multiple responses.

  For example:

  How do you must often get help with the system (tick one)

  Onlien manual ☐

  Contextual help system ☐

  Command prompt ☐

  Asix a colleague ☐

Notes

Which type of software have you used (tick all that apply)?

Word processor ☐

Database ☐

Spreadsheet ☐

Expert system ☐

Online help system ☐

Complier ☐

- Ranked: in this method, respondent provides rank to the response as per their preferences.

For example:

Please rank the usefulness of these methods of issuing a command (I most useful, 2 next, 0 if not used).

Manu selection ☐

Command line ☐

Control key accelerator ☐

It is recommended to use scalar, ranked and multi-choice method as possible to reduce the efforts of respondent. These methods provide user with alternative responses which reduces their efforts. In addition to this they are easy to analyse.

3.3.6 Evaluation through Monitoring Physiological Responses

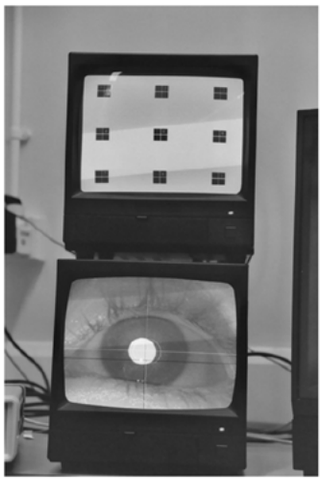
Most evaluations are depending on observations and responses from user about how and what they are feeling. To overcome this dependency, system should measure the things directly. This can be achieved using objective usability testing. Objective usability testing is the way of monitoring physiological aspects of computer use. It gives us clear inside out information of what user do when they interact with computers and how they feel. This monitoring can be done in two ways:

- Eye tracking
- Physiological measurement

In eye tracking, camera and light resources are mounted in desk unit to monitor the eye (refer following image).

Eye tracking Equipment

Notes



Calibrating the eye tracker

Furthermore, software used for eye-tracking control equipment has been improved rapidly. This software also used for the analysis and visualization of the data produced by eye-tracking control equipment. Eye tracking measure where people look and pattern of their eye movement. It helps to find out which area of the screen is easy and difficult to process by users. There are many possible measurements:

Number of fixations: the more fixations the less efficient the search strategy.

Fixation Duration: Longer fixations may indicate difficulty with display.

Scan path: indicates area of interest, search strategy.

Emotional responses are closely related to physiological changes. It includes changes in heart rate, breathing and skin secretions. Physiological measurements involve attaching various probes and sensors to the user. These measures number of factors:

Heart activity indicated by blood pressure, volume and pulses. These may respond to stress and anger.

Activity of sweat glands, indicated by skin resistance. It indicates level of arousal and mental efforts.

Electrical activity in muscles measured by electromyogram. It reflects involvement in a task.

Electrical activity in the brain measured by electroencephalogram. It is associated with decision making, attention and motivation.

Summary:

- There are two distinct evaluation styles:
  - Those performed under laboratory conditions
  - Those conducted in the work environment or in the field.
- In laboratory studies, users are asked to come out of their working environment and take part in controlled tests.

Notes

- In laboratory studies, users are asked to come out of their working environment and take part in controlled tests.
- In field studies, designers or evaluators are asked to come out into user's work environment to observe system in action.
- Participants has to be chosen in such a way that it will match expected user population.
- Independent variables are those which can be changed whereas dependent variables are those that can be measured.
- Independent variables are changed to produce different conditions for comparison.
- Hypotheses is the prediction of outcome of experiment.
- Interview is not a controlled experimental technique.
- Think aloud process has following advantages:
  - The process is easy to learn and use by evaluator.
  - The user is encouraged to criticize the system.
  - Evaluator can clarify the points where confusion occurred.
- Questions can be asked in following styles:
  - General
  - Open ended
  - Scalar
  - Multi-choice
  - Ranked
- It is recommended to use scalar, ranked and multi-choice method as possible to reduce the efforts of respondent.
- Physiological monitoring can be done in two ways:
  - Eye tracking
  - Physiological measurement

Activity:

1. Illustrate real life examples for open ended, scalar, ranked and multi choice questions.
2. Compare lab studies with field studies.

Unit - 3.4: Choosing an Evaluation Method

Notes

Recall Session:

In the previous unit, you studied about:

1. Styles of evaluation.
2. Empirical method of evaluation.
3. Evaluation using model-based method.
4. Observational and query techniques and physiological monitoring.

Unit Outcome:

At the end of this unit, you will be able to:

1. Understand factors distinguishing evaluation techniques.
2. Classify evaluation techniques.

3.4.1 Introduction

Previously you have studied that various techniques are available to evaluate interactive systems at every stage. So, it is difficult to decide most appropriate method as per the need.

To decide most appropriate method, there is no such a rule. Every method has its own pros and cons. Hence, each method is useful if it is applied properly.

However, you can consider some factors while selecting evaluation techniques. Using factors, you can categorize various methods which will help to compare and choose between them.

3.4.2 Factors Distinguishing Evaluation Techniques

By using eight factors, you can distinguish different evaluation techniques. This will help to make appropriate choice. Those 8 factors are:

- The stage in the cycle at which the evaluation is carried out
- The style of evaluation
- The level of subjectivity or objectivity of the technique
- The type of measures provided
- The information provided
- The immediacy of the response
- The level of interference implied
- The resources required.

Notes

Design vs. Implementation

The first factor can affect where evaluation is required. Earlier, you have learnt evaluation is necessary throughout the design lifecycle. However, there is difference between evaluation of design and evaluation of implementation.

Design evaluation need to be quick and cheap hence it involves only design experts. Whereas, evaluation of implementation needs to be in detail therefore it involves users as participants.

Laboratory vs Field Study

In previous unit, already you have studied strength and weaknesses of these two styles of evaluation. Laboratory studies provides controlled environment for experiments and observations. Therefore, it loses natural observations of user's environment. Field study can be used with implementations but it does not provide control over user activity. Ideally, it is suggested to include both the styles of evaluation.

Subjective vs. Objective

Subjective methods can be powerful if used properly and provide information which cannot be given by objective methods. In subjective method, evaluator can act bias. Therefor, it is necessary to recognize and avoid this problem. It can overcome by using more than one evaluator.

On the other hand, objective method generates repeatable results which are not depend on result of evaluation by evaluator. E.g. controlled experiments. Ideally, both subjective and objective approaches should be used.

Qualitative vs Quantitative Measures

Evaluation techniques provide two types of measurement: qualitative and quantitative.

Subjective techniques provide qualitative measurement while objective techniques provide quantitative measurement.

Information Provided

The information required to evaluator at any stage of evaluation vary from low level to higher level information. Controlled experiments are good at providing low level information. Questionnaire and interview provide high level information.

Immediacy of Response

Some methods like think aloud record the user's behaviour at the time of interaction itself. However, some methods are depending on user's recollection of events. Such a recollection can result into bias recall which might be incomplete.

Intrusiveness

Certain techniques which produces immediate results are at high risk of influencing the way user behaves.

Notes

Resources

It is important to consider availability of resources while selecting evaluation techniques. Resources include equipment, time, money, participants, expert evaluators etc. Some decisions are forced by resource limitations. For example: it is not possible to generate video protocol without access to video camera. Sometimes time and money are limited. In such a situation evaluator must decide which evaluation method will produce more effective and more useful information.

3.4.3 A Classification of Evaluation Techniques

Using previously studied 8 factors you can classify evaluation techniques. Following table shows classification analytic techniques.

	Cognitive walkthrough	Heuristic evaluation	Review based	Model based
Stage	Throughout	Throughout	Design	Design
Style	Laboratory	Laboratory	Laboratory	Laboratory
Objective	No	No	As source	No
Measure	Qualitative	Qualitative	As source	Qualitative
Information	Low Level	High Level	As source	Low Level
Immediacy	N/A	N/A	As source	N/A
Intrusive	No	No	No	No
Time	Medium	Low	Low-Medium	Medium
Equipment	Low	Low	Low	Low
Expertise	High	Medium	Low	High

Classification of analytic evaluation technique

Following table represents classification of experimental and evaluation techniques.

	Experiment	Interviews	Questionnaire
Stage	Throughout	Throughout	Design
Style	Laboratory	Lab/Field	Lab/Field
Objective	Yes	No	No
Measure	Quantitative	Qualitative/ Quantitative	Qualitative/ Quantitative
Information	Low/High Level	High Level	High Level
Immediacy	Yes	No	No
Intrusive	Yes	No	No
Time	High	Low	Low
Equipment	Medium	Low	Low
Expertise	Medium	Low	Low

Notes

Classification of experimental & query evaluation techniques

Following table shows classification of observational techniques.

	Think aloud	Protocol analysis	Post-task walkthrough
Stage	Implementation	Implementation	Implementation
Style	Lab/Field	Lab/Field	Lab/Field
Objective	No	No	No
Measure	Qualitative	Qualitative	Qualitative
Information	High/Low Level	High/Low Level	High/Low Level
Immediacy	Yes	Yes	No
Intrusive	Yes	Yes	No
Time	High	High	Medium
Equipment	Low	High	Low
Expertise	Medium	High	Medium

Classification of observational techniques

Classification of monitoring evaluation techniques is as shown in following table.

	Eye tracking	Physiological measurement
Stage	Implementation	Implementation
Style	Lab	Lab
Objective	Yes	Yes
Measure	Quantitative	Quantitative
Information	Low Level	Low Level
Immediacy	Yes	Yes
Intrusive	No	Yes
Time	Medium/High	Medium/High
Equipment	High	High
Expertise	High	High

Classification of monitoring evaluation technique

Summary:

- Every method has its own pros and cons. Hence, each method is useful if it is applied properly.
- By using eight factors, you can distinguish different evaluation techniques. This will help to make appropriate choice.
- Design evaluation need to be quick and cheap hence it involves only design experts. Whereas, evaluation of implementation needs to be in detail therefore it involves users as participants.

