COURSE GUIDE

NATIONAL OPEN UNIVERSITY OF NIGERIA

FACULTY OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

CIT 353
INTRODUCTION TO HUMAN COMPUTER INTERACTION

National Open University of Nigeria
Headquarters
University Village
Plot 91, Cadastral Zone,
Nnamdi Azikiwe Express way
Jabi, Abuja

Lagos Office
14/16 Ahmadu Bello Way
Victoria Island, Lagos

e-mail: centralinfo@noun.edu.ng
website: www.nouedu.net

Published by
National Open University of Nigeria

Reprinted 2020

ISBN: 978-058-212-2

COURSE CODE CIT 353

COURSE TITLE INTRODUCTION TO HUMAN COMPUTER INTERACTION

COURSE DEVELOPER JOSHUA ADEKUNLE AKINJOBI (CHF)
CRAWFORD UNIVERSITY, IGBESA,OGUN STATE, NIGERIA

TABLE OF CONTENTS

MODULE 1: SURVEY OF HUMAN-COMPUTER INTERACTION CONCEPTS, THEORIES AND PRACTICE

UNIT 1 CONCEPTS, THEORIES AND HISTORY

Table of contents

UNIT 2: SURVEY OF HUMAN COMPUTER INTERACTION PRACTICES

TABLE OF CONTENTS

UNIT 3: BASIC COMPONENTS OF HUMAN COMPUTER INTERACTION

Table of contents

UNIT 4: CRITICAL EVALUATION OF COMPUTER BASED TECHNOLOGY

Table of contents

MODULE 2: USER PERSPECTIVES OF HUMAN-COMPUTER INTERACTION

UNIT 1: USER ORIENTED PERSPECTIVE OF HUMAN COMPUTER INTERACTION : SOCIAL HUMAN THRUST

Table of contents

UNIT 2: USER ORIENTED PERSPECTIVE OF HUMAN COMPUTER INTERACTION : COGNITIVE HUMAN THRUST

Table of contents

UNIT 3: SYSTEM ORIENTED PERSPECTIVE OF HUMAN COMPUTER INTERACTION-IMPROVING THE USER'S TECHNOLOGICAL PERSPECTIVE

Table of contents

UNIT 4: DEVICES TECHNOLOGICAL PERSPECTIVE
INTERACTION STYLES AND DEVICES TECHNOLOGICAL PERSPECTIVE
Table of contents

MODULE 3: DESIGNS OF HUMAN-COMPUTER INTERACTION
UNIT 1: DESIGN GUIDELINES, RULES AND PRINCIPLES
Table of contents

UNIT 2: EVALUATION METHODS
Table of contents

UNIT 3: PARTICIPATORY DESIGN

Table of contents

UNIT 4: SYSTEM INTERACTIVE DESIGN PATTERNS

Table of contents

MODULE 4: DESIGN OF USER AND SYSTEM INTERFACES

UNIT 1: DESIGN OF USER INTERFACE CONCEPTS

Table of contents

UNIT 2: USER INTERFACE DESIGN PRINCIPLES AND CRITERIA/RATIONALE

Table of contents

MODULE 1: SURVEY OF HUMAN-COMPUTER INTERACTION CONCEPTS, THEORIES AND PRACTICE

UNIT 1 CONCEPTS, THEORIES AND HISTORY

Table of contents

1.0 INTRODUCTION

This unit introduces you to the basic concepts of Human-computer interaction and the theories driving it. It is meant as an overview towards appreciating the early efforts made to improve upon human computer interaction. It therefore discusses the history, and the paradigm shifts. It is meant to provide a general background for the understanding and design of Human-computer interaction.

2.0 OBJECTIVES

By the end of this unit, you should be able to

- Understand the concepts of Human-computer interaction
- Express the goals of Human-computer interaction research and study
- Explain Human-computer interaction technique
- Understand the history and paradigms of Human-computer interaction

.0 MAIN CONTENT

3.1 Definition

The following definition is given by the Association for Computing Machinery
"Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them."

Because of the interaction that occurs between users and computers at the interface of software and hardware such as between computer peripherals and large-scale mechanical systems in aircraft and power plants, human-computer interaction is the study of that interaction between people (otherwise called users) and computers. It can also be regarded as the intersection of computer science, behavioral sciences, design and several other fields of study.

3.2 Overview

Since human-computer interaction studies a human and a machine in conjunction, it draws from supporting knowledge on both the machine and the human side. On the machine side, techniques in computer graphics, operating systems, programming languages, and development environments are relevant. On the human side, communication theory, graphic and industrial design disciplines, linguistics, social sciences, cognitive psychology, and human performance are relevant. Engineering and design methods are also relevant.
The multidisciplinary nature of HCI enables people with different backgrounds contribute to its success.
HCI is also sometimes referred to as man—machine interaction (MMI) or computer—human interaction (CHI).

2.1 The goals of HCI Studies:

A basic goal of HCI study is to improve the interactions between users and computers by making computers more usable and receptive to the user's needs in the following ways:

- Methodologies and processes for designing interfaces in their related styles (i.e., given a task and a class of users, design the best possible interface within given constraints, optimizing for a desired property such as learn ability or efficiency of use)
- Methods or techniques for implementing interfaces (e.g. software toolkits and libraries; efficient algorithms)
- Techniques for evaluating and comparing interfaces
- Developing new interfaces and interaction techniques
- Developing descriptive and predictive models and theories of interaction
- Design systems that minimize the barrier between the human's cognitive model of what they want to accomplish and the computer's understanding of the user's task.

2.2 Research

Part of research in human-computer interaction involves exploring easier-to-learn or more efficient interaction techniques for common computing tasks. This includes inventing new techniques and comparing existing techniques using the scientific method as follows:

1. Designing graphical user interfaces and web interfaces.
2. Developing new design methodologies,
3. Experimenting with new hardware devices,
4. Prototyping new software systems,
5. Exploring new paradigms for interaction, and
6. Developing models and theories of interaction.

. 3 Interaction technique

An interaction technique or user interface technique is a combination of input and output consisting of hardware and software elements that provides a way for computer users to accomplish a simple task. For example, one can go back to the previously visited page on a Web browser by either clicking a button, hitting a key, performing a mouse gesture or uttering a speech command.

The computing perspective of interaction technique:
Here, an interaction technique involves one or several physical input devices, including a piece of code which interprets user input into higher-level commands, possibly producing user feedback and one or several physical output devices.
Consider for example, the process of deleting a file using a contextual menu. This first requires a mouse and a screen (input/output devices). Then, a piece of code needs to paint the contextual menu on the screen and animate the selection when the mouse moves (user feedback). The software also needs to send a command to the file system when the user clicks on the "delete" item (interpretation).

The user view of interaction technique:
Here, an interaction technique is a way to perform a simple computing task and can be described by the way of instructions or usage scenarios. For example "right-click on the file you want to delete, then click on the delete item".

The conceptual view of interaction technique:
Here, an interaction technique is an idea and a way to solve a particular user interface design problem. It does not always need to be bound to a specific input or output device. For example, menus can be controlled with many sorts of pointing devices.
Interaction techniques as conceptual ideas can be refined, extended, modified and combined. For example, pie menus are a radial variant of contextual menus. Marking menus combine pie menus with gestures. In general, a user interface can be seen as a combination of many interaction techniques, some of which are not necessarily widgets.

3.4 Interaction styles

Interaction techniques that share the same metaphor or design principles can be seen as belonging to the same interaction style. Examples are command line and direct manipulation user interfaces.
More details are provided in subsequent chapter of this guide.

3.5 Paradigms and History

Paradigms are predominant theoretical frameworks or scientific world views such as the Aristotelian, Newtonian, and Einsteinian (relativistic) paradigms in physics
Understanding HCI history is largely about understanding a series of paradigm shifts

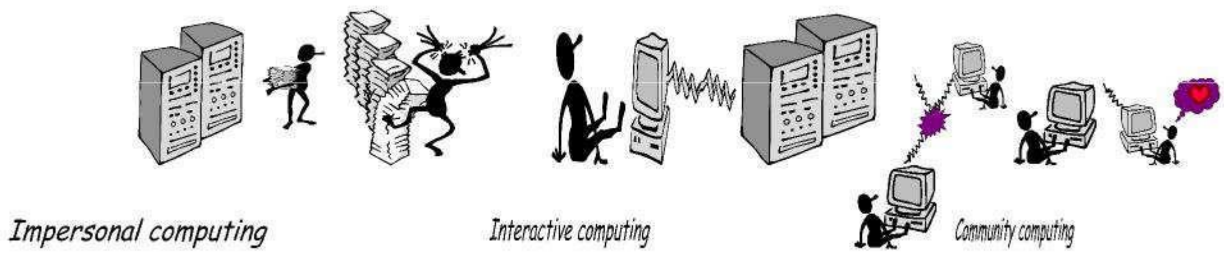
The study of paradigms is concerned about how an interactive system is developed to ensure its usability and how that usability can be demonstrated or measured.
The history of interactive system design also provides paradigms for usable designs

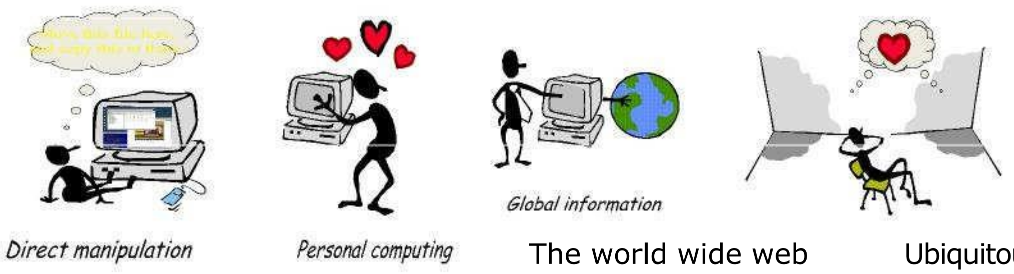
3.5.1 Paradigms of interaction

Paradigms of interaction conceptually outline the arrival of new technologies creating a new perception of the human-computer relationship.
Some of these paradigms shifts can be traced in the history of interactive technologies as follows:.
Batch processing
Timesharing
Networking
Graphical display
Microprocessor
World Wide Web (WWW)
Ubiquitous computing

The initial paradigm started with batch processing that signified impersonal computing.
The paradigm shifts commenced fron timesharing processing system that signified an interactive computing.
This was followed by another paradigm shift in networking that represented a community computing.
The graphical display was an innovation whose era indicated a paradigm shift to direct manipulation of devices.
The Micro processor innovation provided opportunity for personal computing as an example of another paradigm shift.
The World Wide Web (www) turned the world into a global village by creating environment for a global information access and transmission. This represents a significant paradigm shift in the life of educated humans all over the world.
Ubiquitous computing can be regarded as another paradigm shift as it is presently a symbiosis of physical and electronic worlds in service of everyday activities.

Pictorial representations of the shifts are illustrated below:





The world wide web   Ubiquitous Computing

Figure 1: Approximate time lines showing where work was performed on some major technologies. .5.2 The History of Paradigm Shifts

Time-sharing:
1940s and 1950s witnessed explosive technological growth in computing and in 1960s , there was the need to channel the power.
Hence J.C.R. Licklider at ARPA introduced the single computer that supported multiple users.

Video Display Units:
Video Display Units provided more suitable medium than paper and so in 1962, Sutherland introduced the Sketchpad computers for visualizing and manipulating data. So, one person's contribution drastically changed the history of computing.

Programming toolkits
Engelbart at Stanford Research Institute in 1968 augmented man's intellect by demonstrating the NLS Augment system. This became the right programming toolkit that provided the building blocks to produce complex interactive systems.

Personal computing:
The era of personal computing came on board in 1970s with the introduction of the Papert's LOGO language for simple graphics programming by children. This system became popular as it became easier to use.
The era of computing in small but powerful machines dedicated to the individual was witnessed such as was demonstrated by Kay at Xerox PARC with the Dyna-book as the ultimate personal computer.

The Window systems and the WIMP interface:
The Window systems and the WIMP interface enabled humans to pursue more than one task at a time such as in the windows used for dialogue partitioning, to "change a topic". This became a reality in 1981

with the Xerox Star as the first commercial windowing system comprising windows, icons, menus and pointers (WIMPs) as familiar interaction mechanisms.

Direct manipulation

In 1982, Shneiderman improved upon direct manipulation of objects on the computer by introducing a graphically-based interaction of visibility of objects. This provided incremental action and rapid feedback, its reversibility facility encouraged exploration and syntactic correctness of all actions. It replaced language with action.

In 1984 using the Apple Macintosh , the model-world metaphor 'What You See Is What You Get (WYSIWYG) ' became popular.

This related Language with action to confirm that actions do not always speak louder than words! The direct manipulation interface replaced underlying system of language paradigm and interface as mediator. The interface acted as the intelligent agent since programming by example is both action and language.

Hypertext

In 1945, Vannevar Bush and the Memex gave the computing world the key to success in managing explosion of information by introducing the hypertext. In mid 1960s, Ted Nelson described hypertext as non-linear browsing structure.

Within the same period, Nelson Xanadu started a project on hypermedia and multimedia; this gave bedrock for research in this area.

In the World Wide Web, the Hypertext, as originally realized, was a closed system. It comprises simple, universal protocols (e.g. HTTP) and mark-up languages (e.g. HTML) that made publishing and accessing easy. It allowed emancipation of critical mass of users that led to a complete transformation of our information economy.

Applying hypertext technology to browsers allows one to traverse a link across the world with a click of the mouse.

.5. 3 History of Basic Interactions

The Mouse: The mouse was developed at Stanford Research Laboratory in 1965 as part of the NLS project to be a cheap replacement for light-pens, which had been used at least since 1954. Many of the current uses of the mouse were demonstrated by Doug Engelbart as part of NLS in a movie created in 1968

Drawing programs: Much of the current technology was demonstrated in Sutherland's 196 3 Sketchpad system. The use of a mouse for graphics was demonstrated in NLS (1965). In 1968 Ken Pulfer and Grant Bechthold at the National Research Council of Canada built a mouse out of wood patterned after Engelbart's and used it with a key-frame animation system to draw all the frames of a movie.

Text Editing: In 1962 at the Stanford Research Lab, Engelbart proposed, and later implemented a word processor with automatic word wrap, search and replace, user-definable macros, scrolling text, and commands to move, copy, and delete characters, words, or blocks of text. Xerox PARC's Bravo was the first WYSIWYG editor-formatter developed in 1974. It was designed by Butler Lampson and Charles Simonyi who had started working on these concepts around 1970 while at Berkeley. The first commercial WYSIWYG editors were the Star, LisaWrite and then MacWrite.

Spreadsheets: The initial spreadsheet was VisiCalc which was developed by Frankston and Bricklin between 1977 and 1978 for the Apple II while they were students at MIT and the Harvard Business School. The solver was based on a dependency-directed backtracking algorithm by Sussman and Stallman at the MIT AI Lab.

Computer Aided Design (CAD): The same 196 3 IFIPS conference at which Sketchpad was presented also contained a number of CAD systems, including Doug Ross's Computer-Aided Design Project at MIT in the Electronic Systems Lab and Coons' work at MIT with SketchPad. Timothy Johnson's pioneering work on the interactive 3D CAD system Sketchpad 3 was his 196 3 MIT MS thesis.

Video Games: The first graphical video game was probably SpaceWar by Slug Russel of MIT in 1962 for the PDP-1 including the first computer joysticks. The early computer Adventure game was created by Will Crowther at BBN, and Don Woods developed this into a more sophisticated Adventure game at Stanford in 1966 Conway's game of LIFE was implemented on computers at MIT and Stanford in 1970.

Gesture Recognition: The first pen-based input device, the RAND tablet, was funded by ARPA. Sketchpad used light-pen gestures (196 3). Teitelman in 1964 developed the first trainable gesture recognizer. A very early demonstration of gesture recognition was Tom Ellis' GRAIL system on the RAND tablet in 1964,. It was quite common in light-pen-based systems to include some gesture recognition. A gesture-based text editor using proof-reading symbols was developed at CMU by Michael Coleman in 1969. Gesture recognition has been used in commercial CAD systems since the 1970s
.
Multi-Media: The FRESS project at Brown used multiple windows and integrated text and graphics in 1968. The Interactive Graphical Documents project at Brown was the first hypermedia (as opposed to hypertext) system, and used raster graphics and text, but not video between 1979 and 198 3.. The Movie Manual at the Architecture Machine Group (MIT) was one of the first to demonstrate mixed video and computer graphics in 198 3

3-D: The first 3-D system was probably Timothy Johnson's 3-D CAD system in 196 3. The "Lincoln Wand" by Larry Roberts was an ultrasonic 3D location sensing system, developed at Lincoln Labs in 1966. That system also had the first interactive 3-D hidden line elimination. An early use was for molecular modeling. Also, the military-industrial flight simulation work of between the 60's and the 70's led the way to making 3-D real-time with commercial systems from some firms.

Virtual Reality and "Augmented Reality": The original work on VR was performed by Ivan Sutherland when he was at Harvard between 1965 and 1968.

Computer Supported Cooperative Work. Doug Engelbart's 1968 demonstration of NLS included the remote participation of multiple people at various sites. Electronic mail, still the most widespread multi-user software, was enabled by the ARPAnet, which became operational in 1969, and by the Ethernet from Xerox PARC in 197 3. An early computer conferencing system was Turoff's EIES system at the New Jersey Institute of Technology in 1975.

Natural language and speech: The fundamental research for speech and natural language understanding and generation has been performed at CMU, MIT, SRI, BBN, IBM, AT&T Bell Labs and BellCore, much of it government funded.

Software Tools and Architectures

The area of user interface software tools is quite active now, and many companies are selling tools. Most of today's applications are implemented using various forms of software tools.

UIMSs and Toolkits: The first User Interface Management System (UIMS) was William Newman's Reaction Handler created at Imperial College, London between 1966 and 1967. Most of the early work was done at the university of Toronto, George Washington University and Brigham Young University. Early window managers such as Smalltalk developed in 1974 and InterLisp, both from Xerox PARC, came with a few widgets, such as popup menus and scrollbars. The Xerox Star of 1981 was the first commercial system to have a large collection of widgets. The Apple Macintosh (1984) was the first to actively promote its toolkit for use by other developers to enforce a consistent interface.

Interface Builders: These are interactive tools that allow interfaces composed of widgets such as buttons, menus and scrollbars to be placed using a mouse. The Steamer project at BBN carried out between 1979 and 1985 was probably the first object-oriented graphics system. Trillium was developed at Xerox PARC in 1981. Another early interface builder was the MenuLay system and was developed by Bill Buxton at the University of Toronto in 198 3. The Macintosh in 1984 included a "Resource Editor" which allowed widgets to be placed and edited. Jean-Marie Hullot created "SOS Interface" in Lisp for the Macintosh while working at INRIA in 1984.

Component Architectures: The idea of creating interfaces by connecting separately written components was first demonstrated in the Andrew project by Carnegie Mellon University's Information Technology Center in 198 3. It is widely popularized by Microsoft's OLE and Apple's OpenDoc architectures.

Multimodality

A mode is a human communication channel. Hence multimodality places emphasis on simultaneous use of multiple channels for input and output.

Computer Supported Cooperative Work (CSCW)

The Computer Supported Cooperative Work (CSCW) removes bias of single user with single computer system but one can not neglect the social aspects.

Electronic mail is most prominent success of Computer Supported Cooperative Work.

Agent-based Interfaces

Agent-based Interfaces are original interfaces with commands given to computer and it is language-based. It involves direct manipulation using the WIMP interface. Commands are performed on "world" representation

and it is aaction based.

The agents return to the language by instilling proactively and "intelligence" in command processor. Example is found in Avatars, a natural language processor.

Ubiquitous Computing:

One example of ubiquitous computing is the ubiquitous graphical interface used by Microsoft Windows 95, which is based on the Macintosh, which is based on work at Xerox PARC, which in turn is based on early research at the Stanford Research Laboratory (now SRI) and at the Massachusetts Institute of Technology. Virtually all software written today employs user interface toolkits and interface builders, concepts which were developed first at universities. The spectacular growth of the World-Wide Web is also a direct result of HCI research. Interface improvements more than anything else has triggered this explosive growth. Computers will communicate through high speed local networks, nationally over wide-area networks, and portably via infrared, ultrasonic, cellular, and other technologies. Data and computational services will be portably accessible from many if not most locations to which a user travels.

Sensor-based and Context-aware Interaction

Humans are good at recognizing the "context" of a situation and reacting appropriately. They are also good at automatically sensing physical phenomena (e.g., light, temperature, location, identity) becoming easier Sensors utilized the concept of senses of physical measures to interactions that behave as if made "aware" of the surroundings.

Metaphor

The LOGO's turtle dragging its tail enabled an effective teaching technique with file management on an office desktop, word processing for typing, and financial analysis using the spreadsheets

The problems with metaphors are that some tasks do not fit into a given metaphor while some can be culturally biased.

4.0 Conclusion

The study of paradigms is concerned about how an interactive system is developed to ensure its usability and how that usability can be demonstrated or measured.

The history of interactive system design also provides paradigms for usable designs

Paradigms of interaction conceptually outline the arrival of new technologies creating a new perception of the human-computer relationship. Understanding Human-computer interaction history is largely about understanding a series of paradigm shifts.

Some of these paradigms shifts can be traced in the history of interactive technologies as outlined.

5.0 Summary

Human Computer Interaction is the interaction between computer users and its interface of software and hardware.

Its study requires the knowledge of computer graphics, operating systems, programming languages, cognitive psychology, and human performance among others.

The study of HCI paradigms concerns its development while the history concerns the understanding of the paradigm shifts. Paradigms are predominant theoretical frameworks or scientific world views.

The unit has looked at the concepts and techniques of HCI, the history of paradigm shifts, the history of basic interactions and the personalities behind the innovations of HCI.

6.0 Tutor Marked Assignment

1. What do you understand by the expression Human Computer Interaction?
2. Mention any 5 scientific methods of conducting research in human-computer interaction.
3. Distinguish between an interaction technique and an interaction style.
4. Explain what you understand as the paradigms of human Computer interaction
5. Explain any 5 innovations relating to the history of paradigm shifts

7.0 Further Reading/References

- Andrew Sears and Julie A. Jacko (Eds.). (2007). Handbook for Human Computer Interaction (2nd Edition). CRC Press. ISBN 0-8058-5870-9
- Julie A. Jacko and Andrew Sears (Eds.). (200 3). Handbook for Human Computer Interaction. Mahwah: Lawrence Erlbaum & Associates. ISBN 0-8058-4468-6
- Stuart K. Card, Thomas P. Moran, Allen Newell (198 3): *The Psychology of Human—Computer Interaction*. Erlbaum, Hillsdale 198 3 ISBN 0-89859-24 3-7
- Brad A. Myers: *A brief history of human—computer interaction technology.* Interactions 5(2):44-54, 1998, ISSN 1072-5520 ACM Press. http://doi.acm.org/10.1145/2744 30.2744 36

UNIT 2: SURVEY OF HUMAN COMPUTER INTERACTION PRACTICES

TABLE OF CONTENTS

1.0 INTRODUCTION

Nature of Interactivity

Long ago going down memory lane, 'remote computer interaction' was done through *batch* processing involving punched card stacks or large data files prepared with long wait for the line printer output. And if it is not right the wait continued indefinitely ...

But now most computing is 'truly' interactive with rapid feedback and the user is in control most of the time with the thinking taken over by the computer.

A typical computer system interaction is carried out through input devices such as: the screen or monitor, keyboard, mouse or track pad.

The input devices exist in variations of desktop, laptop, mainframe computers and Personal Digital Assistants (PDAs). The devices dictate the styles of interaction that the system supports.

If we use different devices, then the interface will support a different style of interaction.

This unit sets out to remind you of the various nature of human-computer interaction that you may have normally come across.

In order to understand the nature of human—*computer* interaction, one needs to understand the computer systems.

2.0 OBJECTIVES

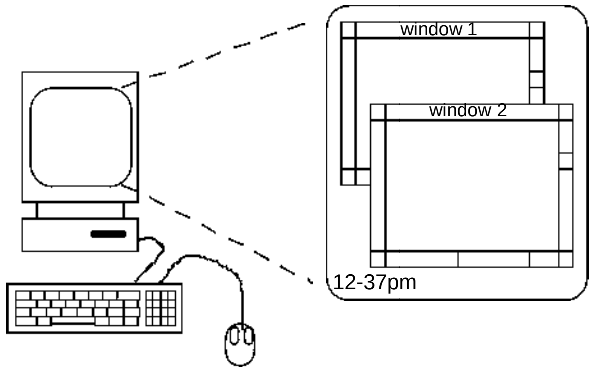
By the end of this unit, you should be able to:

- Understand handwriting and speech recognition
- Identify positioning, printing and drawing devices
- Know display devices
- Recognise physical controls and sensors
- Differentiate between printers and scanners
- Know memory and storage formats

.0 MAIN CONTENT

.1 Overview of the Computer System

The computer system is made up of various elements and each of these elements affects the interaction in the following manner:
Input devices are used for text entry and pointing.
Output devices are used for display and print of processed data on Visual Display Unit(screen) and printer on digital paper
Virtual reality affected using special interaction and display devices
Physical interaction carried out through sound, haptic, and bio-sensing devices
Paper is used for printing output and for scanning inputs.
Storage and memory utilised through accessing large capacity Random Access Memory (RAM) and permanent storage media
Processing carried out using high speed processing units and networks
See a pictorial representation below:



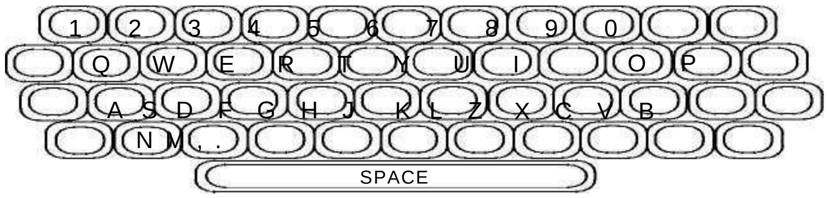
You can therefore consider the types and variations of computers that help operate your devices as follows:
In your house, these include
Personal Computers
TV, VCR, DVD, HiFi, Cable/satellite TV
Microwave, Cooker, Washing Machine
Central heating system
Security system
can you think of more variations ?