Notes

- Laboratory studies provides controlled environment for experiments and observations. Therefore, it loses natural observations of user's environment.
- Field study can be used with implementations but it does not provide control over user activity.
- Subjective methods can be powerful is used properly and provide information which cannot be given by objective methods.
- Objective method generates repeatable results which are not depend on result of evaluation by evaluator.
- Subjective techniques provide qualitative measurement while objective techniques provide quantitative measurement.
- Controlled experiments are good at providing low level information.
- Questionnaire and interview provide high level information.
- It is important to consider availability of resources while selecting evaluation techniques.

Activity:

1. What are the factors governing the choice of an appropriate evaluation method for different interactive systems? Give brief details.

Exercise

1. Evaluation must be carried out throughout the design life cycle.
   a. True
   b. False
2. Evaluation is divided into _____ main goals.
   a. One
   b. Two
   c. Three
   d. Four
3. Evaluation techniques are of _____ Types.
   a. One
   b. Two
   c. Three
   d. Four
4. Expert analysis is consist of _____
   a. Cognitive walkthrough
   b. Heuristic Evaluation
   c. The use of models

Notes

d. Use of previous work.

a. Both a & b

b. Both b & c

c. Both c & c

d. None of them

5. Cognitive walkthrough was proposed by _____

Users prefer to learn the system by exploring its functionality by training or manual rather than hands on.

a. True

b. False

6. Heuristic evaluation was developed by _____

Which of the following is not a heuristic?

a. User control and freedom

b. Error prevention

c. None of them

d. Both of them

7. Evaluator can provide severity rating on a scale of _____

a. 0-3

b. 0-4

c. 1-3

d. 1-4

8. If the system has to be located in dangerous locations then _____ studies is the good option.

a. Laboratory

b. Field

9. _____ study is preferable than _____ studies because it allows to study interaction which occurs in real time.

a. Field, lab

b. Lab, field

c. Both of them

d. None of them

10. Experiments change and manipulate the _____ under controlled conditions.

a. Variables

b. Participants

c. Hypotheses

Notes

d. design

11. _____ variables are those which can be changed whereas _____ variables are those that can be measured.

a. Independent, dependent

b. Dependent, independent

12 _____ is the prediction of outcome of an experiment.

a. Variables

b. Participants

c. Hypotheses

d. Design

13. In experimental condition, variables have been changed.

a. True

b. False

14. Interview follows _____ approach.

a. Top-down

b. Bottom-up

c. None of them

d. Both of them

15. _____ technique is less flexible than _____ technique.

a. Questionnaires, interview

b. Interview, questionnaires

16. _____ type of questions provides predefined responses to respondent.

a. Scalar

b. Ranked

c. Multi-choice

d. Open ended

17. Blood pressure, volume and pulses indicates _____

a. Electrical activity in muscles

b. Electrical activity in brain

c. Activity of sweat glans

d. None of them

18. _____ activity is measured by electromyogram.

a. Electrical activity in muscles

b. Electrical activity in brain

Notes

c. Activity of sweat glans

d. None of them

Answer Keys (Exercise):

Question	Answer	Question	Answer	Question	Answer
1	True	2	Three	3	Two
4	None of them	5	Polson and colleagues	6	False
7	Jakob Nielsen and Rolf Molich	8	Both of them	9	0-4
10	Laboratory	11	Field, lab	12	Variables
13	Indepen-dent, De-pendent	14	Hypoth-eses	15	True
16	Top-down	17	Question-naires, interview	18	Multi-choice
19	None of them	20	Electrical activity in muscles		

Module - IV: Models and Theories

Notes

Course Contents:

- Cognitive models
- Communication and collaboration models
- Task Decomposition
- Dialog notations and semantics
- Models of the system

Key Learning Objectives:

At the end of this module, you will be able to:

1. Evaluate various models: cognitive, communication and collaborative, interaction models.
2. Understand task decomposition and standard formalism.
3. Analyse dialog notations and semantics.

Structure:

Unit 4.1: Cognitive Model

4.1.1 Introduction

4.1.2 Challenges of display-based system

Unit 4.2: Communication and collaboration models

4.2.1 Introduction

Unit 4.3: Task decomposition

4.3.1 Introduction

4.3.2 Knowledge based Analysis

Unit 4.4: Dialog notations and semantics

4.4.1 Introduction

4.4.2 Dialog semantics

Unit 4.5: Models of the system

4.5.1 Introduction

4.5.2 Standard Formalism

4.5.3 Interaction Models

Unit 4.6: Hypertext, multimedia and World Wide Web

4.6.1 Introduction

Notes

Unit - 4.1: Cognitive Model

Unit Outcome:

At the end of this unit, you will be able to:

1. Understand cognitive model.
2. Analyse challenges of display-based system.

4.1.1 Introduction

Cognitive model is a computer program which work like a human. It may perform cognitive, perceptual and motor processes similar to humans to complete the task. Cognitive model may take same time that human take to perform and learn the task, it may generate the same errors like human do. Using cognitive model, you can have a computer which behaves like human.

In HCI, cognitive model used to implement better interfaces for users.

Cognitive models represent user of interactive system. Cognitive models can be:

1. **Hierarchical models:** Hierarchical models represent a task of user and structure of goal. It uses mental processing as divide and conquer. User achieves goal by solving sub goal. GOMS, CCT are the models with this feature. Consider the example of sales report. You have to generate the sales report of some textbooks. You can achieve this goal by dividing goal into sub goals like gathering the data, producing tables and histograms and writing descriptive material. Gathering the data can be divided into several sub goals like finding the names. In this way, you can keep dividing sub goals into further sub goals and once you found certain level of details then you can stop.

```
produce report
    gather data
    .   find book names
    .   .   do keywords search of names database
                <<further subgoals>>
    .   .   sift through names and abstracts by hand
                <<further subgoals>>
    .   search sales database
            <<further subgoals>>
    layout tables and histograms
        <<further subgoals>>
    write description
        <<further subgoals>>
```

Above image shows structure of hierarchical model.

GOMS: The GOMS model introduced by Card, Moran and Newell is an acronym for Goals, Operators, Methods and Selection.

It has four elements as following:

- **Goals**: It describes user's goals which user wants achieve. It represents memory point for the user. Using memory point, user can evaluate what system should do.

Notes

- Operators: These are low level analysis a and basic action. To use the system, user has to perform these actions.
- **Methods**: There are various methods to divide goals into sub goals. For example in a window manager a currently selected window can be closed either by selecting the 'CLOSE' option from a pop-up menu, or by hitting the 'L7' function key. In GOMS these two goal decompositions are called as methods. Those methods are CLOSE-METHOD and the L7-METHOD:

```
GOAL: ICONIZE-WINDOW
.     [select GOAL: USE-CLOSE-METHOD
.             .     MOVE-MOUSE-TO-WINDOW-HEADER
.             .     POP-UP-MENU
.             .     CLICK-OVER-CLOSE-OPTION
              GOAL: USE-L7-METHOD
.             .     PRESS-L7-KEY]
```

- **Selection**: Selection helps to choose methods from various methods. Selection of method is depends on user, state of the system and details of the goals. For example, a user, Sam, never uses the L7-METHOD, except for one game, 'blocks', where the mouse needs to be used in the game until the very moment the key is pressed.

  GOMS captures this in a selection rule for Sam:

  User Sam:

  Rule 1: Use the CLOSE-METHOD unless another rule applies.

  Rule 2: If the application is 'blocks' use the L7-METHOD.

CCT [Cognitive Complexity Theory]

Kieras and Polson invented a cognitive complexity theory. It depends on goal decomposition from GOMS and provides a model with more predictive power. It has two parallel description:

- User's goal
- Computer system

The goals are expressed using production rules. The sequence of rules is nothing production rules. The syntax is as following:

If condition then action

In above syntax, condition is a statement about the contents of working memory. If condition is true then production rule is ready to perform its action. Action can be either single action or group of actions. LISP-like language is used to write production rule program.

2. **Linguistic models:** It represents grammar of user-system.

   Example: BNF, TAG

   BNF: Linguistic approach can be represented using Backus-Naur Form (BNF) rules.

Notes

It also represents dialog grammar. It considers only syntactic level of dialog while ignoring the semantics of the language. It used to specify the syntax of computer programming language. It also describes system dialogs very easily.

For example, imagine a graphics system that has a line-drawing function. To select the function the user must select the 'line' menu option. The line-drawing function allows the user to draw a polyline that is a sequence of line arcs between points. The user selects the points by clicking the mouse button in the drawing area. The user double clicks to indicate the last point of the polyline.

draw-line: = select-line + choose-points

+ last-point

select-line: = position-mouse + CLICK-MOUSE

choose-points: = choose-one

| choose-one + choose-points

choose-one: = position-mouse + CLICK-MOUSE

last-point: = position-mouse + DOUBLE-CLICK-MOUSE

position-mouse: = empty | MOVE-MOUSE + position-mouse

The names in the description are of two types: non-terminals, shown in lower case, and terminals, shown in upper case. Terminals represent the lowest level of user behavior, such as pressing a key, clicking a mouse button or moving the mouse. Non-terminals are higher-level abstractions. The non-terminals are defined in terms of other non-terminals and terminals by a definition of the form

name: = expression

The ': =' symbol is read as 'is defined as'. Only non-terminals may appear on the left of a definition. The right-hand side is built up using two operators '+' (sequence) and '|' (choice).

TAG: TAG stands for Task Action Grammar. BNF rules ignore the advantages of consistency both in the language's structure and in its use of command names and letters. To overcome this TAG includes elements such as parametrized grammar rules to emphasize consistency and encoding the user's world knowledge

3. **Physical and device models;** It represents human motor skills.

KLM: The base of KLM is human motor system which is used for detail prediction about user performance. It concentrates on unit tasks within interaction. For example search and replace feature or changing the font of a word. It does not considers complex actions such as generating diagram. It considers that complex task can be divided into subtasks before mapping them into physical actions.

The task is divided into two phases:

- Acquisition of the task: user builds mental preparation of the task
- Execution of the task: user executes the task using system's facilities

KLM gives prediction for the activity of latter stage. During acquisition phase, user decides how to finish the task using system primitives. Hence, during execution activity

Notes

user is an expert. KLM can be consider as very low level GOMS model with given method. The execution phase is divided into five different physical motor operators, a mental operator and a system response operator as following:

K Keystroking, actually striking keys, including shifts and other modifier keys.

B Pressing a mouse button.

P Pointing, moving the mouse (or similar device) at a target.

H Homing, switching the hand between mouse and keyboard.

D Drawing lines using the mouse.

M Mentally preparing for a physical action.

R System response which may be ignored if the user does not have to wait for it, as in copy typing

4.1.2 Challenges of the Display-based System

Initially, goal hierarchical and grammar-based techniques were developed. At that time interactive systems were command line and keyboard, cursor based.

But the thing is how well these techniques will deal with modern interfaces. Both the techniques ignore output of the system which describes what user can see. It is assumed that users know what they want to do and therefore, they can execute appropriate command blindly. To overcome this TAG has been extended so that it can explain how display can affect the grammar rule.

Another problem is low level of lexical structure. Pressing a cursor key is a compatible for lexeme but moving a mouse one pixel is not appropriate. In addition, dialogs based on pointer are more display oriented. Clicking a cursor at particular point on the screen is depend on contents of current screen.

Goal hierarchy methods have different issues. The reason behind issues is, this method considers that display oriented system provides less structured methods to achieve the goal. Instead of working with well designed plan, users are exploring the task, finding direction and backing out of other. If level has chosen properly, goal hierarchies can partially work with display-oriented system.

Summary:

- Cognitive model is a computer program which work like a human.
- In HCI, cognitive model used to implement better interfaces for users.
- Cognitive models can be:
  - Hierarchical models: e.g. GOMS, CCT
  - Linguistic models: e.g. BNF, TAG
  - Physical and device models: e.g. KLM

Notes

Unit - 4.2: Communication and Collaboration Model

Recall Session:

In the previous unit, you studied about:

1. Cognitive Models.
2. Challenges in display-based system,

Unit Outcome:

At the end of this unit, you will be able to:

1. Understand communication and collaboration model.

4.2.1 Introduction

Groupware systems are consisting of more than one person. E.g. emails, conferencing systems etc. Groups can change with respect to composition and behaviour. Effective group working is depending on working environment. Therefore, study of behaviour of group and evaluation of groupware is very difficult. Effective communication is the base of group work and many systems focus to support communication at a distance.

There are various models for communication and collaboration as following:

- **Face to face communication:** It includes speech, hearing, body language and eye-gaze. To learn new norms, person should know existing norms. The important factor in this communication is personal space. Personal space varies according to the context, environment, diversity and culture. These factors also matter in case of video conference between two individuals from different background.

  During video conference, factor of eye-gaze is important because cameras are mounted away from the monitor and eye contact is important during conversation.

  To claim the turn in conversation, participants can use some interruptions like ‘um’ and ‘ah’.

- **Conversation:** Heavily annotated conversation structure can be used by transcripts but it lacks the back-channel information. There is one more structure called turn taking which can be interpreted as Adjacency pair. E.g. A-x, B-x, A-y, B-y. The context can change according to the conversation. The focus of context can also change. Hence, it is difficult to keep the track of context using adjacency pairs.

- **Text-based communication:** Text based communication is used for asynchronous and sometimes for synchronous groupware. Text-based communication is divided into four types:

  a. Discrete: Current message can depend on previous message.

     Example: email

  b. Linear: Single transcript is the good example of linear communication.

Notes

c. Non-linear: In this type, messages are linked to other messages in the format of hypertext.

d. Spatial: In this type, messages are arranged in two-dimensional surface.

Text based communication differ from face to face communication because text based don't have back channels and states. Text based communication used turn-taking as a fundamental structure.

Text based communication lacks the back channel. It can badly affect the conversation on two parties. The speaker would pause to seek back channel confirmation, the text speaker must either continue regardless or finish the message, effectively passing the turn. In text based communication in addition to back channel, the speaker's tone of voice and body language are also absent. Body language and tone of voice provides affective state of the speaker like happy, sad and angry. Email users have developed explicit tokens of affective states using smiles and acronyms.

For example:

:-) – smiling face, happy

:-(– sad face, upset or angry

;-) – winking face, humorous

LOL – laughing out loud.

In a text-based system, different participants can compose simultaneously, but they lack cotemporality. As we saw, even if the messages appear as they are produced, they will not be read in real time. In addition, the messages may only be delivered when complete and even then may be delayed by slow communication networks.

Turn taking is a fundamental structure of text based communication. It is simple for pair of participants. First one person says something, then the other. The only problem is deciding exactly when the exchange should happen. With three or more participants, turn-taking is more complex. They must decide who should have the next turn.

In an ordinary speech context is very important. Utterances are highly ambiguous and are only meaningful with respect to external context, the state of the world, and internal context, the state of the conversation. Both of these are problems in text-based communication. Whatever the means of direct communication, remote participants have difficulty in using deictic reference. They cannot simply say 'that one', but must usually describe the referant: 'the big circle in the corner'.

- **Group working:** The roles and relationship between individuals of group are different and it can change during the conversation. To maintain the factors in group working, physical layout is very important.

Summary:

- Face to face communication is very difficult.
- Body movement, facial expression, eye-gaze and eye-contact can be used for back channel.
- Text-based communication has very low feedback.

Notes

Exercise:

Go into an office or other place where several people are working together. Try to note down in as much detail as possible what they are doing and when. Do this with different foci: focus on the direct interpersonal communications, focus on the shared objects such as a calendar or document, or focus on one worker at a time. Whilst collecting data and when ordering your notes, look for breakdowns and misunderstandings, and for implicit communication through objects.

Look also at a particular task over a period of time, and note the number of interruptions as a worker performs the task, or the way a single task is contributed to by several workers.

Unit - 4.3: Task Decomposition

Notes

Recall Session:

In the previous unit, you studied about:

1. Various communication and collaboration model.

Unit Outcome:

At the end of this unit, you will be able to:

1. Understand task decomposition and knowledge-based analysis.

4.3.1 Introduction

The process of analysing working style of people is called task analysis.

For example, in the job of housekeeping we can think of following things:

"In order to clean the house: get the vacuum cleaner out, fix the correct attachments, clean the rooms, when the dust bag become full empty it."

To perform the task mentioned above, you must have information about vacuum cleaners, their attachments, dust bags, cupboard, rooms etc.

Task analysis have three approaches as following:

- **Task decomposition:** It is the process of diving task into subtask.
- **Knowledge based analysis:** It determines the knowledge of user regarding objects and action involved in the task.
- **Entity-relation-based analysis:** It is an object-based approach. It focuses on identifying objects and actors, relation between them and actions performed by them.

In the example of vacuum cleaner mentioned above, you can observe how the task "clean the house" is decomposed into several subtasks. Hierarchical task analysis (HTA) is the major approach towards task decomposition. The output of HTA includes hierarchy of task and subtask. It also generates the plan which describes order and conditions to perform subtask.

0	in order to clean the house 1. get the vacum cleaner out 2. fix the appropriate attachement 3. clean the rooms 3.1 clean the hall 3.2 clean the living rooms 3.3 clean the bedrooms 4. empty the dust bag 5. put the vacum cleaner and attachments away
Plan 0:	do 1-2-3-5 in that order. when the dust bag gets full do 4
Plan 3:	do any of 3.1. 3.2 or 3.3 in any order depending on which rooms need cleaning

How to clean the house

Notes

The above figure represent house cleaning example. In that figure, you can observe decomposition of task “clean the rooms”. Indentation denotes levels in task hierarchy and to emphasize this hierarchy numbers have provided. The plans are labelled by the task to which they are associated.

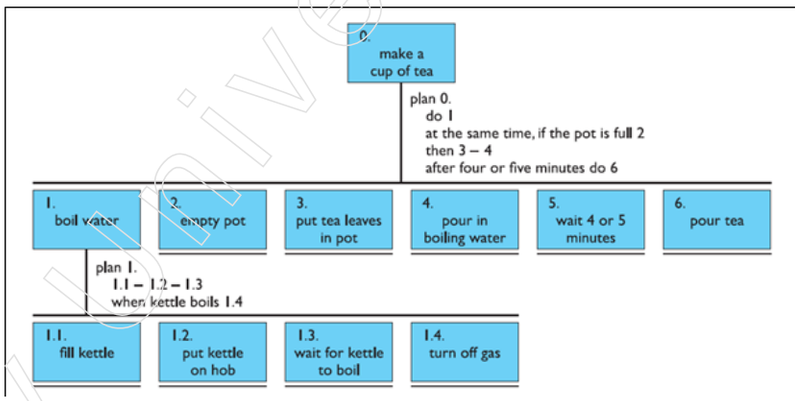
By observing the plans, you can say that it is not necessary to perform all the subtasks and it is also not necessary to follow their order of presentation. The process of generating hierarchy is iterative. Assume you want to perform some task, such as house cleaning. You can think of subtasks required to accomplish the main task. For this you can refer to various sources such as direct observation, expert opinion, documentation and so on. You can then observe each subtask and seek to subdivide it, and so on.