And then portable ones that can be put in your pockets
Personal Digital Assistants
Phone, Camera
Smart card and card with magnetic strip
Electronic car key with automatic opening and closing of doors
USB memory

.2 TEXT ENTRY DEVICES

Most of these common text input devices allow rapid entry of text by experienced users and usually connected by cable but can also be wireless.
The richer interaction is enabled through faster text entry devices using 'QWERTY' keyboards , chord keyboards and phone pads

QWERTY layout



Standardised layout — QWERTY keyboard with non-alphanumeric keys are placed differently because accented symbols are needed for different scripts.
An example of QWERTY arrangement is shown above:
This QWERTY arrangement is not optimal for typing because initial layout was to prevent typewriters jamming! Alternative designs however allow faster typing but large social base of QWERTY typists produces reluctance to change.
Alternative keyboard layouts introduced later, have the following characteristics:
Alphabetic:
Here the keys are arranged in alphabetic order but are however neither faster for trained typists nor for beginners too!

Dvorak:
This has common letters under dominant fingers with bias towards right hand.
Common combinations of letters alternate between hands resulting in 10-15% improvement in speed and reduction in fatigue.
Expectedly, large social base of QWERTY typists produce market pressures not to change.

Special keyboards:
These are designed to reduce fatigue for RSI and also for one handed use.

Example is the Maltron left-handed keyboard shown above.

Chord keyboards:
These have only a few keys - four or five in number. Letters are typed as combination of key presses; the key presses reflect the letter shape.
Its compact size makes it ideal for portable applications.
It has a short learning time, and it is fast once you have trained.
However, social resistance and fatigue creep in after extended use.

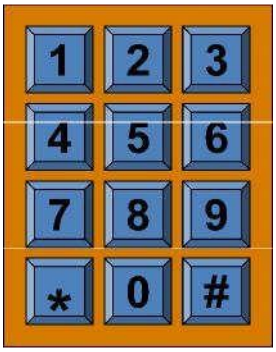
Phone pad and T9 entry:

These use numeric keys with multipl e presses as shown above and keys extracte and shown below

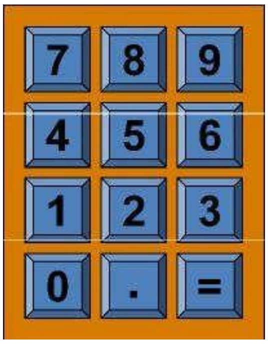
2 — a b c   6 - m n o
3- d e f   7 - p q r s
4 - g h i 8 - t u v
5 - j k l   9 - w x y z

T9 predictive entry:
This allows typing as if single key for each letter. It uses dictionary to 'guess' the r ight word

Numeric keypads
These are also developed for easier uman computer interaction since they prov ide the means of entering numbers quickly. They could be foun d in calculators, PC keyboards and telephon s.



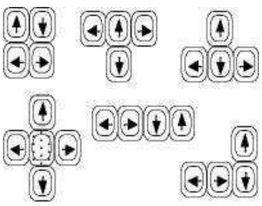
Telephone



Calculator

If you compare your phone keyboard with your calculator keyboard, you would n otice the difference as shown above.

Cursor keys



There are four keys (up, down, left, r ight) on keyboard.
The keys are very, very cheap, but sl w and provide basic motion for text-editing tasks.
There is no standardised layout, but nverted "T" that is most common with discr ete positioning controls.

. 3 Handwriting, Speech Recogni tion and other Devices

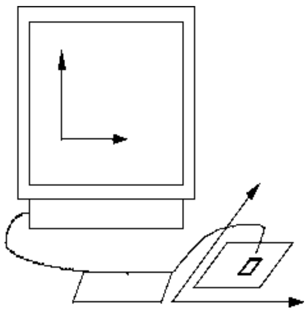
3.3.1 Handwriting Recognition.
Another initiative that improves human interaction is Handwriting recognition.
Here text can be input into the computer, using a pen and a digesting tablet thereby producing natural interaction
However, this generates the following technical problems:
Difficulties may be experienced while capturing all useful information in a natural manner and segmenting joined up writing into individual letters.
Also difficult is interpreting individual letters and coping with different styles of andwriting.
They are commonly used in PDAs (Personal Digital Assistants), and tablet computers.

3. 3.2 Speech Recognition:
This provides the advantage of leaving the keyboard on the desk, doing some other thing and talking to the computer!
The speech recognition is improving rapidly and is most successful when a single user has initial training and learns the peculiarities of limited vocabulary systems
The speech recognition system may have problems with
External noise interfering, imprecision of pronunciation, large vocabularies, and variation effects due to different speakers.

Other devices:
Other classes of human interaction devices are the positioning, pointing and drawing devices that include the mouse, touchpad, trackballs, joysticks, touch screens, eye gaze and cursors.

.4 Positioning, Pointing and Drawing Devices

The Mouse
This is a human computer interaction medium and is a handheld pointing device that is very common and easy to use. It has two characteristics, the planar movement and the buttons. Usually from Ito 3 buttons on top, used for making a selection, indicating an option, or to initiate drawing etc. The mouse is located on desktop and requires no large physical space and no arm fatigue.



Relative movement only is detectable which moves the screen cursor. The screen cursor is oriented in x and y plane while the mouse movement is in x and z plane ...
It is an *indirect* manipulation device that does not obscure screen, it is accurate and fast.
There are two methods for detecting its motion
Mechanical
- Ball on underside of mouse turns as mouse is moved rotating orthogonal potentiometers
- It can be used on almost any flat surface

Optical

– Here light emitting diode is located on underside of mouse
– It may use special grid-like pad placed on desk
– It is less susceptible to dust and dirt
– It detects fluctuating alterations in reflected light intensity to calculate relative motion in (x, z) plane
– It can be used even by foot using foot *mouse similar to* car pedals, sewing machine speed control, organ and piano pedals

Trackball and thumbwheels

A Trackball has a ball rotated inside a static housing like an upside down mouse!
Its relative motion moves the cursor. It is an indirect device and fairly accurate using separate buttons for picking. It is very fast for gaming and used in some portable and notebook computers.
Thumbwheels are used for accurate Computer Aided Design (CAD). It has two dials for X-Y cursor positioning. For fast scrolling , a single dial is used on mouse.

Joystick and keyboard nipple

Joystick has an indirect pressure of stick that equals the <u>velocity</u> of movement.
It has buttons for selection on top or on front like a trigger.
It is often used for computer games, aircraft controls and 3D navigation
A Keyboard nipple is useful for laptop computers. It has a miniature joystick in the middle of the keyboard.

Touchpad

These are small touch sensitive tablets with 'stroke' to move mouse pointer used mainly in laptop computers. It has 'acceleration' settings in form of fast stroke with lots of pixels per inch moved and initial movement to the target
Other types have slow stroke with less pixels per inch for accurate positioning.

Touch-sensitive screen

A touch-sensitive screen detects the presence of finger or stylus on the screen. It works by interrupting matrix of light beams, capacitance changes or ultrasonic reflections. It is a *direct* pointing device.

- Advantages are:
  - It is fast, and requires no specialised pointer
  - It is good for menu selection
  - It is suitable for use in hostile environment; being clean and safe from damage.
- Disadvantages:
  - Finger can mark the screen
  - It could be imprecise because the finger is a fairly blunt instrument!, hence difficult to select small regions or perform accurate drawing
  - The user lifting his arm can be tiring

Stylus and light pen

Stylus is a small pen-like pointer to draw directly on screen which may use touch sensitive surface or magnetic detection.
It is used in PDA, tablets PCs and drawing tables
Light Pen is now rarely used but uses light from screen to detect location.
Both stylus and light pen are very direct and obvious to use but they can obscure the screen.

Digitizing tablet

This is a mouse like-device with cross hairs used on special surface and it is rather like stylus.
It is very accurate and used for digitizing maps

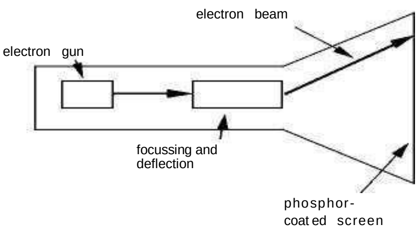
Eye gaze
This controls interface by eye gaze direction such as looking at a menu item to select it.
It uses laser beam reflected off retina at a very low power laser!
It is mainly used for evaluation and has potential for hands-free control.
Its high accuracy requires headset. The cheaper and lower accuracy devices are available under the screen like a small web cam.

Discrete positioning controls
These can be found in phones, TV controls etc.
They have cursor pads or mini-joysticks with discrete left-right and up-down movement used mainly for menu selection. See below.

3.5 Display devices
These are bitmap screens Cathode Ray Tube (CRT) and Liquid Crystal Displays (LCD).There are also large and situated displays.

Bitmap Screen displays
The screen contains vast number of coloured dots with the following resolution and colour depth:
Resolution: This is the number of pixels on screen (width x height), for example SVGA has 1024 x 768, PDA has around 240 x 400.
Density of pixels (in pixels or dots per inch - dpi) is typically between 72 and 96 dpi
Aspect ratio: This is the ratio between width and height, for example ratio 4 to 3 for most screens, ratio 16 to 9 for wide-screen TV.
Colour depth: This expresses the number of different colours for each pixel. e.g black and white or greys only, 256 from a palette, 8 bits each for red, green, and blue contains millions of colours.

Cathode Ray Tube (CRT)



Stream of electrons emitted from electron gun, focused and directed by magnetic fields, hit phosphor-coated screen which glows and used in TVs and computer monitors.

Health hazards of CRT!
X-rays are largely absorbed by the screen though not from the rear.
Ultra Violet and Intra Red radiations emanate from phosphors in the tube but at insignificant levels.
Radio frequency emissions, plus ultrasound about 16kHz, are common.
Electrostatic field leaks out through tube to the user. The Intensity is dependent on distance and humidity and this can cause rashes to the user.
Electromagnetic fields between 50Hz and 0.5MHz create induction currents in conductive materials, including the human body. Two types of effects are attributed to this: effect on visual system with high

incidence of cataracts in VDU operators, and concern over reproductive disorders such as miscarriages and birth defects.

Health hints ...
Do not sit too close to the screen
Do not use very small fonts
Do not look at the screen for long periods without a break
Do not place the screen directly in front of a bright window
Work in well-lit surroundings
Take extra care if pregnant.
Also watch your posture, ergonomics and stress while using the system.

Liquid Crystal Displays (LCD)
This is smaller, lighter, and without radiation problems.
They are found on PDAs (Personal Digital Assistants), portables and notebooks, and increasingly on desktop and even home TV.
It is also used in dedicated displays such as in digital watches, mobile phones, High Fidelity (HiFi) controls etc.
How it works ...
Top plate is transparent and polarised. Bottom plate reflects the light that passes through the top plate and crystal, and reflects back to user's eye. The voltage applied to the crystal changes polarisation and hence the colour.
The light from the display is reflected and not emitted and so causes less eye strain.

Special displays of LCD
Random Scan comprising directed-beam refresh and vector display
These draw the lines to be displayed directly without jaggies . The lines need to be constantly redrawn. They are however rarely used except in special instruments

Direct view storage tube (DVST)
- Similar to random scan but persistent => no flicker
- Can be incrementally updated but not selectively erased
- Used in analogue storage oscilloscopes

large displays
- used for meetings, lectures, etc.
- technology
  - plasma — usually wide screen
  - video walls — lots of small screens together
  - projected — RGB lights or LCD projector
    - hand/body obscures screen
    - may be solved by 2 projectors + clever software
  - back-projected
    - frosted glass + projector behind

Direct view storage tube (DVST)
Similar to random scan but persistent and has no flicker. It can be incrementally updated but not selectively erased
It is used in analogue storage oscilloscopes .
Has large displays used for meetings, lectures, etc.
The technology comprises plasma that is usually a wide screen. The video walls contain lots of small screens together and has projected RGB lights or LCD projector.

Situated displays
These are displays in 'public' places a
for a small group.
The display is only for information re
The interactive one uses stylus and t
meaning of information or interactio

3.6 Physical Controls, Sensors et
These are special displays and gauge
controls, environmental and bio-sen

Dedicated displays like physical cont
Analogue representations that includ
Digital displays as identified in small
Head-up displays found in aircraft co

Sounds
Sounds are beeps, bongs, clonks, wh
confirmation of actions e.g. key click

Touch, feel and smell devices otherw
Here touch and feeling are importan
Also important in simulation that req
However, current technology in text

BMW iDrive
This is used for controlling menus wi
options by feel.
It uses haptic technology from Imme

Physical Controls:
These are specialist controls that exi
See examples below.

Environment and bio-sensing
These are sensors all around us such

Car courtesy light — small switch on door
Ultrasound detectors — as in security and washbasins.
RFID security tags in shops
Bio-sensors are you for personal health as applicable in the use of iris scanners, measurement of body temperature, heart rate, galvanic skin response and blink rate.

3.7 Print Technology:
The following elements of interaction are provided by the print technology
Fonts, Page description, What You See Is What You Get (WYSIWYG), Scanning, Optical Character Reading (OCR) etc

Printing
This is an image made from small dots. It allows any character set or graphic to be printed.
Critical features are resolution expressed in size, spacing of the dots in dots per inch (dpi).
Speed is usually measured in pages per minute.
Types of dot-based printers
i. Dot-matrix printers: These use inked ribbon (like a typewriter). They have line of pins that can strike the ribbon hence dotting the paper.
Typical resolution is between 80 to 120 dots per inch (dpi)
ii. Ink-jet and bubble-jet printers
These work by sending tiny blobs of ink from print head to paper. Resolution is typically 300 dpi or more.
iii. Laser printers
These are like photocopiers: Here dots of electrostatic charge are deposited on drum, which picks up toner (black powder form of ink) rolled onto paper which is then fixed with heat.
Typically 600 dpi or more.

Printing (aspect of human computer interaction) in the workplace
Shop tills:
Dot matrix printer uses same print head for several paper rolls; may also print cheques
Thermal printers use special heat-sensitive paper, paper heated by pins makes a dot.
Though of poor quality, printing is simple and maintenance is low.
They are used in some fax machines.

Fonts
Font is a particular style of text according to some examples given below;
Courier font, Helvetica font, Palatino font, Times Roman font, IIIITICIIIMITI (special symbol)
Size of a font is measured in points (1 point is about 1/72"), about related to its height.

Pitch
Fixed-pitch : In this case, every character has the same width,e.g. Courier
Variable-pitched — some characters are wider e.g. Times Roman — compare the 'i' and the "m"
Sans-Serif or Serif :
Sans-serif contain square-ended strokes such as in in Helvetica

Serif These are characters with splayed ends such as in in Times Roman or Palatino

Readability of text
Lowercase: easy to read shape of words
UPPERCASE: better for individual letters and non-words e.g. flight numbers: BA79 3 vs. ba79 3
Serif fonts: helps your eye on long lines of printed text but sans serif is often better on screen

Page Description Languages
Useful when pages become very complex with different fonts, bitmaps, lines, digitised photos, etc.
Description languages can convert all into a bitmap and send to the printer.
Using a page description language sends a *description* of the page for example, instructions for curves, lines, text in different styles, etc. can be sent.
It is like a programming language for printing!
PostScript is the most common form of Page description language

3.8 Scanners
These accept paper and convert it into a bitmap
Two sorts of scanner exist:
Flat-bed: Here paper is placed on a glass plate with whole page converted into bitmap
Hand-held: Here scanner is passed over paper, digitising strip typically 3-4" wide
Shines light at paper and note intensity of reflection with colour or greyscale
Typical resolutions are from 600-2400 dpi
Scanners are used in desktop publishing for incorporating photographs and other images.
They are used in document storage and retrieval systems thereby doing away with paper storage.
Special scanners exist for slides and photographic negatives

Optical Character Recognition (OCR):
Optical character recognition (OCR) converts bitmap back into text with different fonts creating problems for simple "template matching" algorithms.
However, more complex systems segment text, decompose it into lines and arcs, and decipher characters that way.
Page formatting is done on columns, pictures, headers and footers.

Paper-based interaction
Paper is usually regarded as *output* only but can be *input* too in OCR, scanning, etc. operations.
Xerox Paper Works is a paper based interaction that involve glyphs containing small patterns of /WW//WWW
Used to identify forms etc. and also used with scanner and fax to control applications
More recently, papers are micro printed - like watermarks. Watermarks identify *which* sheet and *where* you are.
Special 'pen' can read locations to know where they are writing.

3.9 Memory Interaction
Exists in form of short term and long term
One needs to have knowledge of the characteristics of a particular memory for an effective and valuable interaction. Such include the speed, capacity, the compression formats and mode of access.
Short-term Memory
Short-term Memory otherwise called Random access memory (RAM) are made of silicon chips. Most RAMs have 100 nano-second access time and are usually volatile, losing information if power is turned off.
Data transfer rate is around 100 Mbytes/sec
Some *non-volatile RAMs(ROMs) are* used to store basic set-up of information
Typical desktop computers have between 64 to 256 Mbytes of RAM.
Long-term Memory
These include magnetic disks comprising:
i. Floppy disks that can store up to 1.4 Mbytes and more; hard disks that can store between 40 Gbytes to hundreds of Gbytes
Access time approximate to 10 micro seconds while transfer rate is around 100kbytes per second.

ii. Optical disks that use lasers to read and sometimes write. They are more robust that magnetic media. They include CD-ROMs that have same technology as home audio, with storage capacity approximating 600 Gbytes and DVDs used for AV applications and for very large files.
iii. PDAs (Personal Digital Assistants) that often use RAM for their main memory.
iv. Flash-Memories used in PDAs, cameras etc.. They are silicon based but persistent. Flash memories are Plug-in USB devices used for data transfer.

Virtual memory

Problems calling for the use of virtual memory include running lots of programs. Each program could be very large but with insufficient RAM size to run it, hence the need for the use of Virtual memory.
The solution provided by Virtual memory is to store some programs temporarily on disk thereby making RAM appear bigger. But the program on disk that needs to run again and copied from disk to RAM slows down processing.

Compression

This is the reduction in the amount of storage required. The two types that could be identified are:

Lossless compression:

i. Here, the exact text or image is recovered e.g. as in GIF, ZIP formats.
ii. By looking for commonalities in text as demonstrated below:
iii. If text= AAAAAAAAAABBBBBCCCCCCCC then there are 10 As,5 Bs, and 8 Cs and is written as 10A5B8C
iv. Video: Here successive frames are compared and changes are stored.

Lossy compression:

This recovers something like the original — e.g. as in JPEG and MP 3 formats.

Exploited perception:

In JPEG, perception is exploited by losing rapid changes and some colour while in MP 3, perception is exploited by
reducing accuracy of drowned out notes

3.10 Storage formats

Text formats:

ASCII : This is a 7-bit binary code that represents each letter and character
UTF-8 : This is an 8-bit encoding of 16 bit character set
RTF (rich text format): This is a text plus formatting and layout information
SGML (standardized generalised mark-up language): These are documents regarded as structured objects
XML (extended mark-up language): This is a simpler version of SGML for web applications

Media formats:

Many storage formats exist for images : Such formats include PostScript, GIFF, JPEG, TIFF, PICT, etc. in addition to different compression techniques that reduce their storage requirements.
For Audio/Video, there are lots of formats as well. Such include QuickTime, MPEG, WAV, etc.. Compression is even more important here to optimise available storage space.
For network delivery of data, 'streaming' formats are also available.

Methods of access:

For any of the data type above, a large information store takes long time to search. Therefore, an index storage technique is used. Whatever is indexed is what can be accessed. Simple index needs exact match. Accessing without structure involves free text indexing all the words in a document hence requires lots of space!!

3.11 Processing and Networks Interactions

There is finite speed applicable to processing which is also governed by Moore's law.
There are also limits of interaction both in single user and networked computing.

Finite processing speed

Designers tend to assume fast processors and therefore make interfaces more and more complicated. But problems do occur, because processing cannot keep up with all the tasks it needs to do for the following reasons:

i. Cursor overshooting because system has buffered key presses.

ii. Icon wars Here, the user clicks on an icon and nothing happens, clicks on another, then system responds and windows fly everywhere.

iii. Also problems do occur if system is too fast. For example, help screens may scroll through text much too rapidly to be read.

Moore's law

The Moore's law observes that computers get faster and faster!

In 1965, Gordon Moore, co-founder of Intel, noticed a pattern that processor speed doubles every 18 months e.g

PC in1987, speed was 1.5 Mhz, up till 2002 when the speed became 1.5 GHz.

Similar pattern also occurs for memory and storage. But this doubles every 12 months!! e.g.

Maximum Hard disk requirement in 1991 was 20 Mega bytes but rose in 2002 to 30 Giga bytes

The myth of the infinitely fast machine:

The implicit assumption here is that there are no delays in processing on an infinitely fast machine.

This takes us to the question: What is a good design for real machines?

Networked computing

Networks allow access to large memory and processing, access to other people (such as in groupware and email) and other shared resources especially the web

Issues relating to interaction in network computing are network delays which cause slow feedback, and unexpected processing delay as a result of many people updating data simultaneously.

Lastly is the unpredictability nature of networks.

The internet as an example of a network, (international network of computers)

Short History:

In 1969: DARPANET US Department of Defence had 4 sites. In 1971, the sites increased to 2 3. In 1984, it became 1,000 and in 1989, it increased to 10,000 .

Common language protocols used are:

i. TCP — Transmission Control protocol: This operates at lower level on packets (like letters) between machines.

ii. IP — Internet Protocol: This provides a reliable channel (like phone call) between programs on machines.

Email and HTTP (Hypertext Transmission Protocol) build on top of these.

4.0 Conclusion

In spite of all the discussions about the various interactions above, some limitations are placed on interactive performance as follows:

Computation bound: Computation may take a long time, causing frustration for the user.

Storage channel bound: Bottlenecks may occur in the transference of data from disk to memory.

Graphics bound: Here common bottleneck is that updating displays requires a lot of effort, effort sometimes helped by adding a graphics co-processor optimised to take on the burden

Network capacity: Though many computers are networked with sharable resources and files and access to printers etc. yet still have interactive performance reduced by slow network speed.

5.0 Summary

The knowledge of the computer systems and the types of user interface devices such as the text entry, hand writing and speech recognition, pointing and drawing, display and storage together with the kind of network facilities have their individual impacts on human computer interactions.

6.0 Tutor Marked Assignment

1. What is the reason for the difference in the arrangement of Numeric keypads existing between your phone keyboard and the calculator keyboard
2. What are the specific difficulties experienced in handwriting recognition as a human interaction medium
3. Mention 2 advantages and 2 disadvantages of the touch sensitive screen interface
4. Explain the characteristics of the eye gaze interaction.
5. Mention 5 health hazards that may likely exist as a result of interacting with the CRT
6. (a) What do you know of haptic devices and biosensors?
   (b) Describe the function and value of Page Description language
7. What is Virtual memory and its benefit in Computer processing?

7.0 Further Readings / References

- Card, S.K., "Pioneers and Settlers: Methods Used in Successful User Interface Design," in *Human-Computer Interface Design: Success Stories, Emerging Methods, and Real-World Context*, M. Rudisill, *et al.*, Editors. 1996, Morgan Kaufmann Publishers: San Francisco. pp. 122-169.
- Baecker, R., *et al.*, "A Historical and Intellectual Perspective," in *Readings in Human-Computer Interaction: Toward the Year 2000, Second Edition*, R. Baecker, *et al.*, Editors. 1995, Morgan Kaufmann Publishers, Inc.: San Francisco. pp. 35-47.
- Brooks, F. "The Computer "Scientist" as Toolsmith--Studies in Interactive Computer Graphics," in *IFIP Conference Proceedings.* 1977. pp. 625-6 34.
- Buxton, W., *et al.* "Towards a Comprehensive User Interface Management System," in *Proceedings S IGGRAPH'83: Computer Graphics.* 198 3. Detroit, Mich. 17. pp. 35-42.
- Engelbart, D. and English, W., "A Research Center for Augmenting Human Intellect." *Reprinted in ACM S IGGRAPH Video Review, 1994.*, 1968. 106
- Goldberg, A., ed. *A History of Personal Workstations*. 1988, Addison-Wesley Publishing Company: New York, NY. 5 37.

UNIT 3: BASIC COMPONENTS OF HUMAN COMPUTER INTERACTION

Table of contents

1.0 Introduction

The components of Human Computer Interaction comprise the Interaction models that concern translation between the user and the computer system, Ergonomics that describe the physical characteristics of interaction, the Interaction styles that express the nature of user and system dialog and finally the context of the social, organizational and the motivational aspect of interaction.
This unit briefly describes each of these components with a desire of giving an overview of the general requirements for the design of human computer interaction systems.