A rule, which is particularly suitable when the aim is to design training materials,

is the P × C rule. This says that if the probability of making a mistake in the task

(P) multiplied by the cost of the mistake (C) is below a threshold, then stop expanding. That is, simple tasks do not require expansion (because no one needs training), unless they are critical.

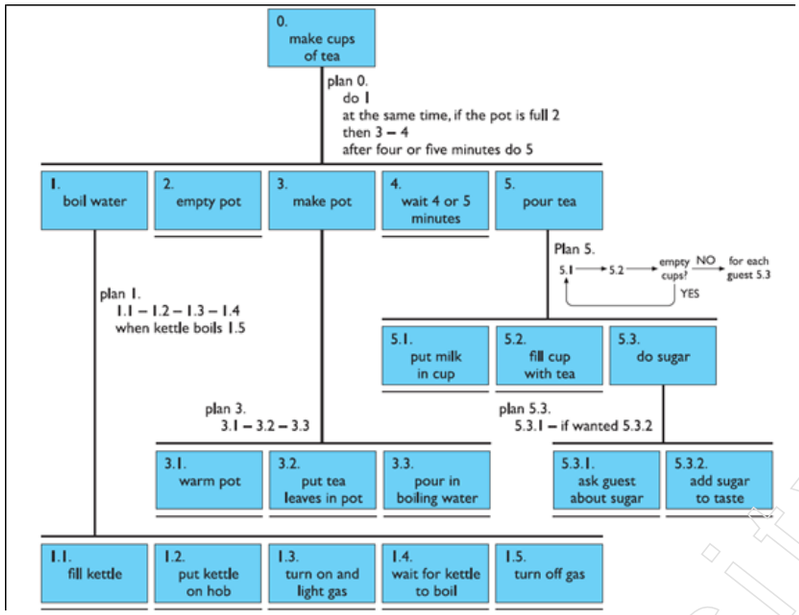
The task hierarchy can be represented in the form diagram as well as text. Following figure shows a task hierarchy for making a tea.



Task hierarchy for making a cup of tea

You can modify task hierarchy with different plans to achieve different goals. Following figure represents modified task hierarchy for making lots of cup of tea.

Notes



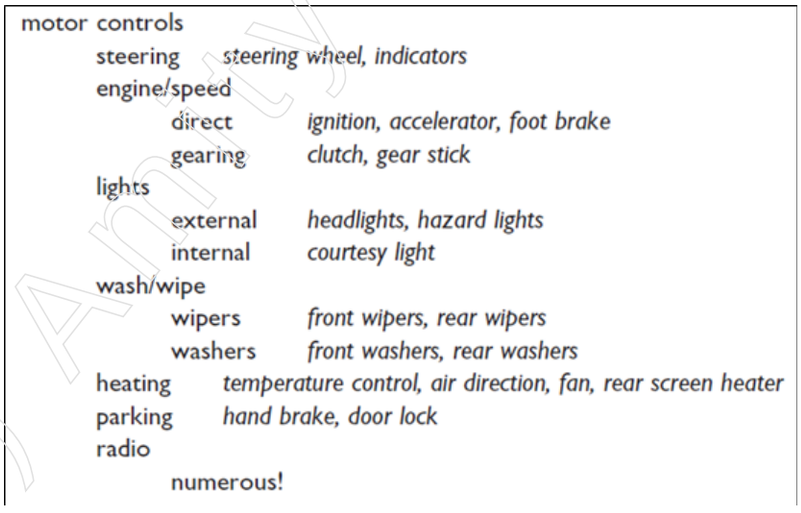
Modified task hierarchy for making lots of cup of teas

4.3.2 Knowledge based analysis

Knowledge base analysis is consisting of list of all objects and taxonomy between them. Taxonomy is same as hierarchical description we find in biology: animals are invertebrates or vertebrates, vertebrates are fish, birds, reptiles, amphibians or mammals, etc.

The aim of this analysis is to understand the knowledge required to perform a task and to help in generating teaching material and assessing the common knowledge between different tasks.

Consider controls in motor car. Following figure 4.3.4 represents taxonomic structure of motor controls.



Taxonomy of car control

Notes

In above figure you can observe every control has exactly one place in hierarchy.

Now we will consider how to build such a hierarchy and how to use it. The car controls are particularly simple, as you can simply get in the car and look for them all.

If we extended our analysis to driving a car in general, you would have to consider more objects like the instruments (speedometer, etc.), the car keys, seat-belts, road signs, other cars etc. Hence, only use of HTA can be hard to know when to stop. However, with any such procedure it is best to start by listing everything you can, later removing items which are felt to be unnecessary. Other sources for forming a list of objects include manuals, transcripts and observation.

After forming the list of objects, it is necessary to build taxonomy between them. For this, analyst can directly ask domain expert because sometimes the taxonomy is already existing form. Another way for this is use of sorting task. Here, analyst can take the help of users for sorting.

The technique task analysis for knowledge description (TAKD) uses three types of branches in TDH (task descriptive hierarchy) taxonomy:

- XOR: object is present exactly in one branch.
- AND: object should be present on both branch.
- OR: object can be present in one, many or no branch.

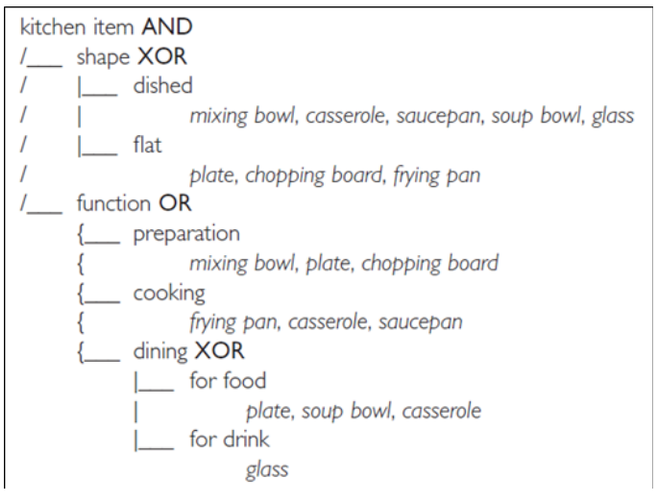
wash/wipe **AND**
- function **XOR**
  - wipe
    - *front wipers, rear wipers*
  - wash
    - *front washers, rear washers*
- position **XOR**
  - front
    - *front wipers, front washers*
  - rear
    - *rear wipers, rear washers*

Example of AND & XOR

kitchen item **OR**
- preparation
  - *mixing bowl, plate, chopping board*
- cooking
  - *frying pan, casserole, saucepan*
- dining
  - *plate, soup bowl, casserole, glass*

Example of OR

TAKD has the uniqueness rule. It states that completed TDH can divide any two specific objects.



Example of uniqueness rule

In above figure tree has been drawn using the characters: '/| {'. These are a characteristic of TDH and represent AND, XOR and OR branches respectively.

Summary:

- Task decomposition is the process of diving task into subtask.
- Knowledge based analysis determines the knowledge of user regarding objects and action involved in the task.
- Indentation denotes levels in task hierarchy and to emphasize this hierarchy numbers have provided.
- A rule, which is particularly suitable when the aim is to design training materials, is the P × C rule
- The technique task analysis for knowledge description (TAKD) uses three types of branches in TDH (task descriptive hierarchy) taxonomy:
  - XOR: object is present exactly in one branch.
  - AND: object should be present on both branches.
  - OR: object can be present in one, many or no branch.

Exercise:

- Produce a high-level hierarchical task analysis showing how you would find information on a website. Assume the site has a search facility as well as normal links.

Notes

Notes

Unit - 4.4: Dialog Notations and Semantics

Recall Session:

In the previous unit, you studied about:

1. Task decomposition and knowledge-based analysis.

Unit Outcome:

At the end of this unit, you will be able to:

1. Understand dialog notation and semantics

4.4.1 Introduction

Dialog is a conversation between two or more users. While designing the interface, dialog has particular meaning. Precisely, it is a structure of conversation between human and computer. There are three level in computer language:

- Lexical: it is a lowest level. It includes shape of icons on the screen and actual keys pressed. It is corresponding to sounds and spellings of words in human language.
- Syntactic: it includes the order and structure of input and output. It is corresponding to grammar of sentence correction in human language.
- Semantic: it includes meaning of conversation in terms of its effect on computer's internal data structure. It is corresponding to meaning drawn by various users during conversation in human language.

Consider following program:

```
rate = 10
term = 25
print “Our current interest rate is 10%”
print “What is your annual salary?”
input salary
max_loan = 3 * salary
print “How much do you want to borrow?”
input amount
if amount > max_loan
then print “That is too much money”
print “Please consult our financial advisor”
goto finish
end if
repeat forever
print “Our standard term is 25 years.”
print “Do you want this (yes/no)?”
input answer
```

Notes

```
if answer == “yes” goto calc
if answer == “no” goto rd_trm
print “You must answer yes or no”
end repeat
rd_trm: print “What term do you require (years)?”
input term
calc: r = ( 100 + rate ) / 100
payment = r^term * ( r - 1 )
amount / ( r^(term-1) - 1 )
print “Monthly repayment is ”, payment
finish: stop
```

In above program, dialogs of user are mixed with program. In this program the biggest hurdle is to check the answer: either yes or no. This kind of interactive program is consisting of so many errors and corrections.

Imagine you have been asked to analyse the dialog in some way: for instance, to list all the possible sequences of user inputs and system responses, or to tell the user how to get the repayments on a 15-year loan. The mixing of user and system choices and the convoluted nature of the program structure make this difficult.