2.0 Objectives

By the end of this unit, you should be able to:

- Understand the Interaction models of interface between the user and system
- Describe the cycle of the execution and evaluation loop
- Explain the concepts of ergonomics
- Know the common interaction styles
- Understand the context of the social, organizational and the motivational aspects of interaction
- The usefulness of the WIMP Interface

.0 Main Content

.1 The Interaction Models

The interaction models comprise
The terms of Interaction,
The Donald Norman Model
The Interaction framework

3.1.1 Terms of interaction

Domain: This is the area of work under study e.g. a graphic design
Goal : This is what you want to achieve e.g. to create a solid red triangle
Task : Concerns how you go about doing it, ultimately in terms of operations or actions e.g select the fill tool, click over the triangle

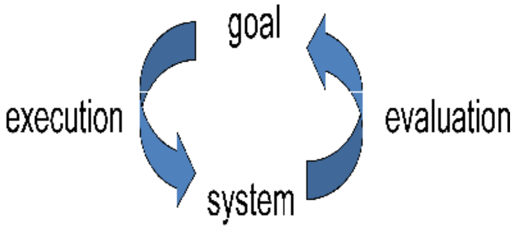
3.1.2 Donald Norman's model:

These are in seven stages as follow:

- The user establishes the goal
- The user formulates intention
- The user specifies actions at interface
- The user executes the action
- The user perceives the system state
- The user interprets the system state
- The user evaluates the system state with respect to goal

Norman's model concentrates on th user's view of the

interface Execution and evaluation loop



Interpretation:
Goal:
The user establishes the goal

Execution:
The user formulates intention
The user specifies actions at interfac e
The user executes the action

Evaluation:
The user perceives the system state
The user interprets the system state
The user evaluates the system state with respect to goal

Donald Norman's model
Norman's model can be applied thro ugh:
Gulf of Execution that evaluates the user's formulation of actions where actions re allowed by the system
Gulf of Evaluation where the user's expectation of changed system state represe nt actual presentation of this state

Interaction could harbour some human errors which may be slips and mistakes.
Slips may include lack of understanding the system and goal, incorrect formulatio n of action, incorrect action and mistake of not even having the right goal!
To fix slips, better interface design sh ould be carried out while to avoid mistakes, one should better understand the system

To avoid some of the Human errors,Abowd and Beale framework is adopted. Ab owd and Beale framework is an extension of Norman model an it has 4 parts namely: i. the user, ii. the inp ut, iii. the system, and iv. the output while each framework has its own unique language.
If interaction is the translation between languages, and if there are problems in i teraction, then there would be problems in translation

Using Abowd & Beale's model
The user intentions could be translated into actions at the interface, translated in to alterations of state, reflected in the output displaysystem or interpreted by the user himself.
The general framework for understanding interaction are that interaction is not restricted to electronic computer systems alone, all major c mponents involved in interaction should be identified. The comparative assessment of systems should be allowed. The framework also considers an abstraction.

3.2 Ergonomics

This considers both the physical aspects of interfaces and the industrial interfaces.
Ergonomics is the study of the physical characteristics of interaction. It is known as human factors.
Ergonomics is good at defining standards and guidelines for constraining the way we design certain aspects of systems
Examples of Ergonomics include:
Arrangement of controls and displays such as the controls grouped according to function, frequency and sequence of use.
Surrounding environment such as the seating arrangements adaptable to cope with all sizes of user, health issues such as the physical position, environmental conditions (temperature, humidity), lighting, and noise.

Use of colour such as the use of red for warning, green for okay, and awareness of colour-blindness etc. The user interacts *with* real world *through* interface issues, feedback and delays.

. 3 Common Interaction styles

Two major classes of interaction styles will be considered, they are:-
Dialogue Style of Interaction between computer and user
Distinct styles of interaction
Both are expressed in the following common forms of interfaces:

- Command line interface
- Menus
- Natural language
- Question and answer, and query dialogue
- Form-fills and spreadsheets
- WIMP
- Point and click
- Three—dimensional interfaces

3. 3.1 Command line interface

This is the way of expressing instructions to the computer directly through the function keys, single characters, short abbreviations, whole words, or a combination suitable for repetitive tasks.
The interface is better designed for expert users than novices because it offers direct access to system functionality. However, the command names and abbreviations used should be meaningful!
A typical example is the Unix system command line interface.

3. 3.2 Menus

Menus is a set of options displayed on the screen. The Menu Options are visible, it has a less recall characteristic that make it easier to use.
The visible options rely on recognition so the names should be meaningful. The selection is done through numbers, letters, arrow keys, mouse and/or combination of any of them e.g. mouse plus accelerators .Often, the options are hierarchically grouped. But sensible grouping is needed.
Menus are restricted form of full WIMP system .

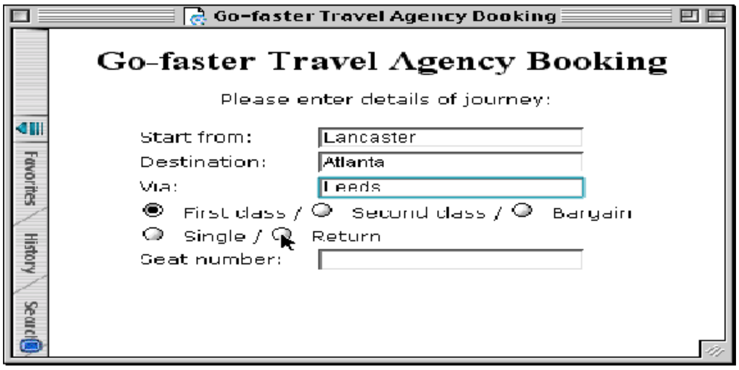
3. 3. 3 Natural language

This is the language familiar to the user. It may be in form of speech recognition or a typed natural language.

Problems with in this kind of interaction are that the language may be vague, ambiguous, and hard to be recognised.
Design solutions to language interface problems are for the user to try to understand a subset and pick on key words .

3. 3.4 Query interfaces
These comprise question and answer interfaces in which the user is led through interaction via series of questions. Though with restricted functionality, this kind of interface is suitable for novice users. It is often used in information systems.
Query languages (e.g. SQL)
This is used to retrieve information from database. It requires understanding of the database structure and language syntax, hence requires some expertise

3. 3.5 Form-fills and Spreadsheets
Form-fills are primarily designed for data entry or data retrieval. It is a screen like paper form to which data is put in relevant place.
It requires a good design and obvious correction facilities.
See illustration below



Example of a form-fill

Spreadsheets
Spreadsheets are sophisticated variations of form-filling in which grid of cells contain a value or a formula. The formula can involve values of other cells e.g. sum of all cells in this column.
The user can enter and alter data in spreadsheet to maintain consistency.
The first spreadsheet introduced was VISICALC, followed by Lotus 1-2- 3. Micro Soft Excel is the most common today

3. 3.6 WIMP Interface
This interface comprises Windows, Icons, Menus, and Pointers or Windows, Icons, Mice, and Pull-down menus!
The interface is the default style for majority of interactive computer systems, especially PCs and desktop machines.
Elements of the WIMP interface
The elements include windows, icons, menus, and pointers. In some other cases they may be buttons, toolbars, palettes, and dialog boxes.
Understanding the concept of 'Look and feel'
WIMP systems have the same elements: as windows, icons., menus, pointers, buttons, etc. but have different window systems that *behave* differently . For example, Macintosh Operating System (MacOS) compared with Windows menus.
The combination of the appearance and the behaviour is the 'look and feel'

Windows

Windows are areas of the screen that behave as if they were independent. They can contain text or graphics and can be moved or resized.

They can overlap and obscure each other, or can be laid out next to one another (tiled)

Icons

Icons are small pictures or images that represent some object in the interface. They appear often as windows or as actions.

Windows can be 'iconised' that is closed down. They are small representations that fit many accessible windows.

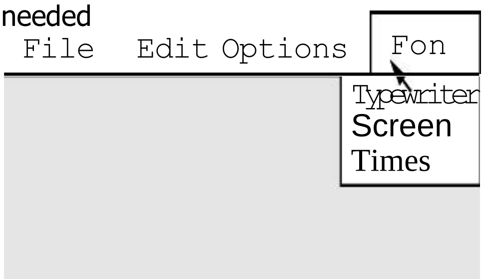
Icons can be many and various. They can be highly stylized with realistic representations.

Menus

These are choice of operations or services offered on the screen

The required option is selected with the pointer. However, this takes a lot of screen space

This problem is partly solved when a pop-up menu appears when needed



Kinds of Menus

Menu Bar at top of screen (normally), menu drags down

i Pull-down menu - mouse hold and drag down me

ii Drop-down menu - mouse click reveals men

i. Fall-down menus - mouse just moves over bar!

Contextual menu appears where you are

Pop-up menus take actions for selected object

Pie menus are arranged in a circle such that it is easier to select item over larger target area. Selection is also quicker because it can move same distance to any option. Pie menus are not widely used!

Cascading menus

This has a hierarchical menu structure in which a menu selection opens new menu and so in ad infinitum

Keyboard accelerators

This comprises key combinations with same effect as menu item.

They operate in two modes

– active when menu open — usually first letter and

– active when menu closed — usually Ctrl + letter

Menus design issues

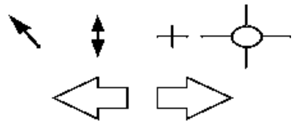
In order to design an effective menu, the following issues should be considered:

- which kind to use
- what to include in menus at all
- words to use (in action or description)
- how to group items
- choice of keyboard accelerators

Palettes and tear-off menus

- Palettes are little windows of actions shown or hidden via menu option in available shapes in drawing package
- In tear-off and pin-up menus, menu 'tears off' to become palette

Pointers
Pointers are important WIMP style components that point on and select. They are activated by the use of mouse, track pad, joystick, trackball, cursor keys or keyboard shortcuts. They are in wide variety of graphical images. See examples below.

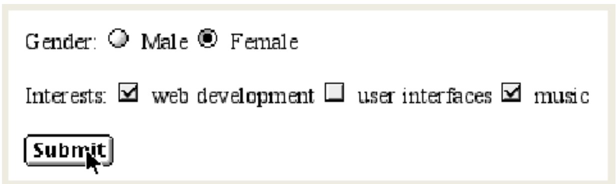


Point and click interfaces
Point and click interfaces are used in multimedia, web browsers, and hypertext. You just click something such as icons, text links or location on map. It requires minimal typing.

Scrollbars
Scrollbars allow the user to move the contents of the window up and down or from side to side.

Title bars
Title bars describe the name of the window

Buttons
This is an individual and isolated region within a display that can be selected to invoke an action
The Special kinds that exist are
The radio buttons with a set of mutually exclusive choices and the check boxes with a set of non-exclusive choices.



Toolbars
These are long lines of icons with fast access to common actions and are often customizable:
You can choose *which* toolbars to see and choose *what* options are on it

Dialogue boxes
These are information windows that pop up to inform of an important event or requested information, for example when saving a file, a dialogue box is displayed to allow the user to specify the filename and location. Once the file is saved, the box disappears.

The interactivity of dialogue boxes
They are easy to focus on look and feel.

Other types of interaction styles are speech—driven interfaces
The development of this kind of interface is yet to be perfect and accurate; though it is rapidly improving. Example of speech driven interface dialogue on an airline reservation:

reliable "yes" and "no"?
+ System reflects back its understanding
"you want a ticket from New York to Boston?"

3. 3.7 Three dimensional interfaces
These are virtual reality 'ordinary' window systems highlighting visual affordance. The indiscriminate use can however be confusing!
There are also three dimensional (3D) workspaces used for extra virtual space with light and occlusion that give deep distance effects.
For typical computer displays, three-dimensional images are projected on them in two dimensions. Three-dimensional graphics are currently mostly used in computer games, art and computer-aided design (CAD). There have been several attempts at making three-dimensional desktop environments likeSun's Project Looking Glass. A three-dimensional computing environment could be used for collaborative work. For example, scientists could study three-dimensional models of molecules in a virtual reality environment, or engineers could work on assembling a three-dimensional model of an airplane.] The Technologies The use of three-dimensional graphics has become increasingly common in mainstream operating systems, but mainly been confined to creating attractive interfaces—eye candy—rather than for functional purposes only possible using three dimensions. For example, user switching is representedby rotating a cube whose faces are each user's workspace, and window management is represented in the form of Exposé on Mac OS X. In both cases, the operating system transforms windows on-the-fly while continuing to update the content of those windows.
Interfaces for the X Window System have also implemented advanced three-dimensional user interfaces through compositing window managers such as Beryl and Compiz using the AIGLX or XGL architectures, allowing for the usage of OpenGL to animate the user's interactions with the desktop.
Another branch in the three-dimensional desktop environment is the three-dimensional GUIs that take the desktop metaphor a step further, like the BumpTop, where a user can manipulate documents and windows as if they were "real world" documents, with realistic movement and physics.
The Zooming User Interface (ZUI) is a related technology that promises to deliver the representation benefits of 3D environments without their usability drawbacks of orientation problems and hidden objects. It is a logical advancement on the GUI, blending some three-dimensional movement with two-dimensional or "2.5D" vector objects.

3.4 Context: Social and Organisational
These issues and concerns involve all possible interactions between a user and a system during its lifecycle, including the development stage, use in context, and the impact of such use on individuals, organizations, society, and future systems development.

Context Analysis
Context analysis includes understanding the technical, environmental and social settings where the information systems will be used. It examines whether and how the interaction between physical and social environment and the physiological and psychological characteristics of the user would impact users interacting with the system.
There are four aspects in Context Analysis: physical context, technical context, organizational context, and social and cultural context. Overall, context analysis can provide ideas for design factors such as metaphor creation, selection and patterns of communications between users and the system.

Physical context: This considers where the tasks carried out, what entities and resources are implicated in task operation, What physical structures and entities are necessary to understand observed task action. For example, an ATM machine can be used in a mall outside a bank office, or in a night club. These environments provide different levels of lighting, crowdedness, and noisiness. Thus legibility of the screen, use of audible devices
for input or output, or even the size of the working space to prevent people nearly to see the screen could be designed differently.

Technical context: This considers the technology infrastructure, platforms, hardware and system software, wired or wireless network connection? For example, an E-commerce website may be designed to allow access only to people with certain browser versions. The website may also be designed to allow small screen devices such as PDA or mobile phone to access.

Organizational context: Organizational context may play different roles in internal and external situations. For an organizational information system to be used by the organization's own employees, organizational context analysis answers questions such as:

- What is the larger system where this information system is embedded?,
- What are the interactions with other entities in the organization?
- What are the organizational policies or practice that may affect individual's attitude and behavior towards using the system?

For example, assuming that Lotus Note is used by an organization as a communication and collaboration tool, management may depend on using the tool to set up meetings by checking employees' calendars on mutually available time slots. The effectiveness of setting up meetings depends on whether employees use the tool, and how they use it. The whether and how questions can be enforced by organizational policies.

Social and cultural context: What are the social or cultural factors that may affect user attitudes and eventual use of the information system? In an E-Commerce website example, the website can be accessed from all over the world. It thus is a design consideration that the website allows access by people with any language and cultural background that can provide credit cards with the foreign currency exchange, or it is only accessible to people who speak certain languages and are from certain cultures.

Interactions are also affected by other social and organizational context as follow:

- By other people: A desire to impress, competition among stakeholders, and fear of failure from management
- Motivation from management as against fear, allegiance, ambition, self-satisfaction that exist among employees
- Existing inadequate systems that may cause frustration and lack of motivation

The organizational, social and cultural context in which humans interact with IT is largely the result of the broad adoption of IT by organizations and society to support organizational functions and goals and to enhance society's development. For example, organizational efficiency may be expected due to redesign of workflows among critical business units that is affected by the implemented IT; satisfaction and retention of customers/clients are anticipated due to accurate and fast information gathering and presentations, to name a few. Some of the organizational or societal impacts may not be tangible or directly attributed to HCI considerations. This assertion is in line with the issues of determining IT values in organizations and societies.
While each of these HCI concerns may have its own importance in different situations in relation to human motivation, it would be helpful for designers to see an overview picture of the potential HCI concerns and goals. The purpose of this picture is not to force every IT to be compliant with all the HCI concerns, but to provide an overall framework so that designers can use it as a roadmap and to apply it according to different situations.

4.0 Conclusion

The knowledge of the basic components of human computer interaction aids in giving direction, focus and human considerations pertaining to interactive design.

5.0 Summary

There are variations of interaction models of the interface established between the user and the computer system. These models are presented as a loop in the execution and evaluation of an interactive design. These interaction models together with the human ergonomics, the interaction styles, and the social and organizational contexts are basic components of human computer interaction.

6.0 Tutor Marked Assignment

1. Explain the three terms of Interaction
2. Mention any 4 of the 7 stages of the Donald Normans model and briefly describe each of the stages.
3. What do you understand as the execution and evaluation loop and how is the loop useful in the user's participation in the design of interactive systems
4. What are slips and mistakes in human Computer interaction and how do you avoid such slips and mistakes before they occur?
5. Briefly express your understanding of the term Ergonomics
6. Describe any four common interaction styles
7. What are Cascading menus and Keyboard accelerators?
8. What are the four constituent aspects of Context analysis? How are they of benefits to the user interacting with the computer system?

7.0 Further Readings / References

. Licklider, J.C.R. and Taylor, R.W., "The computer as Communication Device." *Sci. Tech.*, 1968. April: pp. 21- 31.

. Linton, M.A., Vlissides, J.M., and Calder, P.R., "Composing user interfaces with InterViews." *IEEE Computer*, 1989. 22(2): pp. 8-22.

. Meyrowitz, N. and Van Dam, A., "Interactive Editing Systems: Part 1 and 2." *ACM Computing Surveys*, 1982. 14(3): pp. 321- 352.

. Myers, B.A., "The User Interface for Sapphire." *IEEE Computer Graphics and Applications*, 1984. 4(12): pp. 1 3-2 3.

. Myers, B.A., "A Taxonomy of User Interfaces for Window Managers." *IEEE Computer Graphics and Applications*, 1988. 8(5): pp. 65-84.

UNIT 4: CRITICAL EVALUATION OF COMPUTER BASED TECHNOLOGY

Table of contents

5.0 Summary
6.0 Tutor Marked Assignment
7.0 Further Reading/References

1.0 Introduction

This unit briefly describes some uncommon technologies associated with human computer interaction. These are innovations that improve upon the user interface, particularly those innovations benefiting the disabled. Technologies such as the phonetic typewriter, the ear cons, the auditory icons, the recognition and gesture devices for the disabled and the elderly are described.

2.0 Objectives

By the end of this unit, you should be able to:

- Describe multi-modal, multi-media and multi-sensory systems
- Appreciate the speech and the Phonetic typewriter interfaces
- Understand the Ear cons and Auditory Icons as important components of multi-modal systems
- Know that Recognition and Gestures Devices are essential for the Elderly and Disabled

.0 Main Content

3.1 Multi-Sensory Systems

Here, more than one sensory channel are involved in interaction as in sounds, text, hypertext, animation, video, gestures and vision.
They are used in a range of applications particularly for users with special needs and virtual reality.
The components of Multi-Sensory systems are: Speech, Non-speech sounds, Handwriting, together with their applications and principles.

Usable Senses
The five senses: sight, sound, touch, taste and smell are used by us every day and each is important on its own Together, they provide a fuller interaction with the natural world
We can ideally utilise the Computers to use all the available senses but this becomes practically impossible because computers rarely offer such a rich interaction.
We can use the sight, sound, and sometimes the touch senses, but we cannot yet use the taste and smell.

3.2 Multi-modal and Multi-media systems:

Multi-modal systems
These use more than one sense (or mode) of interaction as in visual and aural senses. For example, a text processor may speak the words as well as echo them to the screen
Multi-media systems
These use a number of different media to communicate information. For example, a computer-based teaching system may use video, animation, text and still images; different media all using the visual mode of interaction. These may also use sounds, both speech and non-speech. Of course two or more media now use different modes.

3.2.1 Speech

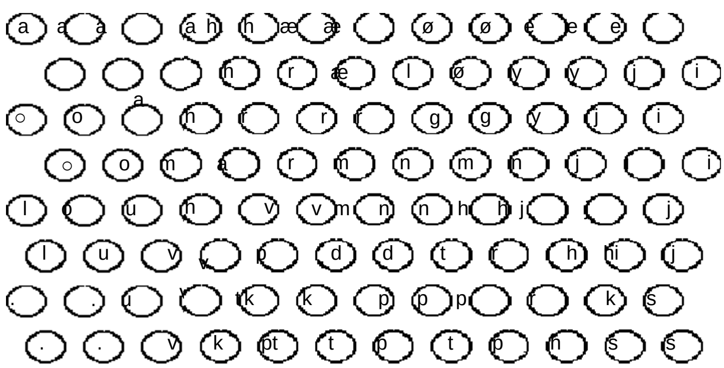
Human beings have a great and natural mastery of speech which makes it difficult to appreciate the complexities but it is an easy medium for communication
Simple terminologies used to describe speech:
The structure of speech is called phonemes and there are 40 of them. The phonemes as basic atomic units that sound slightly different depending on the context they are in, the larger units of phonemes are the

Allophones. Allophones are the sounds in the language between 120 and 1 30 and are formed into morphemes. The morphemes are the smallest units of language that have meaning.
Prosody is the alteration in tone and quality. They are also variations in emphasis, stress, pauses and pitch. They impart more meaning to sentences.
Co-articulation is the effect of context on the sound. It transforms the phonemes into allophones.
Syntax is the term used for the structure of sentences while semantics is the collective term used for the meaning of sentences.

3.2.2 Problems in Speech Recognition.
Different people speak differently because accent, intonation, stress, idiom, volume, etc. differ. The syntax of semantically similar sentences may also vary while background noises can interfere.
People often "ummm....." and "errr……" but words are not enough - semantics are also needed. It requires intelligence to understand a sentence because context of the utterance often has to be known as well as information about the subject and speaker. For example, even if "Errr ….. I, um, don't like this" is recognised, it is a fairly useless piece of information on its own.

3.2. 3 Speech Related Human-Interaction Technologies.
The Phonetic Typewriter
This is developed for Finnish - a phonetic language. This machine trained on one speaker, will generalise the training to others.
A neural network is trained to cluster together similar sounds, which are then labelled with the corresponding character.
When recognising speech, the sounds uttered are allocated to the closest corresponding output, and the character for that output is printed.
It requires large dictionary of minor variations to correct general mechanism.

The Phonetic Typewriter



Usefulness of Speech Recognition:
It is useful for a single user or in a situation in which limited vocabulary systems exist, for example in a computer dictation.
In the public and open places of limited vocabulary systems, it can work satisfactorily e.g. in some voice activated telephone systems.
For the general user with wide vocabulary systems, problems do occur.
Its great potential value however manifests when users hands are already occupied as in driving or during manufacturing particularly for users with physical disabilities.
Another advantage is its lightweight and its use as a mobile device.

Speech Synthesis

This is a generation of speech. It is useful because of its natural and familiar way of receiving information. It is successful in certain constrained applications when the user has few alternatives and is particularly motivated to overcome problems.

However, it has its own problems similar to speech recognition particularly in prosody. Additional problems can arise from intrusion calling the need for headphones particularly due to noise in the workplace .Its transient nature is a problem when it becomes harder to review and browse.

Few examples occur in screen readers that read the textual display to the user e.g. as utilised by visually impaired people. Also in warning signals of spoken information sometimes presented to pilots whose visual and haptic skills are already fully occupied while flying.

3. 3 Sounds

Non-Speech Sounds:

These are bongs, bangs, squeaks, clicks etc. that are commonly used for warnings and alarms. Fewer typing mistakes occur here with key clicks. It is also useful in video games that become uninteresting without sound.

Unlike speech, it is language and culture independent.

Non-Speech Sounds provide dual mode displays in information presented along two different sensory channels. It provides redundant presentation of information in the resolution of ambiguity in one mode through information in another. It is good for providing both transient and background status information e.g. Sound can be used as a redundant mode in the Apple Macintosh. Also, almost any user action (file selection, window active, disk insert, search error, copy complete, etc.) can have a different sound associated with it.

Auditory Icons

These use natural sounds to represent different types of object or action. Natural sounds have associated semantics which can be mapped onto similar meanings in the interaction e.g. throwing something away such as the sound of smashing glass.

Problem sometimes arise because not all things have associated meanings.

Additional information can also be presented on muffled sounds if object is obscured or action is in the background. The use of stereo allows positional information to be added.

Examples:

SonicFinder for the Macintosh: Here, items and actions on the desktop have associated sounds. For example, folders have a papery noise and moving files produce a dragging sound.

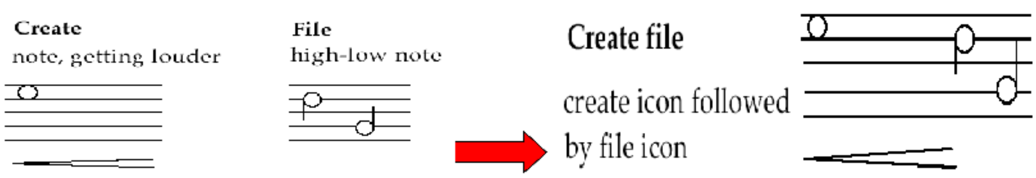
Copying sound gives a sound of a liquid being poured into a receptacle while rising pitch indicates the progress of the copy.

Big files have louder sound than smaller ones.

Earcons

These are synthetic sounds used to convey information. They comprise structured combinations of notes or motives that represent actions and objects. Motives are combined to provide rich information.

Compound earcons are multiple motives combined to make one more complicated earcon



Family ear cons

Here, similar types of earcons represent similar classes of action or similar objects .The family of "errors" would contain syntax and operating system errors. Earcons are easily grouped and refined due to compositional and hierarchical nature.

It is harder to associate with the interface task since there is no natural mapping.

3.4 Recognition and Gestures

Touch recognition:

Comprises:

i. Haptic interaction made up of cutaneous perception that provide tactile sensation and vibrations on the skin

ii. Kinaesthetic comprising movement and position and force feedback.

Touch recognition also include information on shape, texture, resistance, temperature, and comparative spatial factors.

Examples of technologies on touch recognition include electronic Braille displays and force feedback devices e.g. Phantom that recognises resistance and texture.

Handwriting recognition

Handwriting is another communication mechanism which we are used to in day-to-day life

The technology of handwriting consists of complex strokes and spaces.

The handwriting is captured by digitising tablet through strokes transformed to sequence of dots.

Large tablets available are suitable for digitising maps and technical drawings.

Smaller devices, some incorporating thin screens are used to display the information. Such include PDAs such as Palm Pilot and tablet PCs.

The problems associated with handwriting recognition are personal differences in letter formation and co-articulation effects.

The breakthroughs in this technology is the creation of stroke not just bitmap found in special 'alphabet' like Graffeti on PalmOS

The technology is usable even without training though many people prefer to use the keyboards!

Gesture technology;

This can be found in its various applications such as in gestural input - e.g. "put that there" and sign language.

The technology comprises data glove and position sensing devices such as MIT Media Room.

Gesture provides the benefits of natural form of interaction by pointing.

It enhances communication between signing and non-signing users

The problems with gesture interaction are that it is user dependent due to the variable nature of each user. Issues of co articulation are also considered as problems.

3.5 Devices for the Elderly and Disabled

The development of Technology on Human Computer Interaction has helped users with disabilities as follow:

Visual impairment: Use of screen readers and Sonic Finder

Hearing impairment: Use of text communication, gesture and captions

Physical impairment: Use of speech input and output, eye gaze, gesture, predictive systems (e.g. reactive keyboard)

Speech impairment: Use of speech synthesis and text communication

Dyslexia: Use of speech input and output
Autism: Use of communication and education devices
Older people use disability aids, memory aids, and communication tools to prevent social isolation

Others:
Children use appropriate input and output devices for education, games and fun.
In solving cultural differences, the influence of nationality, generation, gender, race, sexuality, class, religion, political persuasion etc. are affected by the interpretation of interface features. e.g. interpretation and acceptability of language, cultural symbols, gesture and colour.

4.0 Conclusion

Since the basic goal of HCI study is to improve the interactions between users and computers by making computers more usable and receptive to the user's needs, there is continuous research in human-computer interaction that involves exploring easier-to-learn or more efficient interaction techniques for common computing tasks. This includes inventing new techniques and comparing existing techniques using scientific methods.

5.0 Summary

Uncommon technologies associated with human computer interaction include consideration of the multi-modal and multi-media systems that incorporate speech recognition and synthesis, the phonetic typewriter, sound interface facilities, recognition and gestures mechanisms. These facilities particularly aid the elderly and the disabled to effectively and comfortably interact with the computer system.

6.0 Tutor Marked Assignment

1. What are multisensory systems, their components, and their relevance in the design of interactive systems?
2. What is speech Synthesis and how is it valuable to the computer user? Give two examples of its application.
3. Distinguish between an auditory icon and an earcon. Explain the limitation of their applications.
4. Explain the three types of Recognition and Gestures mode of interaction. Mention areas where each is effectively applied.

S. Mention any 3 devices that aid the elderly and the disabled in human computer interaction.

7.0 Further Readings / References

- Coons, S. "An Outline of the Requirements for a Computer-Aided Design System," in *AFIPS Spring Joint Computer Conference.* 196 3. 2 3. pp. 299- 304.
- Engelbart, D. and English, W., "A Research Center for Augmenting Human Intellect." *Reprinted in ACM S IGGRAPH Video Review, 1994.*, 1968. 106
- English, W.K., Engelbart, D.C., and Berman, M.L., "Display Selection Techniques for Text Manipulation." *IEEE Transactions on Human Factors in Electronics*, 1967. HFE-8(1)

MODULE 2: USER PERSPECTIVES OF HUMAN-COMPUTER INTERACTION

UNIT 1: USER ORIENTED PERSPECTIVE OF HUMAN COMPUTER INTERACTION : SOCIAL HUMAN THRUST

Table of contents

1.0 Introduction

This unit concerns socio-organizational issues and stakeholder requirements
Organizational issues affect acceptance of new computer systems because where there are conflicts and power, the question arises on who benefits and who encourages the use.
Stakeholders of a new computer system identify their requirements in organizational context. Organizational context may play different roles in internal and external situations. For an organizational information system to be used by the organization's own employees, organizational context analysis answers questions such
as: What is the larger system where this information system is embedded? What are the interactions with other entities in the organization? What are the organizational policies or practice that may affect individual's attitude and behavior towards using the system?
This event is the sociological, organizational, and cultural impact of computing. In other words, the organizational, social and cultural context in which humans interact with IT. This context is largely the result of the broad adoption of IT by organizations and society to support organizational functions and goals and to enhance society's development.

2.0 Objectives

By the end of this unit, you should be able to understand:

i. The socio-technical models that look at human and technical requirements
ii. The soft systems methodology that considers the broader view of human and organizational issues
iii. The participatory design that includes the user directly in the design process
iv. The ethnographic methods that study users in context with unbiased perspective

.0 MAIN CONTENT

3.1 Organisational issues

Organisational factors

Organisational factors can make or break a system. Studying the work group is not sufficient since any system is used within a wider context and the crucial people need not be the direct users.
Therefore, before installing a new system, one must understand:

- Who benefits?

- Who puts in effort?
- The balance of power in the organisation and how the system will be affected.

Even when a system is successful, it may be difficult to measure that success

Conflict and power

In computer supported *cooperative* work (CSCW), people and groups have conflicting goals hence the systems design assumes cooperation will fail! For example, in a computerised stock control, the store man looses control of information and may decide to subvert the system before or after it becomes operational. Therefore, it is important to identify stakeholders — not just the users.

Organisational structures

Groupware affects organisational structures so also communication structures reflect line management. For example, email is a cross-organisational communication.
Organisation structure can also disenfranchise lower management and disaffected staff could 'sabotage'.
Because technology *can* be used to change management style and power structures, the implementation of such technology would improve upon organizational efficiency.
For example, organizational efficiency may be expected due to redesign of workflows among critical business units that is affected by the implemented IT; satisfaction and retention of customers/clients are anticipated due to accurate and fast information gathering and presentations, to name a few. It is noteworthy that some of the organizational or societal impacts may not be tangible or directly attributed to HCI considerations. This assertion is in line with the issues of determining IT values in organizations and societies.

3.2 Invisible workers

Telecommunications improvements allow neighbourhood work centres and home-based tele-working.
Many ecological and economic benefits arise from tele-working such as reduced travel and flexible family commitments.
But 'management by presence' is absent. The presence in the office increases perceived worth and reduces problems for promotion.
Barriers to tele-working are both managerial and social but *not* technological.
The new system should benefit all. Disproportionate effort should be avoided. Examine who puts in the effort and who gets the benefit.
It is possible to get benefit without doing work, if everyone does it, system falls into disuse
To get started, look for cliques to form core user base and design to benefit an initial small user base

Evaluating the benefits

Assuming we have avoided the pitfalls!
How do we measure our success?
Job satisfaction and information flow and economic benefit should diffuse throughout the organisation
There is the need to identify requirements within context of use and the need to take account of stakeholders. Work groups and practices should be identified in organisational context
Many approaches including socio-technical modelling, soft system modelling, participatory design and contextual inquiry are used.

Who are the stakeholders?

The system will have many stakeholders with potentially conflicting interests. Stakeholder is anyone affected by success or failure of system. The primary stakeholders actually use the system, the secondary receive output or provide input, while the tertiary have no direct involvement but are those affected by the success or failure of the new system.

Facilitators are those involved in the development or deployment of the system.

Example: Classifying stakeholders — an airline booking system
An international airline is considering introducing a new booking system for use by associated travel agents to sell flights directly to the public.
Primary stakeholders: The travel agency staff and the airline booking staff
Secondary stakeholders: Customers and the airline management
Tertiary stakeholders: Competitors, civil aviation authorities, customers' travelling companions and, airline shareholders
Facilitating stakeholders: The design team and the Information Technology department staff

Designers need to meet as many stakeholder needs as possible. Usually needs may be in conflict so they have to prioritise, often priority decreases as one moves down the categories

3. 3 Socio-technical modelling

This is a response to *technological determinism* and it is concerned with the technical, social, organizational and human aspects of design. It also describes the impact of specific technology on organization. It is concerned with information gathering such as interviews, observation, discussion with focus groups and document analysis.
Several approaches to be considered are the Stakeholders' Focus and the ESTA (Eight Stage Task Analysis).

3. 3.1 Stakeholders' Focus

This comprises six stage processes that focus on stakeholders
The first describes the organizational context, including primary goals, physical characteristics, political and economic background
The second identifies and describes the stakeholders including personal issues, role in the organization and their job.
The third identifies and describes the work-groups whether formally constituted or not
The fourth identifies and describe task—object pairs; these are tasks to be performed and objects used
The fifth identifies the stakeholder needs: Stages 2—4 described above are in terms of both current and proposed system, the stakeholder needs are identified from the differences between the two.
Lastly, we consolidate and check the stakeholder requirements against earlier criteria.

3. 3.2 ESTA (Eight Stage Task Analysis)

This is an eight stage model that focuses on task
The primary task is identified in terms of users' goals
Secondly, task inputs to the system are identified
Thirdly, the external environment into which the system will be introduced is described, including physical, economic and political aspects
Fourthly, the transformation processes within the system are described in terms of actions performed on or with objects
In the fifth stage, the social system is analyzed by considering existing internal and external work-groups and relationships
At the sixth stage, the technical system is described in terms of configuration and integration with other systems
At the seventh stage, the performance satisfaction criteria are established, indicating social and technical requirements of the system.
The last stage specifies the new technical system.

3.4 Soft systems methodology

The soft systems methodology considers the broader view of human and organizational issues
There is no assumption of technological solution here — emphasis is on understanding the situation fully
This methodology was developed by Checkland.
The seven stages involved here are:

i. Recognition of problem and initiation of analysis
ii. Detailed description of problem situation; it is a rich picture stage
iii. Generation of the root definitions of system: this is known as CATWOE (see definition below)
iv. Conceptual model - this identifies transformations
v. This compares real world to conceptual model
vi. Identifies necessary changes
vii. Determines actions to effect changes

CATWOE (Clients, Actors, Transformation, World View, Owner, Environment) further defines and explains the soft systems methodology.
Clients: those who receive output or benefit from the system
Actors: those who perform activities within the system
Transformations: the changes that are affected by the system
World View - how the system is perceived in a particular root definition
Owner: those to whom the system belongs, to whom it is answerable and who can authorize changes to it
Environment: the world in which the system operates and by which it is influenced

3.5 Participatory design

In participatory design, workers enter into design context while in ethnography (as used for design), the designer enters into work context. Both make workers feel valued in design and encourages workers to 'own' the products. The user is an active member of the design team.

Characteristics
Participatory design is context and work oriented rather than system oriented.
It is collaborative and iterative
Methods involved are: brain-storming, storyboarding, workshops, pencil and paper exercises.

Ethics
The ethics involved the participatory socio-technical approach devised by Mumford.
It states that the system development is about managing change and that non-participants are more likely to be dissatisfied.
There are three levels of participation: consultative, representative, and consensus.
Design groups including stakeholder representatives make design decisions and job satisfaction is the key to solution
(See the unit on Participatory Design for more details)

3.6 Ethnography

This is very influential in CSCW
It is a form of anthropological study with special focus on social relationships and does *not* enter actively into situation.
It seeks to understand social culture, it is unbiased and open ended.

3.7 Contextual inquiry

Here inquiry is conducted in ethnographic tradition but acknowledges and challenges investigator focus.
It creates a model of investigator being apprenticed to the user in order to learn about the work.

The investigation takes place in the workplace with detailed interviews, observation, and analysis of communications, physical workplace, and artefacts.
Number of models created is according to sequence, physical, flow, cultural, and artefact
The models are consolidated across users, while the output indicates task sequences, the artefacts and communication channels needed, the physical and cultural constraints

4.0 Conclusion
Work groups and practices should be identified in organisational context
Many approaches including socio-technical modelling, soft system modelling, participatory design and contextual inquiry have been explained.
In concluding, there is the need to identify requirements within context of use and the need to take account of stakeholders.

5.0 Summary
The socio human thrust analysis explains the socio-technical models of human and technical requirements, the systems methodology that considers human and organizational issues, the participatory design that includes the user directly in the design process and the ethnographic methods that study users in context with unbiased perspective.

6.0 Tutor Marked Assignment
1. "Organizational factors can make or break a system". Explain the concepts of this expression as it affects users in an organization when designing a new interactive computer system.
2. What do you understand by a Computer Supported cooperative Work (CSCW) in an organization. Why is it important to identify and consider stakeholders when designing interactive systems in an organization?
3. What are the factors used in evaluating whether a designed and implemented system is successful in an organization?
4. Who is a stakeholder in a human computer interaction? Differentiate between a primary stakeholder, a secondary stakeholder and a facilitator of a newly designed interactive system.
5. Differentiate between the different goals of the six stage model of Stakeholders' focus and the eight stage model of ESTA within the human aspects of interactive design.
6. What is the full meaning of the acronym "CATWOE"?
7. Explain what you understand by Participatory Design of Human Computer Interaction. What are its characteristics and the Ethics involved?

7.0 Further Readings / References

- Reddy, R., "To Dream the Possible Dream (Turing Award Lecture)." *Communications of the ACM*, 1996. 39(5): pp. 105-112.
- Robertson, G., Newell, A., and Ramakrishna, K., *ZOG: A Man-Machine Communication Philosophy* . Carnegie Mellon University Technical Report Report, Number, August, 1977.

UNIT 2: USER ORIENTED PERSPECTIVE OF HUMAN COMPUTER INTERACTION : COGNITIVE HUMAN THRUST

Table of contents

1.0 Introduction

Cognitive Human Thrust incorporates cognitive psychology and cognitive modeling. Cognitive psychology plays an important role in Human-computer Interaction.
Perception, memory, mental models and metaphors, knowledge representations, problem solving, errors and
learning, are all topics under cognitive psychology that have direct implications to HCI design.

Cognitive modeling involves creating a computational model to estimate how long it takes people to perform a given task. Models are based on psychological principles and experimental studies to determine times for cognitive processing and motor movements. Cognitive models can be used to improve user interfaces or predict problem errors and pitfalls during the design process.

2.0 OBJECTIVES

By the end of this unit, the student should be able to:

- Be familiar with different types of Cognitive models
- Know available inspection methods
- Know how to apply inquiry methods on problems of cognition
- Explain prototyping methods
- Carry out evaluation using the tests and metrics methods

.0 MAIN CONTENT

3.1 Cognitive Models

3.1.1 Parallel Design

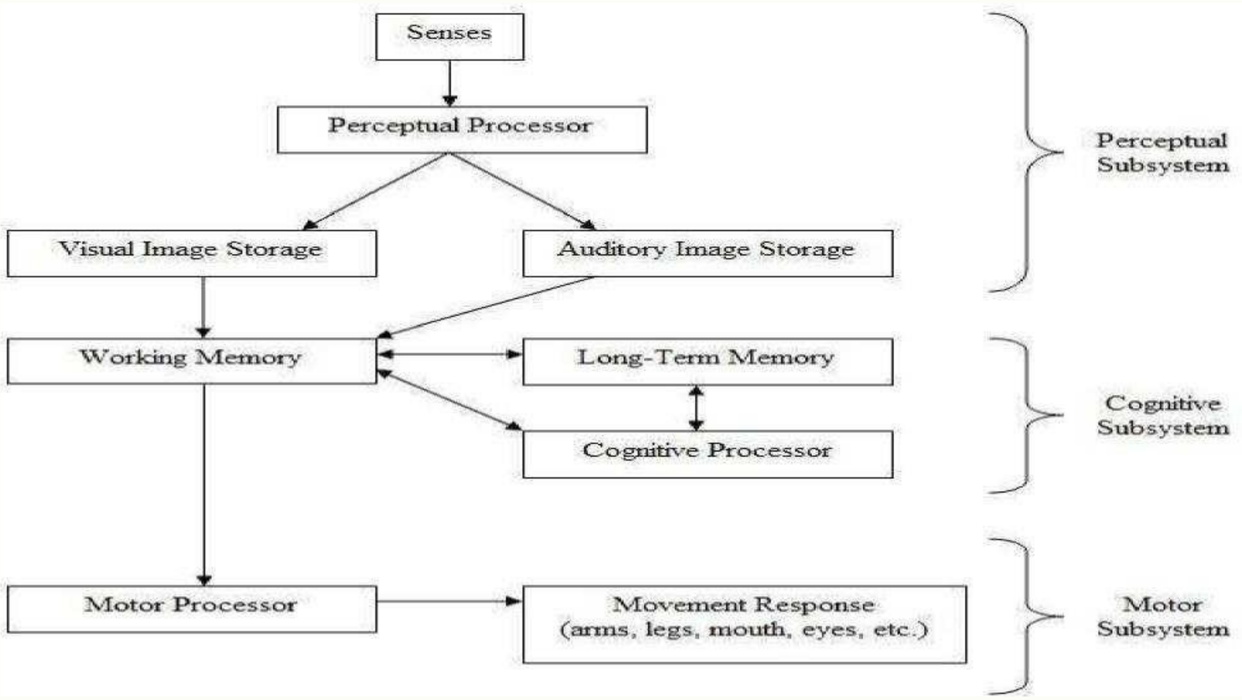
With parallel design, several people create an initial design from the same set of requirements. Each person works independently, and when finished, shares his/her concepts with the group. The design team considers each solution, and each designer uses the best ideas to further improve their own solution. This process helps to generate many different, diverse ideas and ensures that the best ideas from each design are integrated into the final concept. This process can be repeated several times until the team is satisfied with the final concept.

3.1.2 .2 GOMS

GOMS is an acronym that stands for Goals, Operator, Methods, and Selection Rules. It is a family of techniques that analyzes the user complexity of interactive systems. Goals are what the user has to accomplish. An operator is an action performed in service of a goal. A method is a sequence of operations that accomplish a goal. Selection rules specify which method should be used to satisfy a given goal, based on the context.

3.1. 3 Human Processor Model

Sometimes it is useful to break a task down and analyze each individual aspect separately. This allows the tester to locate specific areas for improvement. To do this, it is necessary to understand how the human brain processes information. A model of the human processor is shown below.



Many studies have been done to estimate the cycle times, decay times, and capacities of each of these processors. Variables that affect these can include subject age, ability, and the surrounding environment. For a younger adult, reasonable estimates are:

Parameter	Mean	Range
Eye movement time	230 ms	70-700 ms
Decay half-life of visual image storage	200 ms	90-1000 ms
Perceptual processor cycle time	100 ms	50-200 ms
Cognitive processor cycle time	70 ms	25-170 ms
Motor processor cycle time	70 ms	30-100 ms
Effective working memory	7 items	5-9 items

Long-term memory is believed to have an infinite capacity and decay time.

Keystroke level modeling

Keystroke level modeling is essentially a less comprehensive version of GOMS that makes simplifying assumptions in order to reduce calculation time and complexity.

3.2 Inspection methods

These usability evaluation methods involve observation of users by an experimenter, or the testing and evaluation of a program by an expert reviewer. They provide more quantitative data, as tasks can be timed and recorded.

3.2.1 Card Sorting

Card sorting is a way that involves users in grouping information for a website's usability review. Participants in a card sorting session are asked to organize the content from a Web site in a way that makes sense to them. Participants review items from a Web site and then group these items into categories. Card sorting helps to learn how users think about the content and how they would organize the information on the Web site. Card sorting helps to build the structure for a web site, decide what to put on the home page, and label the home page categories. It also helps to ensure that information is organized on the site in a way that is logical to users.

3.2.2 Ethnography

Ethnographic analysis is derived from anthropology. Field observations are taken at a site of a possible user, which track the artifacts of work such as Post-It notes, items on desktop, shortcuts, and items in trash bins. These observations also gather the sequence of work and interruptions that determine the user's typical day.

3.2. 3 Heuristic Evaluation

Heuristic Evaluation is a usability engineering method for finding and assessing usability problems in a user interface design as part of an iterative design process. It involves having a small set of evaluators examining the interface and using recognized usability principles (the "heuristics"). It is the most popular of the usability inspection methods, as it is quick, cheap, and easy.

Heuristic evaluation was developed to aid in the design of computer user-interface design. It relies on expert reviewers to discover usability problems and then categorize and rate them by a set of principles (heuristics.) It is widely used based on its speed and cost-effectiveness. Jakob Nielsen's list of ten heuristics is the most commonly used in industry. By determining which guidelines are violated, the usability of a device can be determined.

3.2.4 Usability Inspection

Usability Inspection is a review of a system based on a set of guidelines. The review is conducted by a group of experts who are deeply familiar with the concepts of usability in design. The experts focus on a list of areas in design that have been shown to be troublesome for users.

3.2.5 Pluralistic Inspection

Pluralistic Inspections are meetings where users, developers, and human factors people meet together to discuss and evaluate step by step of a task scenario. As more people inspect the scenario for problems, the higher the probability to find problems. In addition, the more interaction in the team, the faster the usability issues are resolved.

3.2.6 Consistency Inspection

In consistency inspection, expert designers review products or projects to ensure consistency across multiple products to look if it does things in the same way as their own designs.

3.2.7 Activity Analysis

Activity analysis is a usability method used in preliminary stages of development to get a sense of a situation. It involves an investigator observing users as they work in the field. Also referred to as user observation, it is useful for specifying user requirements and studying currently used tasks and subtasks. The data collected is qualitative and useful for defining the problem. It should be used when you wish to frame what is needed, or "What do we want to know?"

3. 3 Inquiry methods

The following usability evaluation methods involve collecting qualitative data from users. Although the data collected is subjective, it provides valuable information on what the user wants.

3. 3.1 Task Analysis

Task analysis means learning about users' goals and users' ways of working. Task analysis can also mean figuring out what more specific tasks users must do to meet those goals and what steps they must take to accomplish those tasks. Along with user and task analysis, we often do a third analysis: understanding users' environments (physical, social, cultural, and technological environments).

3. 3.2 Focus Groups

A focus group is a focused discussion where a moderator leads a group of participants through a set of questions on a particular topic. Although typically used as a marketing tool, Focus Groups are sometimes used to evaluate usability. Used in the product definition stage, a group of 6 to 10 users are gathered to discuss what they desire in a product. An experienced focus group facilitator is hired to guide the discussion to areas of interest for the developers. Focus groups are typically videotaped to help get verbatim quotes, and clips are often used to summarize opinions. The data gathered not usually quantitative, but can help get an idea of a target group's opinion.

3. 3. 3 Questionnaires/Surveys

Surveys have the advantages of being inexpensive, require no testing equipment, and results reflect the users' opinions. When written carefully and given to actual users who have experience with the product and knowledge of design, surveys provide useful feedback on the strong and weak areas of the usability of a design. This is a very common method and often does not appear to be a survey, but just a warranty card.

.4 Prototyping methods

3.4.1 Rapid Prototyping

Rapid prototyping is a method used in the early stages of development to validate and refine the usability of a system. It can be used to quickly and cheaply evaluate user-interface designs without the need for an expensive working model. This can help remove hesitation to change the design, since it is implemented before any real programming begins. One such method of rapid prototyping is paper prototyping.

3.4.2 Subject Testing methods

These usability evaluation methods involve testing of subjects for the most quantitative data. Usually recorded on video, they provide task completion time and allow for observation of attitude.

3.4. 3 Remote usability testing

Remote usability testing is a technique that exploits users' environment (e.g. home or office), transforming it into a usability laboratory where user observation can be done with screen sharing applications.

3.4.4 Thinking Aloud Protocol

The Thinking Aloud Protocol is a method of gathering data that is used in both usability and psychology studies. It involves getting a user to verbalize their thought processes as they perform a task or set of tasks. Often an instructor is present to prompt the user into being more vocal as they work. Similar to the Subjects-in-Tandem method, it is useful in pinpointing problems and is relatively simple to set up. Additionally, it can provide insight into the user's attitude, which can not usually be discerned from a survey or questionnaire.

3.4.5 Subjects-in-Tandem

Subjects-in-tandem is pairing of subjects in a usability test to gather important information on the ease of use of a product. Subjects tend to think out loud and through their verbalized thoughts designers learn where the problem areas of a design are. Subjects very often provide solutions to the problem areas to make the product easier to use.

3.5 Other methods

3.5.1 Cognitive walkthrough

Cognitive walkthrough is a method of evaluating the user interaction of a working prototype or final product. It is used to evaluate the system's ease of learning. Cognitive walkthrough is useful to understand the user's thought processes and decision making when interacting with a system, specially for first-time or infrequent users.

3.5.2 Benchmarking

Benchmarking creates standardized test materials for a specific type of design. Four key characteristics are considered when establishing a benchmark: time to do the core task, time to fix errors, time to learn applications, and the functionality of the system. Once there is a benchmark, other designs can be compared to it to determine the usability of the system.

3.5. 3 Meta-Analysis

Meta-Analysis is a statistical procedure to combine results across studies to integrate the findings. This phrase was coined in 1976 as a quantitative literature review. This type of evaluation is very powerful for determining the usability of a device because it combines multiple studies to provide very accurate quantitative support.

3.5.4 Persona

Personas are fictitious characters that are created to represent the different user types within a targeted demography that might use a site or product. Alan Cooper introduced the concept of using personas as a part of interactive design in 1998 in his book *The Inmates Are Running the Asylum*, but had used this concept since as early as 1975.Personas is a usability evaluation method that can be used at various design stages. The most typical time to create personas is at the beginning of designing so that designers have a tangible idea of who the users of their product will be. Personas are the archetypes that represent actual groups of users and their needs, which can be a general description of person, context, or usage scenario. This technique turns marketing data on target user population into a few physical concepts of users to create empathy among the design team.