There are two reasons for using a separate dialog description notation: ease of analysis and separation of the interface elements of the program from the actual calculations (semantics). There is a third reason for using a special notation is to write down the dialog before a program is written. This allows the designer to analyse the proposed structure, or use a prototyping tool to execute the dialog. A dialog notation is also a way for members of a design team to talk about the design and eventually for the designer to pass on the intended dialog to the programmer of the actual application. Thus, dialog notations often form an integral part of prototyping methodologies and tools.

4.4.2 Dialog Semantics

Dialog semantics have two aspects:

1. Inwards: it is towards the application.
2. Outwards: it is towards the presentation.

Consider following approaches for linking dialog and semantics:

- Notation-specific semantic: This is a special purpose semantic, designed as a part of a dialog notation.
- Links to programming languages: It consists of linking of pieces of programming language to form a dialog.
- Links to specification notations: Similar as above but uses specific notations instead of programming language.

1. **Notation-specific semantics:** Augmented transition networks (ATNs) are a form of state transition network. In the ATN, the system is assumed to hold a set of registers,

Notes

storage locations which the transition network can set and test. The arcs have a condition as well as the event; the condition can refer to the system's registers and the arc is only followed if the condition is true and the event occurs. The system response is augmented to include not only feedback and display, but also the setting of registers. These registers can be used simply to describe more complex dialogs, for example a cash dispenser which retains your card after three wrong numbers. They may also be used to communicate with the application and to hold values from the mouse.

Production rules come in many variants and the link to the semantics is equally varied. Often the system's memory contains variables which can be used by the system to store input values such as the mouse position, and can then be examined by the conditions and actions of the rules. These variables may also be used to communicate with the underlying application, or the functions of this may be invoked directly by special forms of action. For instance, the following is a production rule which, when the user clicks within a target region, puts a dot at the mouse location and invokes the application routine 'another_point'.

click_at(x,y) → dot_at(x,y), call another_point(x,y)

2. **Links to programming languages:** Often dialog notations are 'attached' to a conventional programming language. For example, input tools, a regular expression-based notation, uses C to express the dialog semantics. The input tool description consists of 'tool' definitions, including regular expressions intermingled with normal C code. For example, the tool to read a number is as follows:

```
tool number
{ char buf[80];
int index;
int positive;
input { ( digit* + sign; digit; digit* ) ; return }
tool digit
{ input { key:| key_c>='0' && key_c<='9' | }
if ( index < 79 ) /* append character to string */
{ buf[index] = key_c;
index = index + 1;
echo(key_c);
}
}
tool sign
...
tool return
{ input { key:| key_c == '\n' | }
...
}
...
}
```

Notes

Input tools uses its own regular expression syntax and has additional operators. Sequencing is denoted by semi-colon ';' rather than simple juxtaposition, and the '+' symbol is used for choice (like '|' in BNF). The expression 'key:|condition|' is a postfix guard: the expression only matches if the condition is true. The specific input tools' expressions are as follows. The keyword tool introduces a new tool, which is similar to a non-terminal in a BNF grammar, and the regular expression which it denotes is enclosed in the input statement. The tools are arranged in a scoped hierarchy, so that the digit, sign and return tools are private to the number tool. The call to echo simply echoes the character back to the user. Finally, key is a primitive tool which matches a single character read from the keyboard; the actual character read is stored in the global variable key_c. Notice how the sub tool digit communicates its results back to the main tool using shared variable buf. This and the way it accepts values from the key tool are rather untidy.

3. **Links to specific notations:** Alexander's executable specification or prototyping language SPI (Specifying and Prototyping Interaction) is divided into two parts:
   - eventCSP
   - eventISL

eventCSP is a dialog notation based on CSP and eventISL describes dialog semantics. In eventISL, there is a corresponding definition for each event. It is partly standardized and partly depend on host language chosen. The first host language was me-too, a formal specification notation based on VDM, but a C variant is also available. The part which is independent of the host language consists of several elements: a clause giving the global variables used and updated by the event, a precondition expressing when the event can occur, and output and input parts. The host language part simply describes the updates and the precise outputs.

Consider the following eventCSP description of a login sequence:

Login = login-mess → get-name → Passwd

Passwd = passwd-mess → (invalid → Login

[ ] valid → Session)

Session = (logout → Login

[ ] command → execute → Session)

A typical unsuccessful login sequence might be

login: fred

passwd: b9fGk (invisible)

Sorry bad user-id/password

We will not consider the detailed semantics for the commands during the session, but will give the eventISL descriptions of the other events. The two events login-mess and get-name handle the first line of the above dialog:

event: login-mess =

prompt: true

out: "login:"

event: get-name =

Notes

uses: input

set: user-id = input

The first event prints the prompt 'login:' (the out: clause) and says that user input is required (the prompt: clause). This user input is stored in a special variable called 'input'. The second event uses the input (which will be set to the name the user enters) and merely sets the variable 'user-id' to it. Note that 'user-id' is set to a new value by the event, but any previous value is not used.

The sequence for getting the password is similar except that there are two options depending on whether the user has typed a valid password or not:

event: passwd-mess =

prompt: invis

out: "passwd:"

event: valid =

uses: input, user-id, passwd-db

when: passwd-id = passwd-db(user-id)

event: invalid =

uses: input, user-id, passwd-db

when: passwd-id ≠ passwd-db(user-id)

out: "Sorry bad user-id/password"

The password prompt is identical to the login prompt except that no echoing is required. However, the last two events demonstrate two additional features. As well as the user input variable, they also use the variable 'user-id', which was set by the get-name event, and the variable 'passwd-db'. This is assumed to be a database of passwords, so that 'passwd-db(user-id)' is the correct password for the user. The two events also have a 'when' clause. This is a precondition, which specifies what must be true for the event to occur. So the 'valid' event can only occur when the user has typed a correct password and 'invalid' only occurs when it is incorrect.

Summary:

- Dialog is a conversation between two or more users.
- Lexical is a lowest level. It includes shape of icons on the screen and actual keys pressed. It is corresponding to sounds and spellings of words in human language.
- Syntactic includes the order and structure of input and output. It is corresponding to grammar of sentence correction in human language.
- Semantic includes meaning of conversation in terms of its effect on computer's internal data structure. It is corresponding to meaning drawn by various users during conversation in human language.
    - Dialog semantics have two aspects:
    - Inwards: it is towards the application.
    - Outwards: it is towards the presentation.

Unit -4.5: Models of the System

Notes

Recall Session:

In the previous unit, you studied about:

1. Dialog notations and semantics.

Unit Outcome:

At the end of this unit, you will be able to:

1. Understand standard formalism and interaction models

4.5.1 Introduction

In previous unit, you studied about dialog notations and semantics. In this unit you will learn about modelling of the semantics of interactive system. Dialogs tell us about the user actions that what user actions are legal at any point but we are interested in knowing what user actions do to the system.

In this unit, you will learn standard formalisms which specifies interactive system. Standard formalisms are part of software engineering. It can be a way for interface designer to represent their ideas and communicate with system developer.

You will also learn about interaction models which describes general properties of the system.

4.5.2 Standard Formalism

Standard formalisms are general computing notations to specify a particular system. It can be used to specify interactive system. There are various standard formal methods as following:

- Model based: Model based methods describe system states and operations.

  Example: Z and VDM

- Algebraic: It describes effects of sequence of actions.

  Example: OBJ, Larch, ACT-ONE

- Extended logics: It describes when things happen and who is responsible.

  Example: temporal and deontic logics

It uses software engineering formal notations. For communication, it uses notation including common language, remove ambiguity, succinct and precise. For analysis it uses notations like internal consistency, external consistency. External consistency has to be checked with respect to eventual program and requirements like safety and security.

Model based methods uses general mathematics such as numbers, sets and functions. It uses model base methods to fine state and operations on state. Model based methods describes states using variables. Variables are of four types:

Notes

- basic type
- individual item from set
- subset of bigger set
- function

4.5.3 Interaction Models

General computational model was not designed with respect to user. You need models which act as a bridge between software engineering formalism and HCI.

Interaction models are of three types:

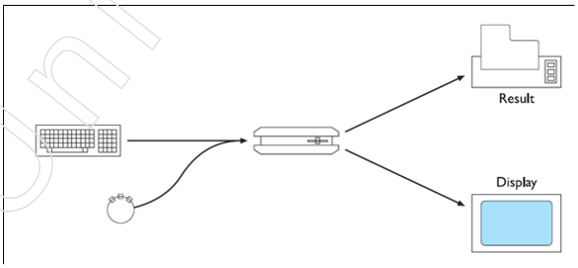
- Formal
- Informal
- Semi-formal

Formal model is a PIE model. It is used for expressing general interactive properties to support usability.

Informal models act as an interactive architecture. It is used to motivate separation and modularisation of functionality and presentation.

Semi-formal models used for status event analysis. It is useful for viewing a slice of an interactive system that spans several layers.

PIE Model



Inputs and outputs from single user

It is black-box model. It does not represent internal structure instead it represents input and output from the user. This concept is illustrated in above figure. User give input through sequence of commands. Commands can be given through keyboard, mouse movement or mouse click. Let us consider 'c' as a set of commands and 'P' as a sequence. So we can relate sequence and commands using following equation:

P = seq C (P is equal to sequence of C)

The output is nothing but effect. Effect is consist of display and result. Consider 'E' as a set of effects. Given the set of sequence, you can map interpretation with effect as following:

I : P -> E

More formally, you can summarize it as shown in following figure.

Notes

[C;E;D;R]
P == seq C

I : P → E
display : E → D
result : E → R

Alternatively, we can derive a state transition function from the PIE.

doit : E × P → E

doit(I(p), q) = I(p q)
doit(doit(e, p). q) = doit(e, p q)

WYSIWYG is clearly related to what you see is what you get. Following are the limited scope general properties which supports WYSIWYG.

- Observability: It is about what you can determine about current state of the system from display.
- Predictability: This is about what you can predict about future behaviour.

Predictability is a special case of observability.

There are some issues for PIE properties:

- Insufficient: PIE defines necessary properties but not sufficient for usability engineering.
- Generic: PIE properties can be applied to any system.
- Proof obligations: This is for system defined in SE formalism.
- Scale: It is about how to prove many properties of large system.
- Scope: this property limits the applicability of certain properties.
- Insight: whatever gained from abstraction is reusable.

Summary:

- Standard formalisms are general computing notations to specify a particular system.
- There are various standard formal methods as following:
  - Model based
  - Algebraic
  - Extended logics
- Model based methods describes states using variables. Variables are of four types:
  - basic type
  - individual item from set
  - subset of bigger set
  - function

Notes

- Interaction models are of three types:
  - Formal
  - Informal
  - Semi-formal
- It is black-box model. It does not represent internal structure instead it represents input and output from the user.

Unit - 4.6: Hypertext, Multimedia and World Wide Web

Notes

Recall Session:

In the previous unit, you studied about:

1. Standard formalism and interaction models.

Unit Outcome:

At the end of this unit, you will be able to:

1. Evaluate hypertext, multimedia and world wide web

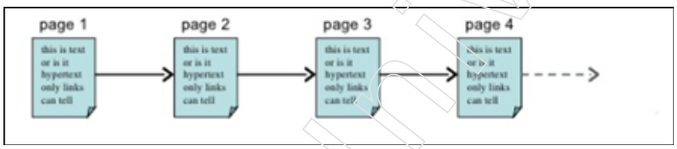
4.6.1 Introduction

Increase in desktop computing power has resulted into rapid growth of multimedia industry. Because of World Wide Web, computers, information and people across the world are connected to each other.

In this unit, you will learn about features of hypertext, multimedia and World Wide Web.

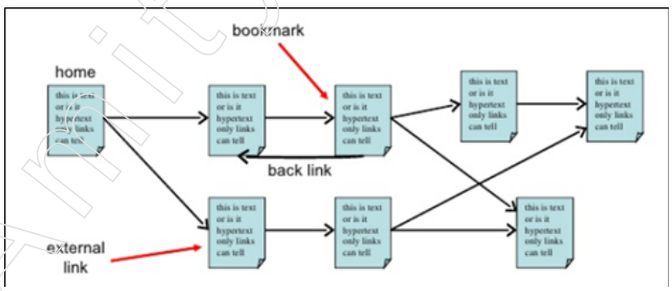
Hyper is a rich content of graphics, audio, video, computation and interaction.

Text is strict linear progression on the reader. It is shown in following figure.



Linear progression

But hypertext is not just linear. It consists of blocks of text (pages). Links between the pages forms a network or mesh. User follow their own path through information using this network. That network is represented in following figure.



Structure of hypertext

When hypertext combine with media it become hypermedia which is not just a text. Hypermedia has number of applications in various domains. Hypermedia is useful for illustration, photograph, video and sounds. Hypermedia can be a link or hotspot in media. It includes areas of pictures, time and locations of video. Sometimes hypermedia is also called as multimedia.

Notes

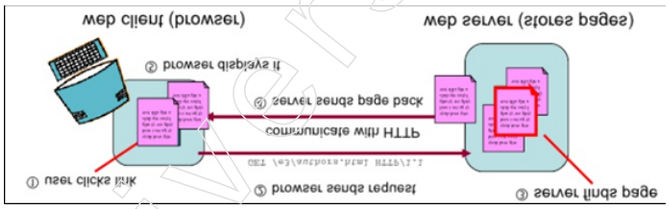
When motions are added to images, it resulted into animation. Animation is used for things that changes in time such as digital and analogue displays and live displays also. It is also useful for showing status and progress.

Animation also used for education and training purpose. Because of animation, students can see that things are happening and class become very interesting and entertaining. Animation is useful for data visualization.

Now you will learn about World Wide Web. The web is consisting of protocols and standards. It uses HTTP to carry the information over the internet. Web uses HTML, XML and graphics format for content display. To show the search results, web requires browser. Sometimes browser has to add plug-ins for more functionality and compatibility.

Initially web was used for only research purpose. Now a days it is used in every field such as corporate, government, commerce and entertainment, advertising, community etc.

Web works on client server mechanism. Web is distributed on different machines across the world.



Client server mechanism

Above figure represents client server mechanism. In this mechanism, pages are stored on servers. Client requests the pages by searching information through browser. After receiving the request from user client, server finds the pages. After searching the pages, as a response to user client request, server sends requested page back to the browser and browser displays it to the user.

Summary:

- Hypertext consists of blocks of text (pages). Links between the pages forms a network or mesh. User follow their own path through information using this network.
- Web works on client server mechanism.
- Web uses HTTP to carry the information over the internet.
- Web uses HTML, XML and graphics format for content display.

Activity:

1. A piece of icon or image on a web page associated with another web page is called as _____

   a. url

   b. hyperlink

Notes

c. plugin

d. extension

2. Web browser is ……..

a. A program that can display a web page

b. A program used to view html documents

c. Which enables user to access the resources of internet

d. All of the above

3. URL stands for …….

a. Unique reference label

b. Uniform reference label

c. Uniform resource locator

d. Unique resource locator

4. PIE model is a black-box model.

a. True

b. False

5. Email is a linear kind of execution.

a. True

b. False

Answer Keys (Exercise):

Question	Answer	Question	Answer	Question	Answer
1	b	2	d	3	c
4	a	5	b		

Notes

Module - V: Research Framework

Course Contents:

- Speech Interfaces
- Information Visualization
- Ubiquitous Computing
- Case Studies

Key Learning Objectives:

At the end of this module, you will be able to:

1. Understand various speech interfaces
2. Illustrate various information visualization techniques
3. Understand ubiquitous computing
4. Analyse various case studies

Structure:

Unit - 5.1: Speech Interfaces

Notes

Unit Outcome:

1. At the end of this unit, you will be able to:
2. Understand various speech interfaces.

5.1.1 Introduction

Framework is the base of strong research, which binds together different topics. It helps researcher to explore subtopics, design ideas and generate new research.

Research framework helps to identify and follow the process towards achieving intended goal and objective. Research framework is based on strong literature survey and various case studies.

In this unit we are going to discuss various case studies regarding speech interface, data visualization and ubiquitous computing.

Speech interface is the application that allows interaction between human and voice enabled device. It is also important to understand 3D displays as they used for data visualization. In old days, computing was fixed like location, device, format was fixed but with the help of ubiquitous computing it is possible to compute anywhere anytime with any format.

5.1.2 Speech Interfaces

As we have already discussed, speech interface is the software interface which allows interaction between human and voice enabled device.

For example, Google assistant, Siri, Alexa.

Speech interface uses speech recognition technology to use and mimic human speech. To design effective speech interface, it is necessary to understand users of the interface, their experience and main thing is purpose of speech interface.

In today's digital world, speech interfaces are common in use. Speech interfaces are embedded in various devices that people use in their day to day life.

Example: smartphones, smartwatches, tablets etc.

It is important to consider user experience while designing speech interface because if you build the interface without considering their experience then might be user can feel interface either boring or too difficult to understand and explore. Hence, it is important to keep user experience in mind.

In addition to user experience, you to think about target audience and how they will interact with interface.

You can use following practices while designing speech interface:

- **Ask better question:** Instead of asking a question, if interface is collecting an information by using a statement then users don't understand how to give

Notes

response. Therefore, asking a question to collect the information is a good practise.

e.g. “may I know your birthdate?” is easier to response than “I would like to your birthdate in the form of year/month/date”.

- **Take advantage of prototyping tools:** Prototyping tools are the tools which helps designer to increase the speed of product implement and make it more effective. With the help of prototyping tools, it is easy to test the interface.

  For example, with the help of Botsociety you can create a high-fidelity prototype of your voice assistant and test it with a beta group of users to identify issues with your speech interface. This makes it easy to detect problems that can lead to poor user experience.

- **Explain the function they are using:** Provide the information of function of speech interface when users are using it.
- **Let users understand what they can do:** Make users aware about capabilities of interface and what users can and cannot do with it.

The most important is user must feel comfortable while using speech interface and meet their goal.

Summary:

- Speech interface is the software interface between human and voice enabled devices.
- To design effective speech interface, it is necessary to understand users of the interface, their experience and main thing is purpose of speech interface.
- It is important to consider user experience while designing speech interface.
- Asking a question to collect the information is a good practise.
- User must feel comfortable while using speech interface and meet their goal.

Activity:

1. Illustrate real life applications of speech interface.

Unit - 5.2: Information Visualization

Notes

Recall Session:

In the previous unit, you studied about:

1. The need of speech interface.
2. Basics of speech interface.

Unit Outcome:

At the end of this unit, you will be able to:

1. Analyse scientific and technical data
2. Understand structured information

5.2.1 Introduction

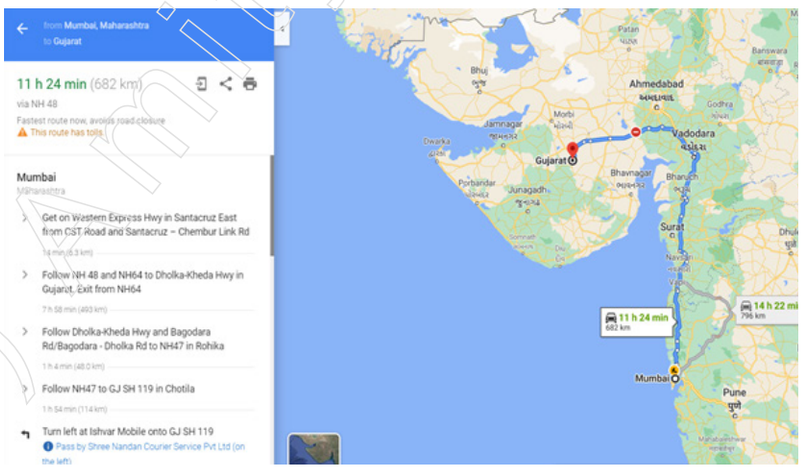
Information visualization is the process of representing data in visual and meaningful way for better understanding of users. It is a study of visual representation of abstract data. Abstract data can be numerical and non-numerical. Data visualization is important because visual summary of information is easier way to find patterns and trends than going through thousands of rows and columns on a spreadsheet. It is also used to understand complex relationships within data.