3.6 Evaluating with tests and metrics

Regardless of how carefully a system is designed, all theories must be tested using usability tests. Usability tests involve typical users using the system (or product) in a realistic environment. Observation of the user's behavior, emotions, and difficulties while performing different tasks, often identify areas of improvement for the system.

3.6.1 Prototypes

It is often very difficult for designers to conduct usability tests with the exact system being designed. Cost constraints, size, and design constraints usually lead the designer to creating a prototype of the system. Instead of creating the complete final system, the designer may test different sections of the system, thus making several small models of each component of the system. The types of usability prototypes may vary from using paper models, index cards, hand drawn models, or storyboards.

Prototypes are able to be modified quickly, often are faster and easier to create with less time invested by designers and are more apt to change design; although sometimes are not an adequate representation of the whole system, are often not durable and testing results may not be parallel to those of the actual system.

3.6.2 Metrics

Designers must use usability metrics to identify what it is they are going to measure, or the usability metrics. These metrics are often variable, and change in conjunction with the scope and goals of the project. The number of subjects being tested can also affect usability metrics, as it is often easier to focus on specific demographics. Qualitative design phases, such as general usability (can the task be accomplished?), and user satisfaction are also typically done with smaller groups of subjects. Using inexpensive prototype on small user groups, provide more detailed information, because of the more interactive atmosphere, and the designers ability to focus more on the individual user.

Testing the metrics

As the designs become more complex, the testing must become more formalized. Testing equipment will become more sophisticated and testing metrics become more quantitative. With a more refined prototype, designers often test effectiveness, efficiency, and subjective satisfaction, by asking the user to complete various tasks. These categories are measured by the percentage that complete the task, how long it takes to complete the tasks, ratios of success to failure to complete the task, time spent on errors, the number of errors, rating scale of satisfactions, number of times user seems frustrated, etc. Additional observations of the users give designers insight on navigation difficulties, controls, conceptual models, etc. The ultimate goal of analyzing these metrics is to discover a prototype design that users like to successfully perform given tasks.

Documenting metrics
After conducting usability tests, it is important for a designer to record what was observed, in addition to why such behavior occurred and modify the model according to the results. Often it is quite difficult to distinguish the source of the design errors, and what the user did wrong. However, effective usability tests will not generate a solution to the problems, but provide modified design guidelines for continued testing.

3.7 Benefits of usability

The key benefits of usability are:
Increased user efficiency
Reduced development costs
Reduced support costs
Corporate integration
By working to improve said factors, corporations can achieve their goals of increased output at lower costs, while potentially creating optimal levels of customer satisfaction. There are numerous reasons why each of these factors correlates to overall improvement. For example, making a piece of software's user interface easier to understand would reduce the need for extensive training. The improved interface would also tend to lower the time needed to perform necessary tasks, and so would both raise the productivity levels for employees and reduce development time and costs.

4.0 Conclusion

All the factors analysed above aid the design process and Increase usability in the workplace. They aid in fostering several responses from employees. Along with any positive feedback, workers who enjoy their work do it better, stay longer in the face of temptation, and contribute ideas and enthusiasm to the evolution of enhanced productivity.

5.0 Summary

Cognitive human thrust comprises cognitive psychology and modeling. Cognitive psychology studies human traits such as perception, memory, mental models and metaphors, among others- those that have direct implications to HCI design.
Cognitive modeling enables estimation of how long it takes people to perform a given task.
Inspection method is the process of observing users by an experimenter, or the testing and evaluation of a program by an expert reviewer.
Inquiry methods involve collecting qualitative data from users; such data provides valuable information on what the user wants.
Prototypes enable the designer test different sections of the system by making several small models of each component of the system.
Evaluation methods test and correct errors in implemented design.

6.0 Tutor Marked Assignment

1. What is the full meaning of the acronym GOMS as a cognitive model? Explain each of the terms in the GOMS
Distinguish between the following Cognitive models: (i) Parallel Design, (ii) Human Processor and (iii) Keystroke level
2. What is the primary benefit of the inspection methods used to evaluate usability in HCI? Briefly explain the following inspection methods:
(i) Card sorting (ii) Ethnography (iii) Heuristic evaluation (iv) Usability inspection (v) Activity analysis
3. What is the value of a Usability inspection and its focus?
4. Explain the following methods of inquiry in usability evaluation:
Task analysis, Group Focus, Questionnaire and Survey methods.

5 (a) What do you understand by the term Prototype as related to the design of Human Computer Interaction?

(b) In designing and evaluating usability, prototyping methods have been very valuable. Show your understanding of the following Prototyping methods:

(i)Rapid Prototyping, (ii) Testing methods, (iii) Remote Usability testing,

(iv)Thinking aloud Protocol, (v) Subjects in Tandem

6. The following are methods used to evaluate the design of the user interaction with the Computer before and /or after implementation: Cognitive Walkthrough, Benchmarking Meta analysis and Persona. Which of the four above do you consider as the most suitable method and why?

7.0 Further Readings / References

- Ronald M. Baecker, Jonathan Grudin, William A. S. Buxton, Saul Greenberg (Eds.) (1995): *Readings in human—computer interaction. Toward the Year 2000*. 2. ed. Morgan Kaufmann, San Francisco 1995 ISBN 1-558-60246-1
- Jakob Nielsen: *Usability Engineering*. Academic Press, Boston 199 3 ISBN 0-12-518405-0
- Donald A. Norman: *The Psychology of Everyday Things*. Basic Books, New York 1988 ISBN 0-465-06709- 3
- Jef Raskin: *The humane interface. New directions for designing interactive systems*. Addison-Wesley, Boston 2000 ISBN 0-201- 379 37-6
- Ben Shneiderman and Catherine Plaisant: *Designing the User Interface: Strategies for Effective Human—Computer Interaction*. 4th ed. Addison Wesley, 2004 ISBN 0- 321-19786-0

UNIT 3: SYSTEM ORIENTED PERSPECTIVE OF HUMAN COMPUTER INTERACTION

IMPROVING UPON THE USER'S TECHNOLOGICAL PERSPECTIVE

Table of contents

1.0 Introduction

The input and output technological perspective of the system from the user's view is enhanced by the kind of support given the users of the system.

There are different types of support available to the users at different times particularly important during the implementation and presentation stages. Hence the computer interaction components require careful design.
Types of user support that can be designed and offered include quick rreferences, task specific help, full explanation, and tutorials.
The kind of help solutions are provided on specific problem oriented operations while documentation solutions are given on system oriented and general operations. The same design principles apply to both.

2.0 Objectives

By the end of this unit, the student should be able to :

- Understand help supports available to the users
- Describe user modelling and knowledge representation
- Know how to design user supports

.0 Main Content

.1 Technical Support offered the System Users

The following are the requirements of an effective design of technical support that can be offered to the system users
Continuous access concurrent to main application should be made available
Help Support
For accuracy and completeness, the help should match and cover actual system behaviour
There must be consistency between different parts of the help system and paper documentation
Robustness should exist for correct error handling and unpredictable behaviour
There should be flexibility such that the system allows the user to interact in a way appropriate to experience and task
Encountering problems while operating the system should not prevent the user continuing with the work

The approaches to user support include the following:
Command assistance should enable the user to request help on a particular command e.g., UNIX and DOS help can be good for quick reference.
If we assume that the user has knowledge of the command and what to look for, then the command prompts should provide information about correct usage when an error occurs and be good for simple syntactic errors

Context sensitive help:
Ensure that the help request is interpreted according to context in which it occurs. e.g. tool tips
The on-line tutorials allow the user to work through the basics of application in a test environment. Though often inflexible, it can be useful.
The on-line documentation though available in paper documentation is also made available on the computer. It should be continually available in common medium as well.
Since on-line documentation can be difficult to browse, then hypertext should be included to support browsing..

Adaptive Help Systems
These use knowledge of the context, the individual user, the task, the domain and the instruction to provide help adapted to the user's needs.

The problems with adaptive help systems are that they require considerable knowledge, the interaction is not controlled and it is difficult knowing what should be adapted and the scope of the adaptation.

Issues concerned in adaptive help:
Initiative ; the question is whether the user retain control or can the system direct the interaction of the user, and can the system interrupt the user to offer help?
Effect ; the question is what is going to be adapted and what information is needed to do this? Only what is needed is modelled.
Scope ; Is the scope of the modelling at application or system level? It is more complex at system level e.g. expertise varies between applications.

Wizards and assistants:
Wizards
Wizard is a task specific tool that leads the user through a task, step by step, using the user's answers to specific questions
Example is in the preparation of resume.
Wizard is useful for the safe completion of complex or infrequent tasks.
It has a limited flexibility in a constrained task execution so it must allow the user to go back to the beginning of the task.
Assistants
Assistants monitor the user behaviour and offer contextual advice though it can be irritating e.g. as in MS paperclip.
They must be under the user control e.g. XP smart tags

3.2 User modelling
In the user modelling of the knowledge representation, all help systems have a model of the user. It may be a single or generic user (non-intelligent user). The models could be an adaptable user-configured or adaptive system-configured.

Approaches to user modelling
Quantification: Here the user moves between levels of expertise based on quantitative measure of what he knows.
Stereotypes: The user is classified into a particular category.
Overlay: The idealized model of an expert use is constructed and actual use compared to ideal. The model may contain the commonality or difference
In a special case, the user behaviour is compared to a known error catalogue.

Knowledge representation
Knowledge representation occurs when knowledge is presented as rules, and facts are interpreted using inference mechanism.
The domain and task modelling of the knowledge representation covers common errors and tasks particularly the current task.
It usually involves the analysis of command sequences.
However the problems here are concerned on how to represent the tasks particularly when interleaved, and how to know the user's intention.

Knowledge representation: Advisory strategy
The advisory strategy for knowledge representation involves choosing the correct style of advice for a given situation in form of a reminder, tutorial, etc.
Few intelligent help systems model advisory strategy, but choice of strategy is still important.

Techniques for knowledge representation
The techniques for knowledge representation are rule based (e.g. logic, production rules) when knowledge are presented as rules and facts are interpreted using inference mechanism. They can also be used in relatively large domains..
It is frame based (such as a semantic network) when knowledge stored in structures with slots are to be filled but useful for a small domain..

Network based
The knowledge is network based when represented as relationships between facts and can be used to link frames.
It is example based when the knowledge is represented implicitly within decision structure and trained to classify rather than programmed with rules. This one requires little knowledge acquisition

Problems with knowledge representation and modelling
The problems here include knowledge acquisition, the resources and the interpretation of user behaviour

3. 3 Designing user support
User support is not an 'add on' but should be designed integrally with the system.
The designer should concentrate on content and context of help rather than on technological issues.

3. 3.1 Presentation issues in designing user support:
How is help requested? Is it at the command level, by button, by on/off function, or by separate application?
How is help displayed? Is it through a new window, or a whole screen, a split screen, pop-up boxes, or hint icons?
The designer should note that effective presentation requires clear, familiar, and consistent language. It should contain instructional rather than descriptive languages. Blocks of text should be avoided.

3. 3.2 Implementation issues in designing user support
Implementation issues are whether the help is in form of an operating system command, a Meta command or an application.
There should be a clear indication of summary and example information.
What are the resources available in terms of screen space, the memory capacity and the speed of processing?
Is the structure of help data in form of a single file, a file hierarchy or a database?
Other issues concern the flexibility and extensibility of implementation and whether it is made in hard copy or by browsing.

4.0 Conclusion
Computer interaction components require careful design. The design should ensure that there are different types of support available to the users at different times particularly during the implementation and presentation stages.
Help solutions should be provided on specific problem oriented operations, while documentation solutions are given on system oriented and general operations.

5.0 Summary
Effective design of technical support reflects in continuous access concurrent to main application being made available

Accuracy and completeness are guaranteed when the help support matches and covers actual system behaviour
Wizard is a task specific tool that leads the user through a task, step by step, using the user's answers to specific questions while assistants monitor the user behaviour and offer contextual advice.
User modelling occurs when all help systems have a model of the user. Knowledge representation is the presentation of knowledge as rules and facts, and interpreted using inference mechanism.
User support should be designed integrally with the system. The designer should concentrate on content and context of help rather than on technological issues.

6.0 Tutor Marked Assignment

1. Mention three kinds of Help Support that can be designed for the Computer System Users. What are their demerits?
2. What is Knowledge representation? Describe the rule based, the frame based and the networked based techniques of Knowledge representation.
3. Designing the user support requires considering some presentation and implementation issues. Describe two of presentation issues and three of implementation issues that should guide the designer

7.0 Further Readings / References

- Henderson Jr, D.A. "The Trillium User Interface Design Environment," in *Proceedings SIGCH I'86: Human Factors in Computing Systems.* 1986. Boston, MA. pp. 221-227.
- Myers, B.A., *et al.*, "Garnet: Comprehensive Support for Graphical, Highly-Interactive User Interfaces." *IEEE Computer*, 1990. 2 3(11): pp. 71-85.
- Stallman, R.M., *Emacs: The Extensible, Customizable, Self-Documenting Display Editor* . MIT Artificial Intelligence Lab Report, Number, Aug, 1979, 1979.

UNIT 4: DEVICES TECHNOLOGICAL PERSPECTIVE
INTERACTION STYLES AND DEVICES TECHNOLOGICAL PERSPECTIVE

Table of contents

1.0 INTRODUCTION

An Interaction style can be described as an interaction technique that shares the same metaphor or design principle. Examples are command line and direct manipulation user interfaces.
Two major classes of interaction styles that will be considered are the dialogue style of Interaction between computer and user and the distinct styles of interaction

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Understand the various available interaction styles
- Distinguish among the interaction styles
- Understand elements of WIMP interface
- Understand and be able to choose Widgets

.0 MAIN CONTENT

.1 Interaction styles

Common Styles of Interaction are:
Command line interface
Menus
Natural language
Question and answer, and query dialogue
Form-fills and Spreadsheets
WIMP (Widows, Icons, Menus, Pointers) Interface
Point and Click
Three—dimensional interfaces

Graphical user interface (GUI)
Copy and paste, Cut and paste
Single Document Interface, Multiple Document Interface, Tabbed Document Interface
Drag-and-drop
Cursor
Widgets (computing)
Direct manipulation interface
Zooming User Interface (ZUI)

Interaction paradigms include
Hypertext, hypermedia and hyperlinks
Speech recognition, Speech synthesis, Natural Language Processing, Non-speech audio input
Mouse gestures and handwriting recognition
Haptics and Telehaptics
Computer-mediated reality
Virtual Reality (VR)
Augmented Reality (AR)
CSCW: Computer Supported Collaborative (or Cooperative) Work, collaborative software
Ubiquitous Computing ("ubicomp")
Wearable computers
Brain-computer interface

Miscellaneous
Handheld devices
Human Computer Information Retrieval
Information retrieval

Internet and the World Wide Web
Multimedia
Software agents
Universal usability
User experience design
Visual programming languages

Brief explanation will be given on each of the common interfaces:

Command line interface
This is the way of expressing instructions to the computer directly and it comprises the function keys, single characters, short abbreviations, whole words, or a combination of them.
The characteristics of command line interface are as follow:
It is suitable for repetitive tasks
It is more valuable for expert users than novices
It offers direct access to system functionality
The command names and abbreviations should however be meaningful for an effective interface.
A typical example is found in the Unix Operating System

Menus
These are set of options displayed on the screen
Its characteristics include the following:
The Menu Options are visible, have less recall and it is easier to use
Because it relies on recognition the names should be meaningful
Menu Selection is done by either clicking numbers, letters, using the arrow keys or mouse or combination (e.g. mouse plus accelerators)
Menu options are often hierarchically grouped in a sensible manner.
It is a restricted form of full WIMP system

Natural language
This is a language that should be familiar to the user
It is in a form of speech recognition or typed natural language
Problems with the use of natural language are that it could be vague, ambiguous, and hard to use.
Part of the solutions to this is for the user to try and to understand a subset of the language thereby picking on key words.

Query interfaces
This basically comprises Question and answer interfaces where the user is led through interaction via series of questions to be answered.
It is suitable for novice users but has restricted functionality
It is often used in information systems
Some of these Query languages include the SQL used to retrieve information from database. This requires understanding of database structure and language syntax, hence requires some expertise

Form-fills
This is primarily used for data entry or data retrieval.
It is like a screen like paper form in which data is input in relevant place.
It requires good design and obvious correction facilities
An example of a Form-fill:

Spreadsheets
The first spreadsheet was VISICALC followed by Lotus 1-2- 3. However MS Excel i the most common spreadsheet today having a sophisticated variation of form-filling.
It has grid of cells containing a value or a formula. The formula can involve values of other cells e.g. sum of all cells in this column. The user can nter and alter data in the spreadsheet with considerable consistency.

Point and click interfaces
This is used in multimedia ,web brow sers and hypertext. It is sometimes called 'j st click something'! using icons, text links or location on map.
It requires minimal typing

Three dimensional interfaces
This is made up of virtual reality, 'ord inary' window systems and 3D workplace
The 'ordinary' window systems com rises highlighting, visual affordance and indi scriminate use
The 3D workspaces have uses for ext ra virtual space, it is light and occlusion to gi ve depth and distance effects

WIMP Interface
'WIMP' stands for Windows, Icon s, Menus and Pointers (or Windows, Icons, Mice, and Pull-down menus!)
It is a default style interface for majo rity of interactive computer systems, especi lly PCs and desktop machines
WIMP Interface
Windows
Icons
Menus
Pointers
... or windows, icons, mice, an d pull-down menus!

Elements of WIMP interface:
The elements of the WIMP interface are:
Windows, Icons, Menus and Pointers
Buttons, Toolbars, Palettes and Dialo g boxes

Details are given below:
Windows

These are areas of the screen that behave as if they were independent and can contain text or graphics which can be moved or resized. They can overlap and obscure each other, or can be laid out next to one another (tiled)

They are made up of

ii. scrollbars that allow the user to move the contents of the window up and down or from side to side
iii. title bars that describe the name of the window

Icons

Icons comprise small picture or image that represents some object in the interface. It often represents a window or action. The windows can be closed down or 'iconised'

A small representation may fit into many accessible windows. The icons can be many and various.

They are highly stylized with realistic representations.

Pointers

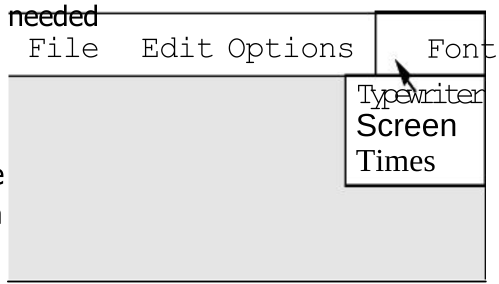
These are important component WIMP style that relies on pointing and selecting things it uses mouse, trackpad, joystick, trackball, cursor keys or keyboard shortcuts to access wide variety of graphical images

Menus

These are choice of operations or services offered on the screen

The required option is selected with pointer. However, this takes a lot of screen space

This problem is partly solved when a pop-up menu appears when needed



Kinds of Menus

Menu Bar at top of screen (normally), menu drags down

i Pull-down menu - mouse hold and drag down me
ii Drop-down menu - mouse click reveals men
iv. Fall-down menus - mouse just moves over bar!

Contextual menu appears where you are

Pop-up menus take actions for selected object

Pie menus are arranged in a circle

This is easier to select item over larger target area

It is also quicker because it can move same distance to any option but this is not widely used!

Menus extras

Cascading menus

This has a hierarchical menu structure in which a menu selection opens new menu and so in ad infinitum

Keyboard accelerators

This comprises key combinations with same effect as menu item.

They are of two kinds:

- active when menu open — usually first letter and
- active when menu closed — usually Ctrl + letter

3.2 Menus design issues

In order to design an effective menu, the following issues should be considered:

- Which kind to use
- What to include in menus at all

- Words to use in action or description
- How to group items
- Choice of keyboard accelerators

Buttons
This is an individual and isolated region within a display that can be selected to invoke an action
The Special kinds that exist are
The radio buttons with a set of mutually exclusive choices and the check boxes with a set of non-exclusive choices.

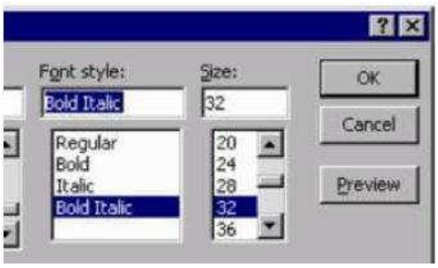
Toolbars
These are long lines of icons with fast access to common actions and are often customizable:
You can choose *which* toolbars to see and choose *what* options are on it

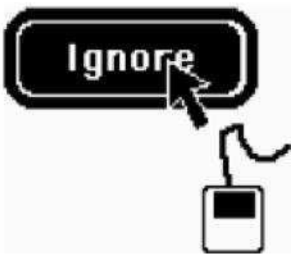
Palettes and tear-off menus
Palettes are little windows of actions shown or hidden via menu option in available shapes in drawing package
In tear-off and pin-up menus, menu 'tears off' to become palette

Dialogue boxes
These are information windows that pop up to inform of an important event or request information, for example when saving a file, a dialogue box is displayed to allow the user to specify the filename and location. Once the file is saved, the box disappears.

. 3 UNDERSTANDING AND CHOOSING WIDGETS

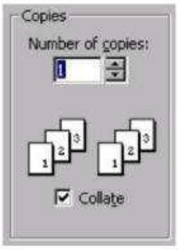
Widgets are bits that make the Graphical User Interface. They are the individual items on a Graphical User Interface (GUI). They can also be called the elements of the WIMP interface.
Examples of widgets include the check boxes, the tool bars, the buttons, etc. See the pictorial illustration below.

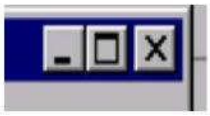
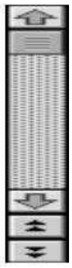




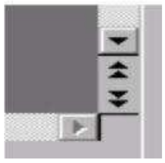
Three aspects of widgets can be identified in the following ways:
By appearance in the way they look like
By the nature of their interaction as to how they behave and
By their semantics as regards their meaning

By appearance:
Appearance includes words that are verbs that represent some action such as qu it, exit, embolden, and italicize.

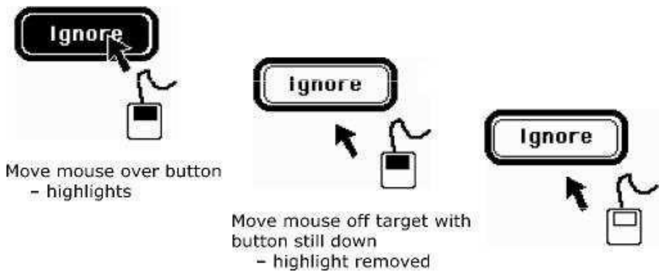



They could also be adjectives that de scribe the state of those words such as bold, italic etc.
They could be nouns that represent the name of the appearance such as Times New Roman, etc.
They could be combination of verbs nd adjectives e.g. embolden + italic

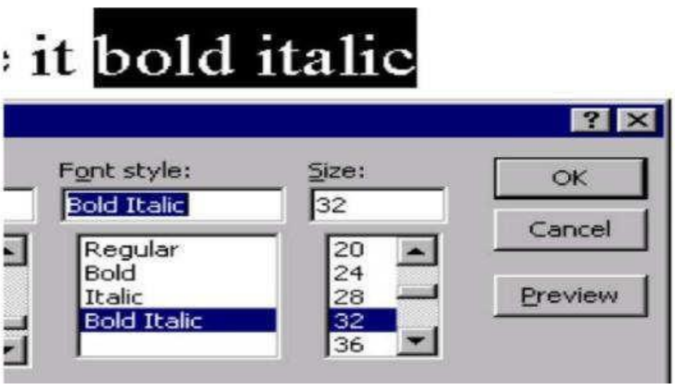


By behavior
This describes the action the toolkit arries out on your behalf and this can be co ntrolled. Examples are drawing and such interactions betwe en the widgets

themselves. But timing issues of this behavior should be watched such as the large selections under Windows applications.
See the pictorial example below.

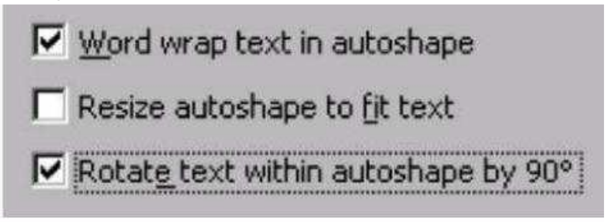


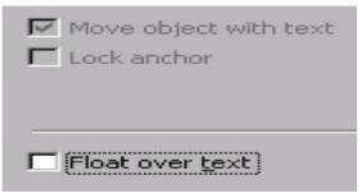
By semantics
Semantics are menus, buttons, etc th at do things as desired by the user..



An example is '... lets make it *bold italic*'

The semantics assignment is determined by the designer or user; YOU say what it means. The semantics is usually up to you.
Although widgets may link direct to database, even then, you say what links to the database.
So to choose the widget for the job, assign meaning first on what you want it to do, followed by appearance and then how you want it done.
You may for example want actions carried out through menu, buttons, or toolbar or you want to set the status of options using checkbox, radio button, or combi-box.





4.0 Conclusion

Designing interaction styles should be based on the following criteria:
Domain — this considers the area of work under study e.g. graphic design
Goal — what the designer wants to achieve e.g. create a solid red triangle
Task — how you go wish to present the style ultimately in terms of operations or actions, e.g. select fill tool, click over triangle
The style should be easy to focus on look and feel because if you want someone to do something, make it easy for them and understand their values.