Basically, data visualization is divided into two types:

- Exploration
- Explanation

Exploration helps to find the meaning of data and explanation inform the meaning of data. Before visualization process starts, you must know your target users and their need. This can be achieved using qualitative search.

Consider following image. It shows the map generated in google map. Google map provides two simple ways of representing route from Mumbai to Gujrat.



Notes

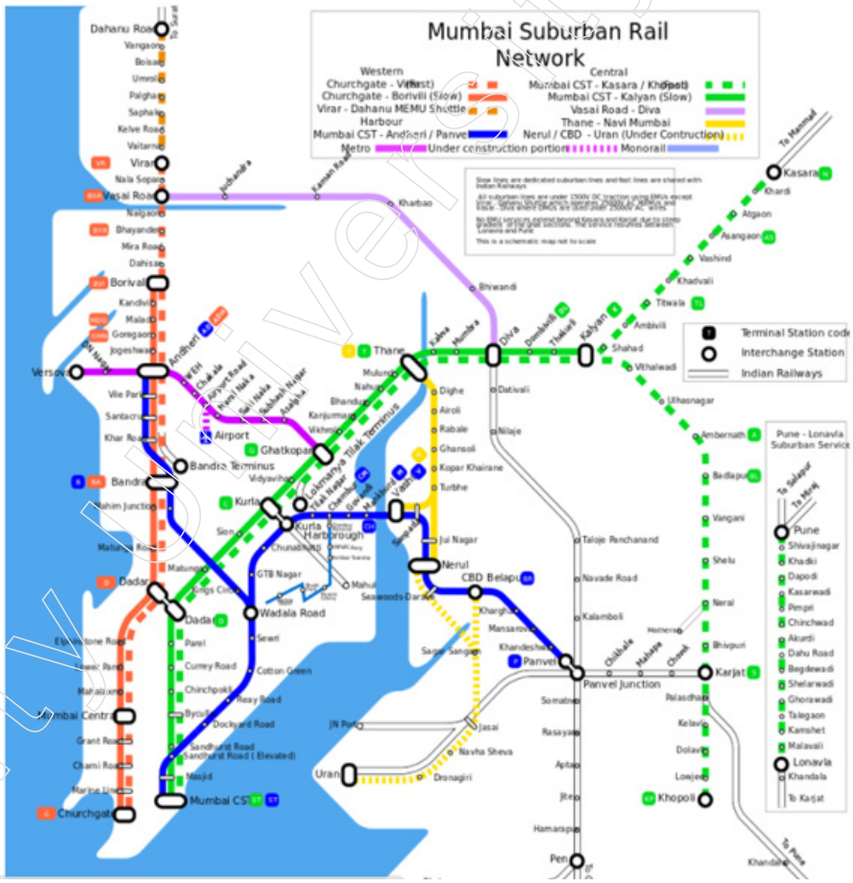
First representation consists of written instruction about how to reach Gujrat from Mumbai. Second presentation is the image of the route shown on a map. Each representation represents different values to different group of people.

The first representation is useful for the people who want to travel from Mumbai to Gujrat directly. Whereas second representation is useful to those people who do not want to reach Gujrat directly. In between they want to know what is on the way.

Both representations are an examples of data visualization. The first, depends on clear instruction and minimum graphical representation. Second representation depends on graphical representation.

Applications of information visualization are as following:

- **Presentation:** Visual representation is most useful to understand and explain the complex concepts.



Above image represents Mumbai rail network. This network is complex hence not easy to understand without presentation. But the graphical presentation given in image 5.2.2 is easy for people to understand it.

- **Explorative Analysis:** With the help of information visualization, explorative analysis can find relationships in data may exist.
- **Confirmation Analysis:** Information visualization used to confirm our understanding and analysis of data.

Notes

Therefore, we can conclude that information visualization is used to find out useful information from data. It is also used to find relationships between data and to confirm our understanding. It provides best way of presentation which is easy to understand.

5.2.2 Scientific and Technical Data

3D representation of scientific and technical data can be categorized using number of dimensions in virtual world. However, most engaging images are where all three dimensions have some physical validity.

Consider the example of virtual wind tunnel. In following images shows physical wind tunnel and virtual wind tunnel.

Physical wind tunnel

Virtual wind tunnel

Physical wind tunnel is a large tube with air blowing through them. The tunnel is used to demonstrate the actions of an object flying through the air or moving along the ground. Wind tunnels are mostly used by researchers to study about how an aircraft will fly.

In a physical wind tunnel, an accurate model of an aircraft is constructed and then subjected to winds that, when appropriately scaled, correspond to realistic situations. The intention is to investigate patterns of air movement and pressure for example, to find out the places where turbulence form. Air is invisible hence small ribbon may be attached to the surface of aircraft. Small bubbles released into the chambers can expose invisible airflow.

Notes

In virtual wind tunnel, air movements are calculated using equations of fluid dynamics. An engineer can see simulated aircraft using VR goggles and can move around virtually. By moving around virtually, he can explore more parts of aircraft.

Model making is expensive and time consuming. This can be overcome by virtual wind tunnel. Virtual wind tunnel can save the money and increase the rate at which changes can be made and evaluated.

5.2.3 Structured Information

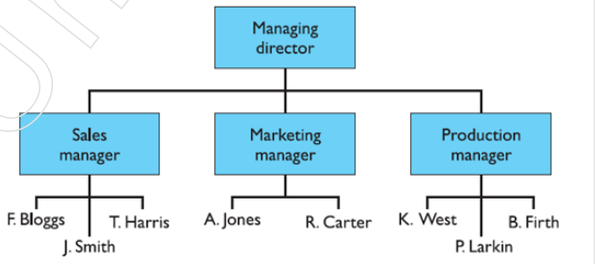
Scientific data are numerical so it can be easily mapped onto a dimension in virtual space. On the other hand, data in information system have many attributes and structures. It can be hierarchies, networks and free text. Free text is the most complex part of structured information.

Examples of hierarchies: file trees and organization chart

Examples of networks: program flow chart, hypertext structures.

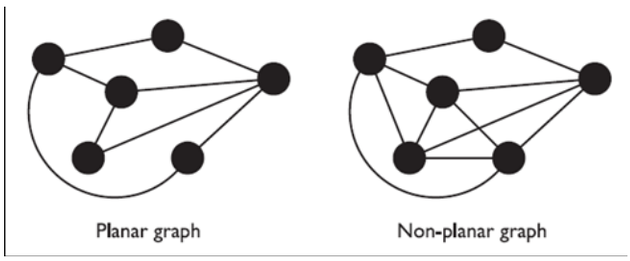
For hypertext structure, number of links need to be traversed between two nodes. For free text similarity of two documents depends on the proportion of words they have in common. To map the data points into two or three dimensions, various techniques can be used. These techniques include statistical multi-dimensional scaling, self-organizing neural networks and simulated gravity. Although the dimensions that arise from these techniques are arbitrary, the visual mapping allows users to see clusters and other structures within the data set.

Networks and hierarchies in two dimensions can be represented in various standard ways.



Two-dimensional organization chart

Above figure represents typical organization chart. And figure represents two dimensional network layout.



Two-dimensional network layout

Summary:

- Information visualization is the process of representing data in visual and meaningful way for better understanding of users.
- Data visualization is important because visual summary of information is easier way to find patterns and trends than going through thousands of rows and columns on a spreadsheet.
- Information visualization is used to find out useful information from data.

Notes

Notes

Unit - 5.3: Ubiquitous Computing

Recall Session:

In the previous unit, you studied about:

1. The need of information visualization.
2. Scientific and technical data.
3. Structured information.

Unit Outcome:

At the end of this unit, you will be able to:

1. Understand physical interaction.
2. Illustrate application themes for ubicomp.

5.3.1 Introduction

Ubiquitous computing allows users to use computing from anywhere. Rather than force the user to search out and find computer's interface ubiquitous computing suggest that interface itself can take the responsibility of locating and serving the user.

Mark Weiser is the father of terminology ubiquitous computing. His vision of ubiquitous computing has given birth to many technologies, it is important to realize that main motivation behind his vision how ubiquitous computing is affected on user experience?

You can define ubiquitous computing as any computing technology that permits human interaction away from a single workstation. It includes pen-based technology, handheld or portable devices, large scale interactive screens, wireless networking infrastructure and voice or vision technology.

Ubiquitous computing has many challenges. You will focus on three of them.

- Defining the appropriate physical interaction experience.
- Discovering general application features.
- Theories for designing and evaluating the human experience within ubicomp.

5.3.2 Defining the Appropriate Physical Interaction Experience

Ubiquitous computing assumes physical interaction between humans and computation will be less like current desktop keyboard/mouse/display system and more like the way human interact with physical world. When humans communicate with each other they speak, gesture and use writing methods. Ubiquitous computing has led to various changes to the input, output and interactions that define human experience with computing.

Towards Implicit Input: Input has changed to beyond the explicit nature of textual input from keyboards and selection from pointing devices to variety of data types. This

Notes

has resulted into variety of input technology as well as shift from explicit way of input to more implicit type of input.

Implicit input means natural interaction with physical environment provides sufficient input to different attendant services without any further user intervention.

Toward multiscale and distributed output: The use of ubiquitous computing capabilities into everyday life also requires output technologies and techniques. Designers of information appliances such as PDA and future home technologies must introduce various forms of technology. Now output is no longer in the form of desktop/ laptop visual displays.

Different sizes of visual displays both smaller and larger than the desktop is being distributed.

Seamless integration of physical and virtual worlds: An important feature of ubicomp technology is integration of physical and virtual world. There are many examples which demonstrates how electronic information overlaid upon real world which is defined as augmented reality. Augmented reality only affects the output. When both input and output begin to intermixed, you begin your journey towards integration of physical and virtual world.

5.3.3 Application Themes for Ubicomp

In HCI, many applications are researcher's oriented and look for profit at any cost. The killer can give them that profit and result into large investment in infrastructure that will produce various ubicomp applications.

Studies conclude that person to person communication is a great killer app for ubicomp because it has caused large investment. Irrespective to killer app, the ubicomp must think human point of view. In the app all the services must be available when and as needed and all of them must work as intended without expert human intervention.

There are many features of ubicomp applications like ability to easily capture and store memories of live experience and serve them for later use. These features result in change in themes of ubicomp applications. Let us consider two established themes.

Context Aware Computing: Active Badge and the Xerox PARCTab, both are location aware appliances. These devices use simple piece of context, user location and provide valuable services like automatic call forwarding for a phone system, automatically updated maps of user locations. These location aware appliances demonstrate the linking of implicit human activity with computational services.

Location of identifiable entities like human is the most common in ubicomp applications. The most popular applications are GPS based car navigation systems and handheld tour guide system.

In addition to position (where) and identity (who), there is more context which is need to be consider. It is as following:

- When: It indicates time. With the help of time system can capture how long a person was at particular location.

Notes

- What: The interaction in current system, either assumes what the user is doing or leaves the question open. But to build successful application, it is necessary to interpret user action to extract information about them. It can be done using "cookies". Cookies describe user activity on World Wide Web.
- Why: Understanding "why" users are doing it, is more complex to understand than "what" they are doing. May be using other forms of contextual information could give answer to this.

An obvious challenge of context-aware computing is making it truly ubiquitous.

Automated Capture and Access

Many people spent their life in listening to and recording, more or less accurately and then try to remember that important information. This is very inefficient and danger to record manually especially when information is about multiple streams. To overcome this problem, there should be tools which can support automated capture and access to live experience. It will reduce the burden of doing something at which humans are not good (e.g. manual recording) and therefore they can focus attention on activities they are good at.

There are number of capture applications in different domain including education and project management to support individuals or groups.

5.3.4 Understanding Interaction in Ubicomp

With widespread of ubicomp, it is necessary to understand interaction in ubicomp. Traditionally, research and evaluation in HCI have been represented by the Model Human Processor theory of human cognition and behaviour. This model focusses on three independent units of sensory, cognitive and motor activity where each unit maintains its own working store of information. As the application of computers has broadened, designers have turned to models that consider the nature of the relationship between the internal cognitive processes and the outside world.

Three main theories that focus on the 'in the world' nature of knowledge is being explored within the ubicomp community as guides for future design and evaluation: activity theory, situated action and distributed cognition.

Activity theory recognizes concepts such as goals, actions and operations. Situated action gives importance to the improvisational aspects of human behaviour and de-emphasizes a priori plans that are simply executed by the person. Distributed cognition also de-emphasizes internal human cognition, but it turns to a systems perspective where humans are just part of a larger system.

5.3.5 Evaluation Challenges in Ubicomp

It is necessary to assess the utility of ubicomp solution. Researchers have recently started discussion about development of assessment and evaluation techniques that meets demands of ubicomp. Because of slow evolution of use of ubiquitous technology, the rate of development of these techniques is very slow. To understand effect of ubicomp on day to day life, it is important to understand how technologies will provide real human need.

Notes

Formative and summative evaluation of ubicomp system is difficult. A number of significant challenges need to be addressed in order to develop appropriate assessment methods and techniques. Even it is not clear yet how to apply qualitative and quantitative methods of evaluation.

The technology used to create ubicomp systems is often on the cutting edge and it is difficult to create reliable and robust systems that support some activity on a continuous basis. Consequently, a good portion of reported ubicomp applications work remains at this level of demonstrational prototypes that are not designed to be robust, although there are notable exceptions.

A number of researchers are seeking to roll out ubiquitous devices into a range of settings, such as museums, outdoor city centers and the home. These researchers are creating 'living laboratories' for ubicomp research by creating testbeds that support advanced research and development as well as use by a targeted user.

Following are the evaluation challenges faced by ubicomp:

- **Applicability of matrices:** Ubicomp applications may not have the same goals like other technologies. Therefore, it may require measurement of different metrics of success. Appropriateness of matrix may vary as per the system being evaluated.
- **Scale:** Ubicomp system must handle the issues of scale across multiple devices, multiple locations or over long period of time or across multiple users.
- **Ambiguity:** Ubicomp systems are sensing-based system. Sensing based system can lead to serious usability problem. Researchers have not found best user interface mechanism to handle sensing issue like ambiguity.

Summary:

- Ubiquitous computing allows users to use computing from anywhere. Rather than force the user to search out and find computer's interface ubiquitous computing suggest that interface itself can take the responsibility of locating and serving the user.
- Mark Weiser is the father of terminology ubiquitous computing.
- Implicit input means natural interaction with physical environment provides sufficient input to different attendant services without any further user intervention.
- When both input and output begin to intermixed, you begin your journey towards integration of physical and virtual world.

Notes

Unit - 5.4: Case Studies

Recall Session:

In the previous unit, you studied about:

1. Physical interaction.
2. Application themes for ubicomp

Unit Outcome:

At the end of this unit, you will be able to:

1. Understand various case studies.

Case study 1: Lab Study

Effective eye contact can be problem for live video and also for pre-recorded video, and this case study shows how even small differences in the physical positioning of devices can have a marked impact on the quality of video communication.

The importance of eye contact has been studied since the very earliest days of video communication, including techniques such as the use of half-silvered mirrors in the Video Tunnel and multiple small screens in Hydra, as well as high-cost commercial video conference rooms. The basic problem is that it is hard to both look at a camera and look at a screen. No matter where you place the camera the other party sees you looking slightly up, down, left or right.

Familiarity with video calls on mobile phones and laptops has meant some of the immediate oddness of this has faded, but still leads to difficulties in reading expressions and attention.

This case study is being written in the midst of the Civid-19 pandemic when many are having to adapt to meetings, lectures and lessons delivered virtually, so the issues are particularly pressing at this moment, but will continue to be so as it is likely the changes in education and society will persist beyond the immediate crisis.

In a studio setting presenters will often use teleprompters that use half-silvered mirrors to enable the script to be read whilst looking directly at the camera. Furthermore both presenters and the production team are used to creating engaging content. For low-budget desktop video, we may need more rough and ready techniques.

When Alan first wanted to create video materials some years ago his first attempts used the laptop's built in camera (see below). This is positioned just above the screen and so quite close to the screen, so his eyes were looking slightly downwards towards the material on the screen, but not excessively so.

Notes

In fact, the problem was almost the opposite. Because the camera was close to the screen for the whole video, Alan found he was effectively staring directly at the camera (below). In face-to-face conversations the speaker only spends about 1/3 of the time looking at the listener and the rest of the time looking slightly away, often whilst thinking. Staring continually at the listener is somewhat unnerving!

The 'talking head' video was inset alongside the screen capture and as a listener one finds oneself looking at the captured screen, not always in Alan's eye; so the fixed stare is not quite as bad as it first seems but still not ideal. This first video is in the Dynamics of Trust case study, see for yourself what you think.

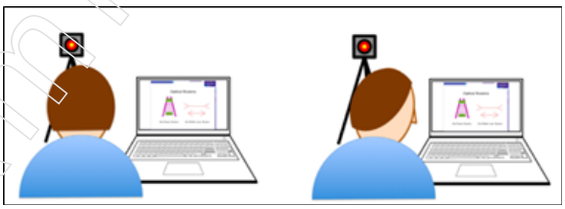
For later videos, a webcam was used, partly simply to improve the raw quality of the captured video, but this also allowed flexibility in positioning the camera. The revised layout can be seen below.

See how the camera has been placed slightly to Alan's left and slightly above the level of the screen.

At first this seems like a bad idea. When looking at the screen he loses eye contact with the audience. However, this is of course precisely what happens during a live lecture when the presenter glances down at notes or the computer to change slides.

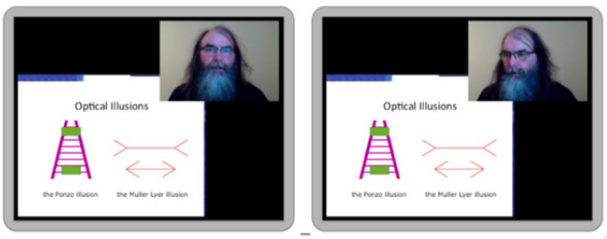
Very quickly Alan began to see the camera as 'the audience', and would alternate talking directly to the audience and glancing down at his laptop, exactly as in a physical presentation.

As can be seen in the schematics, as he looked towards the laptop his head would drop slightly and his eyes would look slightly down and to his right.



When edited (lightly!), the talking head was placed on the top right of the screen (below), in a way that mirrored to physical placement of camera and screen. See how the change in head and eye position naturally points towards the slide when Alan was himself looking at it.

Notes



This of course still not high production value video (!), but the difference that this small manipulation made was dramatic.

References

1. "Human-Computer Interaction" by Alan Dix, Janet Finlay, Gregory Abowd and Russell Beale, 3rd edition, Pearson Publication.
2. "Human-Computer Interaction: Fundamentals and Practice" by Gerard Jounghyun Kim, 1st Edition.