5.0 Summary

Interaction styles are the nature of dialogs between the user and the system.
A graphical user interface (GUI) is a type of user interface which allows people to interact with electronic devices like computers, hand-held devices through graphical icons, and visual indicators by direct manipulation of the graphical elements.
In WIMP Interface, 'WIMP' stands for Windows, Icons, Menus and Pointers (or Windows, Icons, Mice, and Pull-down menus!)
It is a default style interface for majority of interactive computer systems, especially PCs and desktop machines
Widgets are bits that make the Graphical User Interface. They are the individual items on a Graphical User Interface (GUI). They can also be called the elements of the WIMP interface.
Menus are choice of operations or services offered on the screen
A button is an individual and isolated region within a display that can be selected to invoke an action
Toolbars are icons with fast access to common actions and are often customizable:
Palettes are little windows of actions shown or hidden via menu option in available shapes in drawing package
Dialogue boxes are information windows that pop up to inform of an important event or request information such as saving a file.

6.0 Tutor Marked Assignment

1. Mention 8 of the interaction styles available in HCI
2. Distinguish between the following pairs of interfaces
   - (i) Widgets and Graphical User Interface
   - (ii) Menus and Dialogue boxes
   - (iii) Drag — and- Drop and Copy or Cut and paste
   - (iv) Speech Recognition and Natural Language
   - (v) Command Line Interface and Direct manipulation Interface
   - (vi) Query Dialogue and Form- fills
   - (vii) Mouse gestures and handwriting recognition
   - (viii) Buttons and Palettes
3. Differentiate between the following types of widgets
   - (i) Widgets by appearance, (ii) Widgets by behavior, (iii) Widgets by semantics
4. What are keyboard accelerators?
5. In designing menus, what are the issues that should be considered?

7.0 Further Readings / References

- Myers, B.A., "All the Widgets." *SIGGRAPH Video Review*, 1990. 57
- Meyrowitz, N. and Van Dam, A., "Interactive Editing Systems: Part 1 and 2." *ACM Computing Surveys*, 1982. 14(3): pp. 321- 352.
- Koved, L. and Shneiderman, B., "Embedded menus: Selecting items in context." *Communications of the ACM*, 1986. 4(29): pp. 312- 318.
- Myers, B.A., "A Taxonomy of User Interfaces for Window Managers." *IEEE Computer Graphics and Applications*, 1988. 8(5): pp. 65-84.
- Andrew Sears and Julie A. Jacko (Eds.). (2007). Handbook for Human Computer Interaction (2nd Edition). CRC Press. ISBN 0-8058-5870-9
- Brooks, F. "The Computer "Scientist" as Toolsmith--Studies in Interactive Computer Graphics," in *IFIP Conference Proceedings.* 1977. pp. 625-6 34.

MODULE 3: DESIGNS OF HUMAN-COMPUTER INTERACTION

UNIT 1: DESIGN GUIDELINES, RULES AND PRINCIPLES

Table of contents

1.0 INTRODUCTION

This unit offers guidelines for design of user interface software in six functional areas: data entry, data display, sequence control, user guidance, data transmission, and data protection.

The guidelines are proposed here as a potential tool for designers of user interface software. Guidelines can help establish rules for coordinating individual design contributions; can help to make design decisions just once rather than leaving them to be made over and over again by individual designers. It can also help to define detailed design requirements and to evaluate user interface software in comparison with those requirements.

The design of user interface software will often involve a considerable investment of time and effort. Design guidelines can help ensure the value of that investment.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Select design guidelines
- Know how to monitor guidelines and carry out prototype tests
- Understand the concept of translating selected guidelines into design rules
- Know the importance of documenting design rules
- Explain various types of design rules
- Know how to use design rules
- Learn some design principles

.0 MAIN CONTENT

3.1 DESIGN GUIDELINES

Guidelines are more suggestive and general. There are two types of guidelines, they are:

i. Abstract guidelines or principles that are applicable during early design life cycle activities
ii. Detailed guidelines otherwise called style guides that are applicable during the later system life cycle activities

Understanding justification for guidelines aids in resolving conflicts

3.1.1 SELECTION OF DESIGN GUIDELINES

Not all of the guidelines can be applied in designing any particular system. For any particular system application, some of the guidelines will be relevant and some will not. Design guidelines must be generally worded so that they might apply to many different system applications. Thus generally-worded guidelines must be translated into specific design rules before they can actually be applied.

The process of selecting relevant guidelines for application and translating them into specific design rules is referred to here as "tailoring". Who will do this guidelines tailoring? It should be the joint responsibility of system analysts and human factors specialists assessing design requirements, of software designers assessing feasibility, and of their managers. It may also be helpful to include representatives of the intended system users in this process, to ensure that proposed design features will meet operational requirements.

Once all relevant guidelines have been identified, a designer must review them and decide which ones actually to apply.

There are two reasons why a designer might not wish to apply all relevant guidelines.

First, for any given application, some guidelines may conflict, and the designer must therefore choose which are more important.

Second, budgetary and time restrictions may force the designer to apply only the most important guidelines -- those that promise to have the greatest effect on system usability.

3.1.2 EXPERTISE EXPERIENCE VERSUS GUIDELINES

Guidelines cannot take the place of expertise experience.

An experienced designer, one skilled in the art, might do well without any guidelines. An inexperienced designer might do poorly even with guidelines. Few designers will find time to read an entire book of guidelines. If they do, they will find it difficult to digest and remember all of the material. If guidelines and/or the rules derived from guidelines are to be helpful, they must be kept continually available for ready reference.

Guidelines cannot take the place of expert design consultants, or at least not entirely. A design expert will know more about a specific topic than can be presented in the guidelines. An expert will know what questions to ask, as well as many of the answers. An expert will know how to adapt generally-stated guidelines to the specific needs of a particular system design application. An expert will know how to trade off the competing demands of different guidelines, in terms of operational requirements.

3.1. 3 MONITORING GUIDELINES AND PROTOTYPE TESTING

For maximum effectiveness, guideline tailoring must take place early in the design process before any actual design of user interface software. In order to tailor guidelines, designers must have a thorough understanding of task requirements and user characteristics. Thus task analysis is a necessary prerequisite of guidelines tailoring.

The result of guidelines application will be a design for user interface software that may incorporate many good recommendations. However, even the most careful design will require testing with actual users in order to confirm the value of good features and discover what bad features may have been overlooked. Thus prototype testing must follow initial design, followed in turn by possible redesign and operational testing.

Indeed, testing is so essential for ensuring good design that some experts advocate early creation of an operational prototype to evaluate interface design concepts interactively with users, with iterative design changes to discover what works best. But prototyping is no substitute for careful design. Prototyping will allow rapid change in a proposed interface; however, unless the initial design is reasonably good, prototyping may not produce a usable final design.

Considering the system development process overall, guidelines application will not necessarily save work in user interface design, and in fact may entail extra work, at least in the initial stage of establishing design rules. But guidelines application should help produce a better user interface. Because guidelines are based on what is known about good design, the resulting user interface is more likely to be usable. Certainly the common application of design rules by all designers working on a system should result in more consistent user interface design. And the single objective on which experts agree is design consistency.

3.1.4 TRANSLATION OF SELECTED GUIDELINES INTO DESIGN RULES

Because guidelines are intended for use on a variety of systems, they are worded in general terms. Before a guideline can actually be applied it must be translated into specific design rules. For instance, a guideline which states that displays should be consistently formatted might be translated into design rules that specify where various display features should appear, such as the display title, prompts and other user guidance, error messages, command entries, etc.

Any guideline can have different possible translations. A guideline which states that each display should be uniquely identified could be translated into a design rule that display titles will be bolded and centered in the top line of the display. Or it could be translated into a design rule that display titles will be capitalized in the upper left corner of the display.

What would happen if guidelines were not translated into design rules, but instead were given directly to interface designers? If designers do not decide as a group what design rules will be used, then each designer will decide separately in the course of applying guidelines. The result will surely be an inconsistent design.

.2 DESIGN RULES

3.2.1 Types of design rules

Rules based on principles:

These comprise abstract design rules, rules based on low authority, and those based on high generality

Rules based on standards

These are specific design rules from high authority but with limited application

Rules derived from guidelines

These are rules of lower authority but of more general application

3.2.2 DOCUMENTATION, IMPLEMENTATION AND EVALUATION OF DESIGN RULES

After design rules have been specified for each selected guideline, those rules should be documented for reference by software designers and others involved in system development. Documentation of agreed rules, subject to periodic review and revision as necessary, will help coordinate the design process. Documented rules can then be applied consistently for a given application. With appropriate modifications, rules adopted for one application might later be used for other applications.

In the course of design, it may be determined that a particular design rule cannot be used. Therefore, some means must be provided to deal with exceptions. If a design rule is not appropriate for one particular display, then an exception can be made by whoever has been appointed to make such decisions. But if a design rule cannot be implemented at all, perhaps due to other design constraints, then all designers for that particular system must be notified, and perhaps another design rule must be substituted.

Finally, after the design is complete, it must be evaluated against the original design requirements to ensure that all design rules have indeed been followed. To help in the exception process and in the evaluation process, it may be useful to assign different weights to the various rules, indicating which are more important than others. Such weighting will help resolve the trade-offs that are an inevitable part of the design process.

3.2. 3 USING DESIGN RULES

Since design rules suggest how to increase usability, they may differ in generality and authority, therefore there exist various standards that guarantee uniformity of application. Some of those existing standards are:

- The standards set by national or international bodies to ensure compliance by a large community of designers. These standards require sound underlying theory particularly on this slowly changing technology.
- Hardware standards: These are more common than software standards. They are of high authority and low level of detail
- ISO 9241 standards that define usability as the effectiveness, the efficiency and the satisfaction with which users accomplish tasks
- "Broad brush" design rules

- Useful check list for good design
- Better design using these than using nothing!
- Different collections e.g.
  - Nielsen's 10 Heuristics (see Chapter 9)
  - Shneiderman's 8 Golden Rules
  - Norman's 7 Principles

There are Golden rules and heuristics governing designs
These are regarded as "Broad brush" design rules that provide a useful check list for good design. A better design is achieved using these than using nothing!
The different collections include: the Nielsen's 10 Heuristics rules, the Shneiderman's 8 Golden Rules and the Norman's 7 Principles.

3.2.4 The Shneiderman's 8 Golden Design Rules
1. Strive for consistency
2. Enable frequent users to use shortcuts
3. Offer informative feedback
4. Design dialogs to yield closure
5. Offer error prevention and simple error handling
6. Permit easy reversal of actions
7. Support internal locus of control
8. Reduce short-term memory load

. 3 DESIGN PRINCIPLES

3. 3.1 The Norman's 7 Design Principles
1.Use both knowledge in the world and knowledge in the head.
2.Simplify the structure of tasks.
3.Make things visible: bridge the gulfs of Execution and Evaluation.
4.Get the mappings right.
5.Exploit the power of constraints, both natural and artificial.
6.Design for error.
7.When all else fails, standardize.

3. 3.2 DESIGN PRINCIPLES FORMULATED TO SUPPORT USABILITY :
Principle of Learnability : This is the ease with which new users can begin effective interaction and achieve maximal performance
Principle of Flexibility: These are the multiplicity of ways the user and system exchange information
Principle of Robustness: This is the level of support provided the user in determining successful achievement and assessment of goal-directed behaviour

The Principles of learnability are broken down into :
Predictability : This is determining effect of future actions based on past interaction history and its operation visibility
Synthesizability: This is assessing the effect of past actions, its immediate and its eventual honesty
Familiarity: This is how prior knowledge applies to new system and how easy one can guess its affordance Generalizability: This is extending specific interaction knowledge to new situations
Consistency: This concerns the likeness in input and output behaviour arising from similar situations or task objectives

Principles of flexibility comprise:
Dialogue initiative : This is the freedom from system imposed constraints on input dialogue and it compares the system against the user pre-emptiveness.
Multithreading: This is expressing the ability of the system to support user interaction for more than one task at a time. It also looks at the concurrent and interleaving multimodality.
Task migratability: This is passing responsibility for task execution between user and system
Substitutivity: This allows equivalent values of input and output to be substituted for each other. It compares representation multiplicity and equal opportunity
Customizability: This is the modifiability and adaptability of the user interface by user or the modifiability and adaptivity of the user interface by the system.

Principles of robustness are made up of:
Observability: This is the ability of the user to evaluate the internal state of the system from its perceivable representation. It considers the browsability, the defaults, the reachability, the persistence , and the operation visibility.
Recoverability: This concerns the ability of the user to take corrective action once an error has been recognized. It looks at the reachability, the forward and backward recovery and the commensurate effort.
Responsiveness:This is how the user perceives the rate of communication with the system and how stable is the response.
Task conformance: This explains the degree to which system services support all of the user's tasks, the task completeness and its adequacy.

4.0 Conclusion
The goal of interaction design is to design for maximum usability
Design rules comprise the principles of usability which look at the general understanding of the design, the standards and guidelines which set the direction for design, and the design patterns that capture and reuse design knowledge.
In designing computer-based information systems, special attention must be given to software supporting the user interface.

5.0 Summary
Guidelines are more suggestive and general. There are two types of guidelines; Abstract guidelines and detailed guidelines.
Understanding justification for guidelines aids in resolving conflicts.
Before a guideline can be applied, it must be translated into specific design rules. Those rules should be documented for reference by software designers and others involved in system development.
There are Golden rules and heuristics governing designs
For maximum effectiveness, guideline monitoring must take place early in the design process before any actual design of user interface software.
Guidelines cannot take the place of expertise experience.
There should be early creation of an operational prototype to evaluate interface design concepts interactively with users.
There are certain design principles of learnability, flexibility, and robustness that are formulated to support usability.

6.0 Tutor Marked Assignment
1. Mention those responsible for selecting relevant guidelines for application and translation into design rules. Indicate specific area of responsibility for each professional.
2. Why is it necessary to translate selected guidelines into design rules?

3. What are the advantages derivable from documenting design rules, why is the evaluation of the design necessary?
4. When is it most appropriate to monitor design guidelines and carry out prototype testing and why?
5. Distinguish between rules based on principles, those based on standards and those derived from guidelines.
6. What are those design principles formulated to support usability?
7. Mention any 6 of the Shneiderman's 8 Golden rules that govern interactive designs.

7.0 Further Readings / References

- C. Brown: Human-Computer Interface design guidelines. Ablex, 1989.
- W. Galitz: Handbook of screen format design. QED, 1989.
- C. Gram, G. Cockton (eds.): Design principles for interactive software. Capman & Hall, 1996.
- D. Hix, R. Hartson: Developing user interfaces. Wiley, 1993.
- ISO 9241 (Part 10: Dialogue principles, Part 12: Presentation of information, Part 14: Menu dialogues, Part 15: Command dialogues, Part 16: Direct manipulation dialogues, Part 17: Form fill-in dialogues)
- D. Mayhew: Principles and guidelines in software user interface design. Prentice, 1992.

UNIT 2: EVALUATION METHODS

Table of contents

1.0 INTRODUCTION

Evaluation tests usability and functionality of system and can be carried out in the laboratory, in the field and/or in collaboration with users.

The evaluation technique which covers both design and implementation should be considered at all stages in the design life cycle.
The goals of evaluation are to assess extent of system functionality, to assess effect of interface on user, and to identify specific problems from design and implementation.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Express your understanding of the three main evaluation techniques viz a viz the cognitive walkthrough, the heuristic, and the review-based evaluations.
- Master available methods of evaluating interaction design and its implementation
- Select the best evaluation methods among the options available

.0 MAIN CONTENT

3.1 Evaluation Techniques

The Evaluation Design Techniques include:
The Cognitive Walkthrough
The Heuristic Evaluation
The Review-based evaluation
User Participation

3.1.1 Cognitive Walkthrough

This technique was proposed by Polson et al It evaluates design on how well it supports user in learning task and is usually performed by expert in cognitive psychology.
The design expert 'walks through' the design to identify potential problems using psychological principles with forms used to guide the analysis
For each task, the walkthrough considers what impact the interaction will have on the user, the cognitive processes required and the learning problems that may occur.
The analysis focuses on goals and knowledge to establish whether the design leads the user to generate the correct goals.

3.1.2 Heuristic Evaluation

This was proposed by Nielsen and Molich. Here, usability criteria (heuristics) are identified; the designs are examined by experts to see if these are violated.
Examples of heuristics include:
Testing whether the system behaviour is predictable
Testing whether the system behaviour is consistent
Testing whether a feedback is provided
Heuristic evaluation 'debugs' design.

3.1. 3 Review-based evaluation

This evaluation reviews results from the literature that are used to support or refute parts of the design. However, care is needed to ensure the results are transferable to new design.
It is a model-based evaluation which in addition encompasses cognitive models that can be used to filter design options
An example is the GOMS prediction of user performance.
The design rationale can also provide useful evaluation information

3.1.4 Evaluating through user Participation

Evaluation could be carried out in two ways; 1. through laboratory studies and 2. through the field studies

Laboratory studies
Laboratory studies are appropriate if system location is dangerous or impractical for constrained single user systems to allow controlled manipulation of use
The advantages of carrying out laboratory studies are appropriate specialist equipment available and are utilised in an uninterrupted environment.
Disadvantages could be lack of context and difficulty in observing several users cooperating.

Field Studies approach
This approach is appropriate where context is crucial for longitudinal studies
The advantages are that studies are carried out in a natural environment where context of evaluation is retained. Though observation may alter such context. Advantageously, longitudinal studies are also possible.
Obvious disadvantages are that there could be distractions and noise particularly in which the location of the field is within a public place.

3.2 Evaluating Implementations
To evaluate implementations, the evaluator uses artefacts such as simulation, prototypes and the full implementation.

3.2.1 Experimental evaluation
This is a controlled evaluation of specific aspects of interactive behaviour. Here the evaluator chooses the hypothesis to be tested with a number of experimental conditions considered different only in the value of some controlled variable.
The changes in behavioural measure are attributed to the different conditions.

Experimental evaluation factors
The following eexperimental factors are given consideration:
Subjects: This identifies who the representative is, and the measure of the sufficient sample for the experiment.
Variables: These are the things to modify and measure
Hypothesis: This considers what you would like to show
Experimental design: This looks at how you are going to do it

Variables experimental factors
There are two variables; Independent variable (IV) and Dependent variable (DV)
The independent variable characteristics are changed to produce different conditions. Examples of the characteristics include the interface style and number of menu items.
The dependent variable (DV) characteristics are those measured in the experiment. Examples of such characteristics include the time taken and number of errors.

Hypothesis experimental factor
This is a prediction of outcome framed in terms of IV and DV. For example, "error rate will increase as font size decreases".
For example ,if the null hypothesis states that there is no difference between conditions, our aim is to disprove this
e.g. A null hypothesis may be stated that there is "no change with font size". So we disprove.

Experimental design factors
Within groups design: Here each subject performs experiment under each condition.

The advantage here is that transfer of learning is made possible. It is also less costly and less likely to suffer from user variation.
Between groups design: Each subject here performs under only one condition hence there is no transfer of learning. Also more users are required and variation can bias results.

3.2.2 Analysis of data
It is necessary that before you start to do any statistics, you have to look at the data and you have to save the original data.
The choice of statistical technique depends on type of data and the information required.
Type of data : This could either be discrete, that is, comprising finite number of values or continuous, comprising any value.

Analysis - types of test
Parametric test:
This assumes a normal distribution and it is robust and powerful.
Non-parametric test:
This does not assume a normal distribution. It is less powerful but more reliable
Contingency table test:
This classifies data by discrete attributes. It counts number of data items in each group.
The information required is to establish whether there is a difference and how big the difference is. It also seeks to establish how accurate the estimate is.
However, parametric and non-parametric tests are used to mainly establish whether there is a difference.

3.2. 3 Experimental studies on groups
These are more difficult than single-user experiments.
These studies identify problems associated with subject groups, choice of task, the data gathering and the analysis of the data gathered.

Some of the problems identified with subject groups are:
The larger the number of subjects the more expensive the experimental design becomes. It also takes a longer time to 'settle down'. It creates an even more variation that makes it difficult to adhere to timetable.
Therefore, only three or four groups are recommended.
The tasks involved in experimental studies on groups are the needs to encourage cooperation among the groups through the use of multiple channels.
The options that may be adopted are:
Creative task such as *writing a short report on a particular experiment*
Decision games such as desert survival task games modelling a decision phenomenon
Control task such as demonstrated in a particular firm.

3.2.4 The Data gathering and Analysis processes
This can be done using several video cameras with direct logging of application data.
The problems with data gathering are synchronisation of data and the sheer volume of data required.
The one solution to this is to record from each perspective.

Analysis of data
Because of the vast variation between groups,
Carry out experiments within groups
Conduct a micro-analysis such as gaps in speech.

Conduct anecdotal and qualitative analysis and look at interactions between group and media. Realise that controlled experiments may 'waste' resources!

3. 3 Field studies

In field studies, experiments are dominated by group formation but are more realistic because:
There is a distributed cognition with the work studied in context
The real action is a situated action having both the physical and social environment being crucial.
Contrast:
Psychology — controlled experiment
Sociology and anthropology comprises open study and rich data

3. 3.1 Observational Methods

These involve the following:
Think Aloud
Cooperative evaluation
Protocol analysis
Automated analysis
Post-task walkthroughs

Think Aloud method
The user observed performing task as he is asked to describe what he is doing and why, what he thinks is happening etc.
The advantages of this method are :-
It is simple and requires little expertise
It can provide useful insight
It can show how system is actually in use
The disadvantages are:-
It is subjective and selective
The act of describing may alter task performance

Cooperative evaluation method
This is a variation on think aloud. The user collaborates in evaluation such that both the user and the evaluator can ask each
other questions throughout.
Additional advantages here are that:-
It is less constrained and easier to use
The user is encouraged to criticize the system
Clarification is possible between the user and collaborator

Protocol analysis method
This requires paper and pencil; it is therefore cheap and limited to writing speed.
It uses audio that is good for think aloud but difficult to match with other protocols.
It uses video that enables an accurate and realistic analysis but needs special equipment. It is obtrusive.
Other analysis tools involve computer logging that is automatic and unobtrusive in which large amounts of data may be difficult to analyze
It requires a user notebook that is coarse and subjective, providing useful insights and good for longitudinal studies.
However, mixed use of these tools is carried out in practice.
The audio or video transcription is difficult and requires skill. Some automatic support tools are similarly available.

Automated analysis
This is a workplace project involving a post task walkthrough where the user reacts on action after the event. It is used to fill in intention.
Advantages
The analyst has time to focus on relevant incidents
It helps avoid excessive interruption of task
Disadvantages
There is a lack of freshness
There may be post-hoc interpretation of events

Post-task walkthroughs
Here transcript is played back to participant for comment
The advantages are that the response to transcript playback is immediate and is fresh in mind.
The evaluator has time to identify questions and hence useful to identify reasons for actions and alternatives considered
It is mostly necessary in cases where think aloud is not possible.

3. 3.2 Query Techniques
Query technique comprises Interviews and Questionnaires

Interviews
The analyst questions the user on one-to -one basis and is usually based on prepared questions.
The interviews are informal, subjective and relatively cheap to conduct.
The advantages are:
It can be varied to suit context
Issues can be explored more fully
It can elicit user views and identify unanticipated problems
The disadvantages are that it is very subjective and time consuming.

Questionnaires
In this method, set of fixed questions are given to users.
The advantages are that it is quick and reaches large user group.
It can be analyzed more rigorously.
The disadvantages are
It is less flexible and less probing
There is a need for a careful design on what information is required and how answers are to be analyzed.
Styles of question are: general, open-ended, scalar, multi-choice, and ranked.

3. 3. 3 Physiological methods
These comprise Eye tracking and Physiological measurement

Eye tracking
With this method, the head or desk mounted equipment tracks the position of the eye. The eye movement reflects the amount of cognitive processing a display requires.
Measurements include
Fixations: Here, the eye maintains a stable position. The number and duration of measurements indicate level of difficulty with display
Saccades: In this case, there is a rapid eye movement from one point of interest to another.

Scan paths: This involves moving straight to a target with a short fixation at the target being optimal.

Physiological measurements

In physiological measurement, the emotional response is linked to physical changes which may help determine a user's reaction to an interface.

The measurements include:

Heart activity, including blood pressure, volume and pulse.

Activity of sweat glands such as in Galvanic Skin Response (GSR)

Electrical activity in muscle called electromyogram (EMG)

Electrical activity in brain called electroencephalogram (EEG)

However, some difficulties are always experienced in interpreting these physiological responses; therefore, more research is required in this area.

4.0 Conclusion

Guides towards choosing an evaluation method comprise:

Commencement of evaluation process: Design versus Implementation

Style of evaluation: Laboratory versus Field

Nature of evaluation: Subjective versus Objective

Type of measures: Qualitative versus Quantitative

Level of information: High level versus Low level

Level of interference: Obtrusive versus Unobtrusive

Resources available: Time, Subjects, Equipment and Expertise

5.0 Summary

Cognitive Walkthrough evaluates design on how well it supports user in learning task and is usually performed by expert in cognitive psychology.

In heuristic evaluation, usability criteria (heuristics) are identified and the designs are examined by experts to see if these are violated.

Review-based evaluation reviews results from the literature which are used to support or refute parts of design.

User participation evaluation is carried out through laboratory studies and field studies

Experimental evaluation is a controlled evaluation of specific aspects of interactive behaviour by choosing the hypothesis to be

tested with a number of experimental conditions.

Analysis of data is done through parametric test, non-parametric test, and contingency table test of data.

Experimental studies on groups identifies problems associated with subject groups, choice of task, the data gathering and the analysis of the data gathered.

Observational methods involve think aloud, cooperative evaluation, protocol analysis, automated analysis, and post-task walkthroughs

Query technique comprises Interviews and Questionnaires

Physiological methods of evaluation comprise Eye tracking and Physiological measurements.

6.0 Tutor Marked Assignment

1. What is the purpose of carrying out evaluation tests? Enumerate the available techniques used in carrying out the evaluation.
2. What is the objective of the "Cognitive walkthrough" and how is it carried out?
3. Provide 3 examples of Heuristics

4. Explain the two ways by which evaluation is carried out through user participation
5. Carrying out evaluation through the laboratory and field studies has some obvious advantages and disadvantages. What are they?
6. Describe the 4 experimental factors to consider while carrying out an experimental evaluation.
7. Describe 3 types of tests that can be carried out for analyzing data
8. "Experimental studies on subject groups are more difficult than single-user experiments". What are the specific problems associated with subjects groups to justify this statement?
9. Describe the Query techniques of evaluating design. What are their advantages and disadvantages?
10. Distinguish between the "Think Aloud" and Cooperative observational methods of evaluating designs?
11. Describe the physiological methods employed to evaluate interactive design.

7.0 Further Readings / References

- J. Dumas, J. Redish: A practical guide to usability testing. Ablex, 199 3.
- D. Freedman, G. Weinberg: Walkthroughs, Inspections, and technical reviews. Dorset, 1990.
- ISO 9241 (Part 11: Guidance on usability, Part 1 3: User guidance)
- Monk, P. Wright, J. Haber, L. Davenport: Improving your Human-Computer Interface: a practical technique. Prentice Hall, 199 3.
- J. Nielsen, R. Mack (ed.): Usability inspection methods. Wiley, 1994.
- D. Norman: The psychology of everyday things. Basic Books, 1988.

UNIT 3: PARTICIPATORY DESIGN

Table of contents

1.0 INTRODUCTION

Participatory design is an approach to design that attempts to actively involve the end users in the design process to help ensure that the product designed meets their needs and is usable.

In participatory design, end-users are invited to cooperate with researchers and developers during a system interaction design process. They participate during several stages of the design process such as in the initial exploration and problem definition both to help define the problem and to focus ideas for solution. During development, they help evaluate proposed solutions.
Participatory design can be seen as a move of end-users into the world of researchers and developers, while a move of researchers and developers into the world of end-users is known as empathic design. This unit looks at both as necessarily participatory design.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Know the meanings of user centered design, distributed participatory design, and ethnography
- Describe the characteristics of Participatory Design(PD)
- Understand Hybridism and the Third Space concept
- Explain the diversity of Participatory Design techniques operating in third spaces
- Appreciate the unresolved issues in Participatory Design

.0 MAIN CONTENT

3.1 Major Concepts of Participatory Design:

User-design versus User-centered design:
There is a very significant differentiation between user-design and User-centered design.
There is an emancipatory theoretical foundation and systems theory bedrock on which user-design is founded.
In user-centered design, users are taken as centers in the design process, consulting with users heavily, but not allowing users to make the decisions, nor empowering users with the tools that the experts use. For example, most of the internet documentation and information content are user-designed. Users are given the necessary tools to make their own entries. While users are allowed to propose changes or have input on the design, a smaller and more specialized group decide about features and system design.

Ethnography and Participatory design
In participatory design, workers enter into design context while in ethnography (as used for design), the designer enters into work context. Both make workers feel valued in design and encourages workers to 'own' the products. The user is an active member of the design team.

Participatory design in software development
This is the user involvement in design, with more emphasis on the involvement of a broad population of users rather than a small number of user representatives.
Many groups and projects apply participatory design research methods on a regular basis, and, hence, are part of the development and appropriation of the methods, as well as of disseminating the methods to industrial practice.

Distributed participatory design
Distributed Participatory design (DPD) is a design approach and philosophy that supports the direct participation of users and other stakeholders in system interaction analysis and design work. Nowadays design teams most often are distributed, which stress a need for support and knowledge gathered from design of distributed systems. Distributed Participatory design aims to facilitate understanding between different stakeholders in distributed design teams by giving each the opportunity to engage in hands-on activities.

Ethics
The ethics involve a participatory socio-technical approach devised by Mumford.
It states that the system development is about managing change and that non-participants are more likely to be dissatisfied.
There are three levels of participation: consultative, representative, and consensus.
Design groups including stakeholder representatives make design decisions and job satisfaction is the key to solution

3.2 CHARACTERISTICS OF PARTICIPATORY DESIGN

Participatory design is context and work oriented rather than system oriented. It is collaborative and iterative.
Hence the unit focuses on participatory practices that share these attributes, including (a) site-selection of PD work; (b) workshops; (c) story collecting and story telling through text, photography, and drama; (d) games for analysis and design; (e) the correlation of descriptive and functional prototypes and (f) brainstorming, pencil and paper exercises
Participatory design (PD) is a set of theories, practices, and studies related to end users as full participants in activities leading to software and hardware computer products and computer based activities
The field is extraordinarily diverse, drawing on fields such as (a) user-centered design, (b) graphic design, (c) software engineering,
(d) architecture, (e) public policy,(f) psychology, (g) anthropology, (h) sociology, (i) labor studies,(j) communication studies, and (k) political science.
Researchers and practitioners are brought together—but are not necessarily brought into unity—by a pervasive concern for the knowledge, voices, and rights of end users, often within the context of software design and development, or of other institutional settings (e.g., workers in companies, corporations, universities, hospitals, and governments).
This unit primarily addresses methods, techniques, and practices in participatory design, with modest anchoring of those practices in theory with the involvement of software professionals and the end users.

3. 3 HYBRIDITY AND THE THIRD SPACE CONCEPTS OF PARTICIPATORY DESIGN

This is concerned with participatory methods that occur in the hybrid space between software professionals and end users.
Why is this hybrid space important? An influential argument was made that the border or boundary region between two domains, or two spaces, is often a region of overlap or hybridism— that is a "third space" that contains an unpredictable and changing combination of attributes of each of the two bordering spaces. In such a hybrid space, enhanced knowledge exchange is possible, precisely because of those questions, challenges, reinterpretations, and renegotiations.
These dialogues *across differences* and *within differences*—are stronger when engaged in by groups, emphasizing not only a shift from assumptions to reflections, but also from individuals to collectives.

Guides and Expectations of Hybridism or Third Space Concept
There is an overlap between two or more different regions or fields (inbetweenness)
It is not "owned" by any reference field but partaking of selected attributes is done in reference fields.
Potential site of conflicts exist between or among reference fields, hence questioning and challenging of assumptions are unavoidable
Mutual learning and synthesis of new ideas are core benefits when agreed working language among the participants is ensured.
When working assumptions and dynamics are ensured, understandings, relationships and collective actions emerge while dialogues across and within different disciplines exist.

Considerations in the design process:

What is considered to be data are posers
The rules of evidence may become challenges to overcome
How are the conclusions drawn become issues to be resolved
Reduced emphasis on authority and increased emphasis on interpretation are norms
Reduced emphasis on individualism and increased emphasis on Collectivism result in Heterogeneity.

Organizations comprise multiple constituencies each with their own professional identities and views of others. By contrast, the methods
allow for the creation of new perspectives and new locations, and they acknowledge the possibility that each participant can make different choices at different moments about where to locate his or her perspective, standpoint, and thus, accountability.
There is a need for "a new set of skills and competencies that go beyond technical design skills to create conditions that encourage a collaborative design process and active reflection for working with groups. These push on the traditional boundaries between the users and designers"

A large part of the design process, especially in large-scale projects and organizations involving several actors, is not dedicated to analytical work to achieve a solution but mostly to efforts at reconciling conflicting [conceptual] frames or at translating one frame into another. Much work of the designer is concerned with defining collectively what the relevant problem is and how to evaluate such problem.

3.4 PARTICIPATORY DESIGN IN HCI SOFTWARE DEVELOPMENT

Participatory design desires to bridge the gap between two spaces—the world of the software professionals, and the world of the end users. Each world has its own knowledge, practices and well-defined boundaries. Movement from one world to the other is known to be difficult. This difficulty is manifested in our elaborate methods for requirements analysis, design, and evaluation—and in the frequent failures to achieve products and services that meet users' needs and/or are successful in the marketplace.

Traditional scientific practice in HCI has focused on instruments and interventions that can aid in transferring information between the users' world and the software world. Most of the traditional methods are relatively one-directional; for example, we analyze the requirements *from* the users, we deliver a system *to* the users, and we collect usability data *from* the users.
While there are many specific practices for performing these operations, relatively few of them involve two-way discussions, and fewer still afford opportunities for the software professionals to be surprised—*to learn something that we didn't know we needed to know*.
The PD tradition has, from the outset, emphasized mutuality and reciprocity—often in a hybrid space that enabled new relationships and understandings.
"The mutual validation of diverse perspectives": Floyd (1987) analyzed software practices into two paradigms, which she termed product-oriented (focused on the computer artifact as an end in itself) and process-oriented (focused on the human work process, with the computer artifact as means to a human goal). In her advocacy of balancing these two paradigms, Floyd noted that the process-oriented paradigm required mutual learning among users and developers

Most of PD theories and practices require the combination of multiple perspectives—in part, because complex human problems require multiple disciplines (e.g., software expertise and work-domain expertise) for good solutions and in part because the workplace democratic tradition reminds us that all of the interested parties should have a voice in constructing solutions methods suitable for a software professional's organization with *concrete* methods suitable for work with end users.
Muller and colleagues elaborated on this taxonomic dimension by asking *whose work domain serves as the basis for the method*

At the *abstract* end of the continuum, the users have to enter the world of the software professionals in order to participate—e.g., rapid prototyping. At the *concrete* end of the continuum, the software professionals have to enter the world of the users in order to participate; for example, ethnography and end-user "design" by purchasing software for small companies

"What about the practices that did not occur at the *abstract* or *concrete* end-points of the continuum? *What about the practices in between?*" These practices turn out to occur in an uncertain, ambiguous, overlapping disciplinary domain that does not "belong" to either the software professionals or the end users (e.g., these practices occur in neither the users' turf nor the software professionals' turf). The practices in between the extremes are hybrid practices, and constitute the third space of participatory design.

3.5 NEGOTIATION, SHARED CONSTRUCTION, AND COLLECTIVE DISCOVERY IN PD AND HCI

This describes a diversity of participatory design techniques, methods, and practices that provide hybrid experiences or that operate in intermediate third spaces in HCI. Because the theme is hybridism, these descriptions are organized in terms, strategies, and moves that introduce novelty, ambiguity, and renewed awareness of possibilities, occurring at the margins of existing fields or disciplines.

A storytelling method provides a space in which people negotiates the naming and defining of workplace activities

3.5.1 Site Selection

One of the simplest parameters that can be manipulated to influence hybridism is the site of the work. There are two approaches to participatory design: (1) Bring the designers to the workplace or (2) Bring the workers to the design room at a site different from the work place.

The selection of the site can be important in a discussion of participatory architectural practice,

Work place site selection

Being in a foreign environment and with other users, users will tend to take a more general view of things; however, when collaborating with users *in their work context*, users tend to feel more at ease as they are on their home ground—the designers are the visitors. Tools and environment are physically present and easy to refer to. This makes for a conversation grounded in concrete and specific work experiences. The idea was born to create a type of design event with activities in both environments and with two sets of resources to support design collaboration.

New site selection

In terms of hybridism, the selection of site can be a deliberate strategy to introduce new experiences and perspectives to one or more parties in the design process—a de-centering move that can bring people into positions of ambiguity, renegotiation of assumptions, and increased exposure to heterogeneity.

Site selection initially appears to be a matter of *moving across the boundary* between different work cultures, rather than *living within the boundary*. The use of *common design practices across sites*, however, makes those practices (and the membership of the design group) into a kind of movable third space. The practices and the group membership become stable features that persist across multiple sites. At the same time, the practices, and even the membership grow and evolve with exposure to new sites and new understandings. In these ways, the practices become an evolutionary embodiment of the knowledge of the learning of the group

Benefits of using new site:

· Improved learning and understanding.

It is a move from"symmetry of ignorance" toward "symmetry of knowledge" as diverse parties educate one another through a "symmetry of learning"—and even a kind of "transformation" through exposure to new ideas.

The selection of site can also lead to the strengthening of the voices that were comfortable at each site.
· Greater ownership. The procedures could strengthen the user involvement in their projects. There would also be increases in commitment and ownership of the evolving knowledge and design of the group.

3.5.2 Workshops

Workshops may serve as another alternative to other site selection. Workshops are usually held to help diverse parties ("interested parties" or "stakeholders") communicate and commit to shared goals, strategies, and outcomes (e.g., analyses, designs, and evaluations, as well as workplace-change objectives).
Workshops are often held at sites that are in a sense neutral—they are not part of the software professionals' workplace, and they are not part of the workers' workplace.
More importantly, workshops usually introduce novel procedures that are not part of conventional working practices. These novel procedures take people outside of their familiar knowledge and activities, and must be negotiated and collectively defined by the participants. Workshops are thus a kind of hybrid or third space, in which diverse parties communicate in a mutuality of unfamiliarity, and must create shared knowledge and even the procedures for developing those shared knowledge.
The best-known workshop format in PD is the Future Workshop

A Future Workshop proceeds through three stages: (a) *Critiquing* the present, (b) *Envisioning* the future, and (c) *Implementing*, or moving from the present to the future. These three activities involve participants in new perspectives on their work, and help to develop new concepts and new initiatives.
A number of workshops have focused on simple materials and informal diagrams, rather than on formal notations. The tools are simple diagrams or drawings with no special formalisms because staff members participating in the workshops, as well as those to whom the results are later presented, typically have no experience.
Using technical descriptions, a workshop is described as a family of "generative tools" of activities that are selectively combined into strategic design, under an overall conceptual strategy that combines market research ("what people say"), ethnography ("what people do"), and participatory design ("what people make").
The activities include the construction of collages focused on thinking (e.g., "how do you expect your work to change in the future?"), mapping (e.g., laying out an envisioned work area on paper), feeling ("use pictures and words to show a health-related experience in your past"), and story telling.

A type of storyboarding workshop format is described as that in which people create narratives using photographs, putting them in sequences and in many cases altering (typically through the addition of speech bubbles to show what people were thinking or doing).
The various workshop approaches have several commonalities. Each workshop brings together diverse participants to do common work, to produce common outcomes and to develop a plan of joint action They are thus opportunities that require mutual education, negotiation, creation of understanding, and development of shared commitments. Each workshop takes place in an atmosphere and often in a site that is not "native" to any of the participants. Thus, all of the participants are at a disadvantage of being outside of their own familiar settings, and they must work together to define their new circumstances and relationships. The combination of diverse voices leads to syntheses of perspectives and knowledge.
Benefits. Advantages claimed for this type of hybridism include:
· Development of new concepts that have direct, practical value for product design
· Engagement of the interested parties ("stakeholders") in the process and outcome of the workshop.
· Combinations of different people's ideas into unified concepts.
· Production of artifacts that are the expected and useful "inputs" to the next stage of the development process

.5. 3 NARRATIVE STRUCTURES

Stories and Story telling

Stories and story telling have played a major role in ethnographic work since before there was a field called "HCI". . Stories have also had an important history in HCI.

Stories in participatory work may function in at least three ways. First, they may be used as triggers for conversation, analysis, or feedback. Second, they may be told by end users as part of their contribution to the knowledge required for understanding product or service, opportunities, and for specifying what products or services should do. Third, they may be used by design teams to present their concept of what a designed service or product will do, how it will be used, and what changes will occur as a result

Hypermedia technologies can be utilized to enable communities tell their own stories with the intention that "plurality, dissent, and moral space can be preserved. It enables multiple authors reuse community materials selectively, telling different stories within a common context.

The different accounts were organized according to themes, and laid out spatially on the image of a fictitious island for navigation by end users.

The work enters several areas or aspects of hybridism. First, the authors of the stories (e.g., community members) were using hypermedia technology for the first time, and were thus in the role of learners, even while they were the owners of the stories, and were thus in the role of experts. Second, the authors wrote from their own perspectives, which were sometimes in strong conflict with one another. Third, the authors could make use of one another's materials, effectively moving away from single-author narratives and into a kind of collaborative collage of materials, which conveyed interlinked stories.

Fourth, just as the community members were negotiating and defining their roles as learner-experts, the software professionals/researchers were negotiating and defining their roles as expert's facilitators-students.

Using Paper and Pencil to tell stories

A second line of practice and research has emphasized end users telling their stories using a system of paper-and-pencil, card-like templates. The earliest version was the Collaborative Analysis of Requirements and Design (CARD) technique later developed into a more general tool

The card-based practices used pieces of cardboard about the size of playing cards. Each card represents a component of the user's work or life activities, including user interface events (e.g., screen shots), social events (conversations, meetings) and cognitive, motivational, and affective events (e.g., the application of skill, the formation of goals or strategies, surprises and breakdowns, evaluations of work practices). The cards were used by diverse teams in analysis, design, and evaluation of work and technology. Because the cards were novel object to all the participants, they occasioned third-space questionings and negotiations, resulting in new shared understandings and co-constructions. Often, teams used the cards to prepare a kind of story board, narrating the flow of work and technology used and annotating or innovating cards to describe that work. The resulting posters formed narratives of the work that were demonstrated to be understandable to end users, corporate officers, and software professionals, and which led to insights and decisions of large commercial value

Using Photographs for story telling

Stories can be told in many ways. One approach that has informed recent PD work is end-user photography through (a) taking pictures and (b) organizing pictures into albums. These activities allow end users to enter into a kind of native ethnography, documenting their own lives. In keeping with the issues raised in the preceding "Stories" section, it is important that the informants themselves (the end users) control both the camera and the selection of images . They thus become both authors and subjects of photographic accounts of their activities. This dual role leads to one kind of hybridity, in which the photographic activities partake of both the world of common social life, and the world of documenting and reporting on working conditions.

Photo documentaries were used as a means of providing familiar, concrete artifacts to enable design collaborations. Photo documentaries are used as one component of a set of user-composed diary techniques, with a subsequent user created collages to serve as a rich source of discussions.
End-user photography is an interesting case of hybridity and the production of third spaces. Photography is a good example of an "in-between" medium—one that is part of many people's informal lives but that is also an intensively studied medium of communication and argumentation. Photography occurs at the margin of most people's work, and yet can easily be incorporated into their work.
Discussions around the photographs, and combination of the photographs into photo narratives or collages can lead to mutual learning and new ideas, particularly through the inclusion of the voices of the photographers, the viewers, and especially the people depicted in the photographs
Benefits. The use of end-user photographs appears to be new and experimental, and there are few strongly supported claims of benefits. Informal claims of success and contribution include the following:

- Richer, contextualized communication medium between end users and designers. (In some cases, the designers were not, themselves, software professionals.)
- Stronger engagement of designers with end-users' worlds.
- Enhanced sharing of views and needs among end users, leading to stronger articulation by them as a collective voice.

The informants should make their own decisions about what was important, and therefore what they should photograph.

Dramas and Videos
Drama provides another way to tell stories—in the form of theatre or of video. One of the important tensions with regard to drama in PD is the question of whether the drama is considered a finished piece, or a changeable work-in-progress.
Many PD drama-practitioners make reference to Boal's Theatre of the Oppressed. Boal described theatrical techniques whose purpose was explicitly to help a group or a community find its voice(s) and articulate its position(s).
The most influential of Boal's ideas was his Forum Theatre, in which a group of nonprofessional actors performs a skit in front of an audience of interested parties. The outcome of the skit is consistent with current events and trends—often to the dissatisfaction of the audience. The audience is then invited to become authors and directors of the drama, changing it until they approve of the outcome.
Changes in work patterns and work-group relations were acted out by software professionals in the end-users' workplace, using cardboard and plywood prototypes, in anticipation of new technologies, the workers served as the audience, and critiqued the envisioned work activities and working arrangements. The drama was carried out iteratively, with changes, until it was more supportive of the skilled work of the people in the affected job titles. The researchers made repeated visits with more detailed prototypes, again using the vehicle of a changeable drama, to continue the design dialogue with the workers. This work was widely credited with protecting skilled work from inappropriate automation, and leading to a product that increased productivity while taking full advantage of workers' skills.

Muller et al. (1994) presented a related tutorial demonstration piece called Interface Theatre, with the stated goal of engaging a very large number of interested parties in a review of requirements and designs (e.g., in an auditorium). In Interface Theatre, software professionals acted out a user interface "look and feel" using a theatrical stage as the screen, with each actor playing the role of a concrete interface component.
Dramatic approach brings a strong overlap of the world of end users and the world of software developers, showing concrete projections of ideas from one world into the other world—and, in most uses, allowing modification of those ideas. Drama is marginal to the work domains of most software professionals and most end users, and thus moves all parties into an ambiguous area where they must negotiate meaning

and collaboratively construct their understandings. Agreements, conflicts, and new ideas can emerge as their multiple voices and perspectives are articulated through this rich communication medium.

Benefits

- Building bridges between the worlds of software professionals and users.
- Enhancing communication through the use of embodied (e.g., acted-out) experience and through contextualized narratives.
- Engaging small and large audiences through direct or actor-mediated participation in shaping the drama (influencing the usage and design of the technology).
- Increasing designers' empathy for users and their work.
- Simulating use of not-yet-developed tools and technologies to explore new possibilities.
- Fuller understanding by focus group members, leading to a more informed discussion.

3.6 GAMES

From theory to practice, the concept of games has had an important influence in participatory methods and techniques.

Ehn's theoretical work emphasized the negotiation of language games in the course of bringing diverse perspectives together in participatory design. In this view, part of the work of a heterogeneous group is to understand how to communicate with one another.

The work of heterogeneous teams is, in part, the "mutual validation of diverse perspectives"

Games have been an important concept in designing practices, with the convergent strategies of enhanced teamwork and democratic work practices within the team.

When properly chosen, games can serve as levelers, in at least two ways. First, games are generally outside of most workers' jobs and tasks. They are therefore less likely to appear to be "owned" by one worker, at the expense of the alienation of the non-owners. Second games are likely to be novel to most or all of the participants. Design group members are more likely to learn games at the same rate, without large differences in learning due to rank, authority, or background. This in turn can lead to greater sharing of ideas. In addition, games can help groups of people to cohere together and communicate better.

One of the purposes of games is enjoyment of self and others—and this can both liven a project and build commitment among project personnel.

"Design-by-playing" approach, introducing several games into PD practice: Examples include:

- Specification Game, a scenario-based game based on a set of "situation cards," each of which described a workplace situation.

Players (members of the heterogeneous analysis/design team) took turns drawing a card and leading the discussion of the work situation described on the card.

- Organization Kit and Desktop Publishing Game,

in which cards illustrating components of work or outcomes of work were placed on posters, with annotations.

- CARD, a card game for laying out and/or critiquing an existing or proposed work/activity flow
- PICTIVE, a paper-and-pencil game for detailed screen design
- Icon Design Game, a guessing game for innovating new ideas for icons (this game assumes subsequent refinement by a graphic designer).
- Interface Theatre, for design reviews with very large groups of interested parties

The games emphasize hands-on, highly conversational approaches to discussing both the user interface concept itself and the work processes that it was intended to support..

The Technology Game adds simple shapes that stand for technologies, again playing those shapes onto the work environment in the Landscape Game.

Finally, the Scenario Game moves back to the real world, enacting possibilities based on new ideas from the preceding three games. The enactments may be video recording, both for documentary purposes and to generate further video material for another cycle of the four games.

Each of these games would take all of its players outside of their familiar disciplines and familiar working practices, but strategically reduced the anxiety and uncertainty of the situation by using the social scaffolding of games. Each game requires its players to work together through mutual learning to understand and define the contents of the game, and to interpret those contents to one another in terms of multiple perspectives and disciplines. The conventional authority of the software professionals was thus replaced with a shared interpretation based on contributions from multiple disciplines and perspectives.

Benefits. Participatory design work with games has been claimed to lead to the following benefits:

- Enhanced communication through the combination of diverse perspectives.
- Enhanced teamwork through shared enjoyment of working in a game-like setting.
- Greater freedom to experiment and explore new ideas through flexible rules and redefinition of rules during the game.
- Improved articulation of the perspectives, knowledge, and requirements of workers.
- New insights leading to important new analyses and designs with documented commercial value.

3.7 CONSTRUCTIONS

Preceding sections have considered hybridism in participatory activities, such as site selections, workshops, stories, photography, dramas, and games. This section continues the survey of participatory practices that bring users and software professionals into unfamiliar and ambiguous "third space" settings.

Collaborative construction of various concrete artifacts comprising:

- Low-tech prototypes for analysis and design.
- Cooperative Prototyping

Low-Tech Prototypes that includes participatory prototyping:

Low-tech prototypes may lead to "third space" experiences because they bring people into new relationships with technologies—relationships that are "new" in at least two important ways. First, the end users are often being asked to think about technologies or applications that they have not previously experienced. Second, in *participatory* work with low-tech prototypes, end users are being asked to use the low-tech materials to reshape the technologies—a "design-by-doing" approach

In this way, participatory work with lowtech prototypes involves much more user contribution and user initiative than the more conventional use of "paper prototypes" as surrogates for working systems in usability testing

The UTOPIA project provided impressive demonstrations of the power of low-tech cardboard and plywood prototypes to help a diverse group to think about new technologies, office layouts, and new working relations that might result from them.

Benefits. The low-tech participatory prototyping approaches benefits include:

- Enhanced communication and understanding through grounding discussions in concrete artifacts.
- Enhanced incorporation of new and emergent ideas through the ability of participants to express their ideas directly

via the low-tech materials, and through the construction of artifacts that can be used in other techniques, especially drama and video documentaries.

- Enhanced working relations through a sense of shared ownership of the resulting design.
- Practical application with measured successes in using low-tech design approaches to real product challenges, achieving consequential business goals.

Cooperative Prototyping

This last section on participatory methods is concerned with software prototyping.

The potential of cooperative prototyping in several projects, using different technology infrastructures led to enhanced communication
with end users, improved incorporation of end-user insights into the prototypes, and stronger collective ownership and collective action planning by the team. Also observed is the time consuming breakdowns in the design process itself, when new ideas required significant programming effort.
In a different prototyping approach, a system is delivered to its end users as series of iterative prototypes, each of which gradually
adds functionality
What appears to be critical is that the prototype functions as a *crucial artifact* in the end-users' work, such as,
(a) a resource of documents for librarians
(b) an online event checklist that served as the crucial coordination point for the work of diverse contributions or (c) a database supporting funding work in a nonprofit organization . Trigg (2000) provided a series of observations and tactical recommendations about how to engage the users in the evaluations that both they and the software professionals had agreed were needed.
This very brief survey of cooperative prototyping and "iterative delivery" approaches shows several aspects of hybridity. In the case of cooperative prototyping, the cooperative work may be done in a physical third space that is neither the end-users' office nor the software developers' office
In the case of the delivery of iterated prototypes, each prototype is presented in the end users' setting, but is unusual and only partially functional, and thus occasions reflection about its nature, its role in the end users' work, and, ultimately, the work itself. In both cases, the invitation (or perhaps the necessity) of the end-users' actions to help shape the technology becomes an important means of refocusing
their attention, as well as the attention of the software developers. The ensuing conversations are concerned with the interlinked feasibility of changes to technology and to work practices, with attributes of hybridity including polyvocal dialogues, challenging one another's assumptions, and developing plans for collective actions.
Benefits. Some of the virtues of the low-tech prototyping approaches have also been claimed for the cooperative prototyping and "iterative delivery" approaches as follow:

- Enhanced communication and understanding through grounding discussions in concrete artifacts.
- Enhanced working relations through a sense of shared ownership of the resulting design.

Additional claims for software-based prototypes include:

- Earlier understanding of constraints posed by the practical limitations of software.
- Improved contextual grounding of the design in the end-users' work practices.

3.8 Brainstorming

The most well-known idea generation technique is brainstorming, introduced by Osborn (1957). His goal was to create synergy
within the members of a group: ideas suggested by one participant would spark ideas in other participants. Subsequent studies
challenged the effectiveness of group brainstorming, finding that aggregates of individuals could produce the same number of ideas as groups. They found certain effects, such as production blocking, free riding, and evaluation apprehension, were sufficient to outweigh the benefits of synergy in brainstorming groups. Brainstorming, is an important group-building exercise for participatory design; designers may brainstorm ideas by themselves.
Brainstorming in a group is more enjoyable and, if it is a recurring part of the design process, plays an important role in helping group members share and develop ideas together.
The simplest form of brainstorming involves a small group of people. The goal is to generate as many ideas as possible on a pre-specified topic: quantity, not quality, is important.
Brainstorming sessions have two phases: the first for generating ideas and the second for reflecting upon them. The initial phase should last no more than an hour. One person should moderate the session,

keeping time, ensuring that everyone participates, and preventing people from critiquing each other's ideas. Discussion should be limited to clarifying the meaning of a particular idea.
A second person records every idea, usually on a flipchart or transparency on an overhead projector.
After a short break, participants are asked to reread all the ideas and each person marks their three favorite ideas.
One variation is designed to ensure that everyone contributes, not just those who are verbally dominant. Participants write their ideas on individual cards or Post-it notes for a prespecified period. The moderator then reads each idea aloud.
Authors are encouraged to elaborate (but not justify) their ideas, which are then posted on a whiteboard or flipchart.
Group members may continue to generate new ideas, inspired by the others they hear.

Another variant of brainstorming, called "video brainstorming" is a very fast technique for prototyping interaction: instead of simply writing or drawing their ideas, participants act them out in front of a video camera. The goal is the same as other brainstorming exercises, i.e., to create as many new ideas as possible, without critiquing them. However, the use of video, combined with paper or cardboard mock ups, encourages participants to experience the details of the interaction and to understand each idea from the perspective of the user, while preserving a tangible record of the idea.
Each video brainstorming idea should take two to five minutes to generate and capture, allowing participants to simulate a wide variety of ideas very quickly. The resulting video clips provide illustrations of each idea that are easier to understand and remember than hand-written notes.
Video brainstorming requires thinking more deeply about each idea than in traditional oral brainstorming. It is possible to stay vague and general when describing an interaction in words or even with a sketch, but acting out the interaction in front of the camera forces the author of the idea and the other participants to consider seriously the details of how a real user would actually interact with the idea. Video brainstorming also encourages designers and users to think about new ideas in the context in which they will be used. Video clips from a video brainstorming session, even though rough, are much easier for the design team to interpret than written ideas from a standard brainstorming session.
Unlike standard brainstorming, video brainstorming encourages even the quietest team members to participate.

.9 UNRESOLVED ISSUES IN PARTICIPATORY DESIGN:

· Participation by non-organized workforce. The field of PD has long been concerned about how to engage in meaningful participative activities with workers or others who are not organized into a group with collective bargaining power or other collective representation.

· Evaluation and metrics. One of the weaknesses of the literature on participatory practices is the dearth of formal evaluations.
There is a small set of papers that have examined software engineering projects across companies, and have found positive outcomes related to end-user participation
There are no formal experiments comparing participatory methods with non-participatory methods in a credible workplace context. Such studies would be difficult to perform, because they would require that a product be implemented and marketed twice (once with participation, and once without).
The problem is made more difficult because measurements and metrics of organizational outcomes, user participation, and user satisfaction are currently vexing research issues

4.0 CONCLUSION

Participatory design (PD) is a set of theories, practices, and studies related to end users as full participants in activities leading to software and hardware computer products and computer based activities

Hybridism is at the heart of PD, fostering the critical discussions and reflections necessary to challenge assumptions and to create new knowledge, working practices, and technologies. When we consider HCI as a set of disciplines that lie between the space of work and the space of software development, we see that the hybrid third spaces developed within PD have much to offer HCI in general.

5.0 SUMMARY

In user-centered design, users are taken as centers in the design process, consulting with users heavily. In participatory design, workers enter into design context while in ethnography; the designer enters into work context. Both make workers feel valued in design and encourages workers to 'own' the products.

Participatory design in software development is the user involvement in design,

Distributed Participatory design (DPD) is a design approach and philosophy that supports the direct participation of users and other stakeholders in system interaction analysis and design work.

The ethics involved in the participatory socio-technical approach devised by Mumford, states that the system development is about managing change and that non-participants are more likely to be dissatisfied. There are three levels of participation: consultative, representative, and consensus.

Design groups including stakeholder representatives make design decisions and job satisfaction is the key to solution

6.0 Tutor Marked Assignment

1. Differentiate between the following pairs of terms

User design and User Centered design

Ethnography and Participatory design

2(a) What are the benefits derivable from a distributed participatory design exercise?

(b) Briefly describe the characteristics of the Participatory design

3. What do you understand as the "Third Space Concept" and "Hybridism" in participatory design?

4. In participatory design exercise, the designers are either brought to the workplace or the workers are brought to the design room at a different site from the work place. Briefly itemize the various benefits accruing from selecting any of the options.

S. Describe the workshops of participatory design as an alternative to other site selection. What are the obvious benefits of this alternative arrangement on participatory design?

6. What are the advantages of stories and story telling in participatory design?

7.0 Further Readings / References

- Grudin, J. (1993). Obstacles to Participatory Design in Large Product Development Organizations: Schuler, D. & Namioka, A. (1993). Participatory design: Principles and practices. Hillsdale, NJ: Erlbaum.
- Kensing, F. 2003. Methods and Practices in Participatory Design. ITU Press, Copenhagen, Denmark.
- Kensing, F. & Blomberg, J. 1998. Participatory Design: Issues and Concerns In Computer Supported Cooperative Work, Vol. 7, pp. 167-185.

UNIT 4: SYSTEM INTERACTIVE DESIGN PATTERNS

Table of contents

1.0 INTRODUCTION

A pattern is an invariant solution to a recurrent problem within a specific context.
An HCI design pattern is an approach to reusing knowledge about successful design solutions Patterns do not exist in isolation but are linked to other patterns in *languages* which enable complete designs to be generated

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Know the guides for developing effective design patterns
- Understand the processes of design
- Describe various screen designs and layout
- Know how to design acceptable presentation
- Understand Prototyping

.0 MAIN CONTENT

.1 CHARACTERISTICS OF PATTERNS

The characteristics of patterns include the following:

- Capturing the design practice and not the theory
- Capturing the essential common properties of good examples of design
- Representing design knowledge at varying levels of social, organisational, and conceptual framework
- Embodying values and expressing what is humane in interface design

- Patterns are intuitive and readable and can therefore be used for communication between all stakeholders
- A pattern language should be generative and assist in the development of complete designs.

.2 GUIDES AT DEVELOPING EFFECTIVE DESIGN PATTERNS

3.2.1 Commencement of design process:
The human interaction designer would commence his design process by asking the following questions:
The design:

- What is the design all about?
- What are the interventions?
- What are the goals?
- What are the constraints?

The design process

- What happens when?

The Users

- Who are the users?
- What are their likes and dislikes on interactivity?

Navigation of Interaction

- How does the user find his way around a system?

.2.2 Design Considerations:

Scenarios of interaction
Part of the scenarios is a probe of rich stories relating to design issues that include users' experiences and expectations.

Iteration and prototypes
Remember that the designer never get it right the first time!

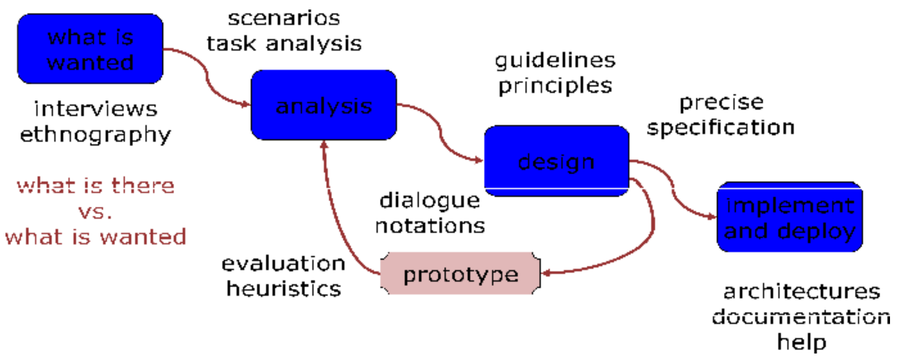
Interactions and Interventions
The designer should design interactions not just interfaces and not just the immediate interaction because technology changes
The designer should design interventions not just artefacts and not just the system, but also related documentation such as manuals and tutorials.

What is design?
Design is achieving goals within constraints, so the design should consider those to benefit from the goals and for what purpose.
The design should consider the constraints in terms of materials and platforms and the corresponding trade-offs.
The Golden rule of design is for the designer to understand his materials for Human—Computer Interaction
Understanding materials means understanding computer's limitations, its capacities, its tools and platforms.
It also means understanding people, their psychological and social aspects.
The design should consider the possibility of human error and their interaction, since to err is human

'To err is human' example:
Example of accident reports on air crash, industrial accident, hospital mistake, reveal the enquiry blaming 'human error' on all. But concrete lintel breaks because of too much weight applied on it so the blame goes to 'lintel error' not design errorsince we know how concrete behaves undestress.
So human 'error' is normal and we kow how users behave under stress hence design for it!
Treat the user at least as well as phy ical materials! The Central focus is the user.

. 3 THE DESIGN PROCESSES

. 3.1 The design life cycle



Explanation of processes in the diagr am:
Requirements: This is identifying wh t is there and what is wanted
Analysis: This is on ordering and und erstanding
Design: This concerns what to do an how to decide on what to do
Interaction and prototyping: Means etting it right and finding what is really nee ed!
Implementation and deployment: M aking it happen and delivering

To carry out the above, consider The limited time available as a desig The trade-off usability: Find out problems that may work against the ultimate usage of the designed interaction and ensure such problem(s) are tackled r ight from the onset.
Remember that a perfect system is o ne that was badly designed; so do not expect to design a perfect system. Systems are dynamic.

3. 3.2 User focuses

It is essential that you know your use rs, their personality and cultural probes.
Talk to those class of users, watch th em and use your imagination of their percep tion of the system they want.
Innovate a 'user' model not necessar ily a real person but carry out a prototype o a system.

Design's cultural probes
Cultural probes can be carried out as follows:

- By direct observation though sometimes hard; for example on psychiatric patients

- By giving out probe packs that consist of items to prompt responses. These are given to people to open in their own environment and to record what is meaningful *to them*. These probe packs are used to inform interviews, prompt ideas and en-culture designers
- Gathered stories for the design are used and reused
- By communicating with others to validate models and understand dynamics
- Finding out what will users want to do by step-by-step walkthrough on what they can see using sketches, screen shots etc or what they are used to doing e.g manipulating keyboard and mouse, etc.
- Find out their thinking on the proposed interaction design
- Explore the depths by exploring interaction to determine what happens when
- Explore cognition to determine the users thinking
- Explore architecture of the system to determine what is happening inside
- Use particular scenarios to communicate with other designers, clients and users
- Validate other models by comparing them with your models
- Express dynamics through screenshots appearances and scenario behaviours.
- Use several scenarios and use several methods since scenarios provide one linear path through system design,

An example of a personality probe for a design.
Sola is 37 years old, She has been the Warehouse Manager for five years and worked for an Engineering company for twelve years. She didn't go to the university, but has studied in her evenings for a business diploma. She has two children aged 15 and 7 and does not like to work late. She did part of an introductory in-house computer course some years ago, but it was interrupted when she was promoted and could no longer afford to take the time. Her vision is perfect, but her right-hand movement is slightly restricted following an industrial accident 3 years ago. She is enthusiastic about her work and is happy to delegate responsibility and take suggestions from her staff. However, she does feel threatened by the introduction of yet another new computer system (the third in her time at the Engineering company).

3. 3. 3 Navigation design
Within the local structure, utilise a single screen
Within the global structure, utilise a whole site

Levels of design to guide the designer include:-
- Widget choice level containing menus, buttons etc.
- Screen design level
- Application navigation design
- Environment design level that comprises other applications and operating systems.

Example of a web design:
A web interaction design comprises:
- The widget choice level containing elements and tags e.g <a href = li…. " >
- The screen design such as page design
- The application navigation design such as site structure
- The environment design such as the web, the browser, and external links

The physical devices interaction design comprises:
- The widget choice level comprising the controls such as buttons, knobs and dials
- The screen design such as the physical layout
- The application navigation design such as the modes of device

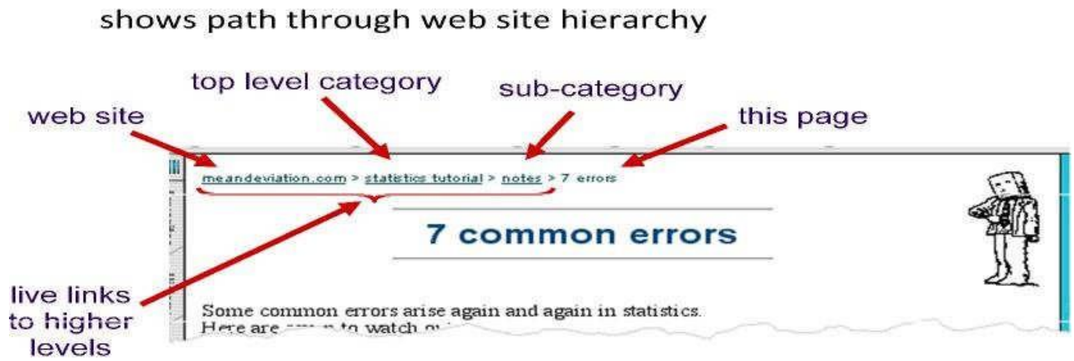
- The environment design such as the real world

Structure of design should be viewed from the following platforms:

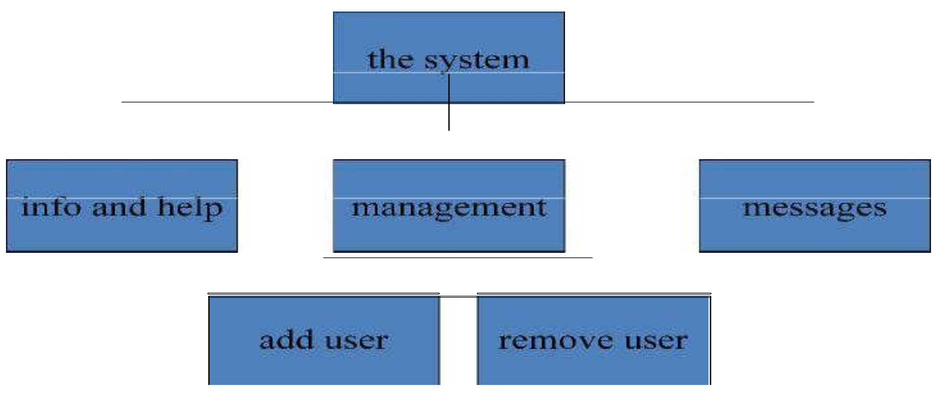
- Within a screen
- Locally looking from one scre en looking out
- Globally from the structure o the site and movement between screens within application
- And wider still, consider relat ionship with other applications

The four golden rules of the design a re;
Knowing where you are
Knowing what you can do
Knowing where you are going or wha t will happen at
Knowing where you have been or wh you have done

Example of where you are of a Web ite address:



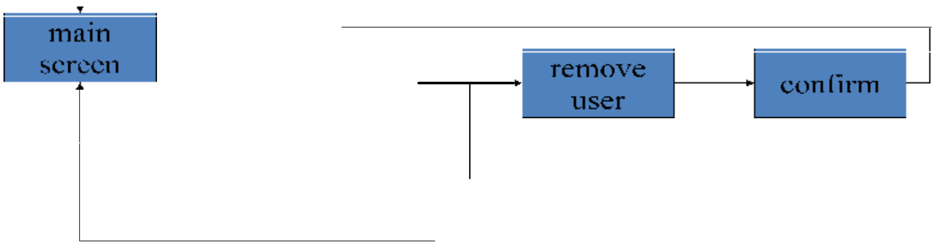
The following Hierarchical diagram s hows parts of application with screens or gro ups of screens typically showing a
Functional separation.



Drawing navigating hierarchies enables short term memory but not the menu sizeen
It also shows many items on each scr e. een with the items structured within the scr

Dialogue in User Interaction design i raction between users and system but details differ each ti a computer dialogue having patterns of int e

Network diagrams show different pa ths through system including branches that re more task oriented than the hierarchy.
It shows the relationship between ap plications and beyond. See illustration belo :



Network diagram shows what leads t onal branches and is
more task oriented than a hierarchic
When considering style issues, it ide
Network diagrams also identify (i) fu navigation issues such as
embedded applications and links to

ign and layout are the grouping of data thatthe screen displays, the a
Use of white space to separate group of dat is also recommended as

r is doing, he should think of what informat ion should be displayed,
parisons. The design should ensure that for m follows function on

o what, what happens when, it shows functi al
diagram.
tify platform standards and consistency
nctional issues such as cut and paste and (ii)
ther applications such as the web browsing.

.4 SCREEN DESIGNS AND LAYOU T:

3.4.1 Principles of design

The basic principles in the screen des
structure, order and their alignment.
shown in the diagrams below.
The designer should ask what the us
the order of display and possible com
display.

.4.2 Grouping and Structure desig n

From the diagram above, one can dis tinguish items that are logically together as against those physically together.

Billing details:
Name
Address: …
Credit card no

Delivery details:
Name
Address: …
Delivery time

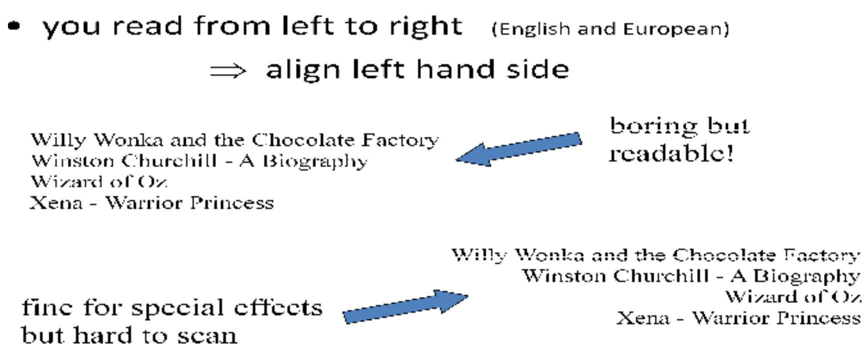
Order details:

item	quantity	cost/item	cost
size 10 screws (boxes)	7	3.71	25.97
……	…	…	…

The order of groups and items is imp ortant and should be considered in the desi n.
Think about what is a natural order; the order should match the screen order!
Use boxes, space etc and set up the ight tabbing.

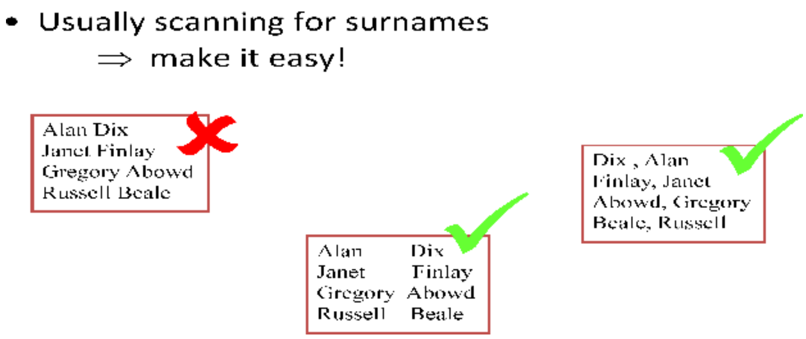
Decoration: Using boxes to group logical items
Use fonts for emphasis and headingsbut these should not be too many!!
Consider the best method to align items; for example, you can separate items usi ng white spaces as in the example above.

.4. 3 Alignment of text



⇒ align left h nd side

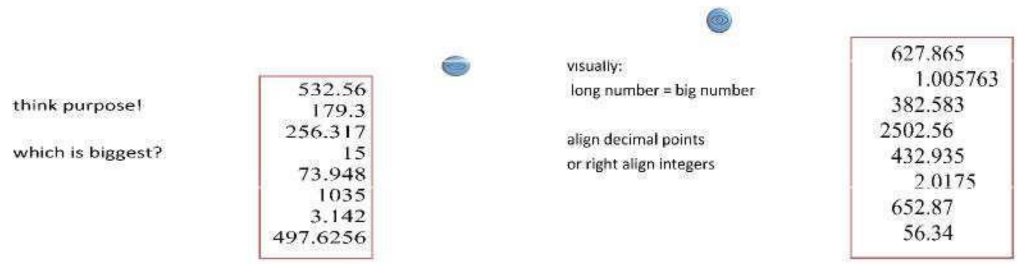
Alignment of names



From the three boxes illustrated above, the alignment of names in the first box does not enable easy recognition or identification of surna mes from the first name. The second and thi rd boxes enable easy recognition of surnames. The secon box has its names separated by white spac es and the third by commas.

Alignment of numbers
While aligning numbers, think of the purpose such alignment would serve. Consi er the biggest and the smallest numbers.
Usually the longest and/or the bigge t numbers appear immediately feasible to t he eye gaze.
Align decimal numbers properly eith er left or right. Right align integers (number s without decimals)

Examine the illustrations below and observe the most feasible.



In multiple columns tables as in the above illustrations, scanning across gaps bet ween lines of data is hard. It is particularly harder with table of data containing large database fields.

sherbert	75
toffee	120
chocolate	35
fruit gums	27
coconut dreams	85

sherbert	75
toffee	120
chocolate	35
fruit gums	27
coconut dreams	85

sherbert	75
toffee	120
chocolate	35
fruit gums	27
coconut dreams	85

sherbert	75
toffee	120
chocolate	35
fruit gums	27
coconut dreams	85

Graying to highlight

Scanning across gaps hard here Use leaders alignment ca n help visuals

In multiple columns, use leaders (lin s that link data with their field names or des cription). See above can To highlight data within a table, you light grey out the data. Do not colour, ot herwise the data would he be covered from being feasible. See last table above.

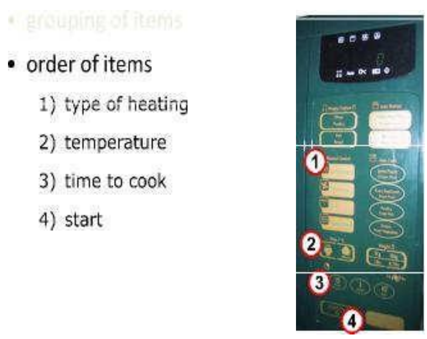
Use space to highlight

Use space to structure

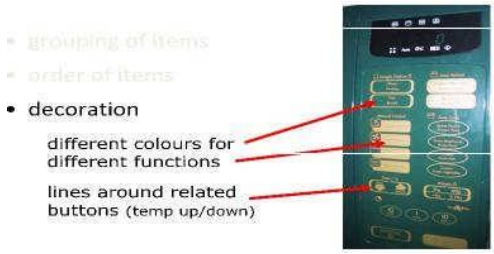
Observing the above tables of data, ne can see that spaces can be used to separ ate data, it can be used to structure and highlight as well.

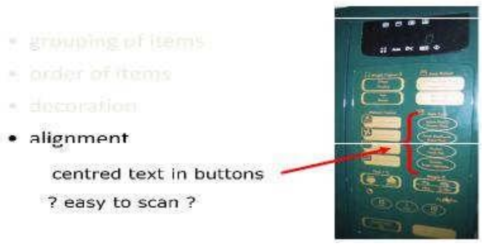
.5 PRESENTATION AND PHYSICA L CONTROLS OF DATA

5.1 Grouping and Ordering of Ite ms



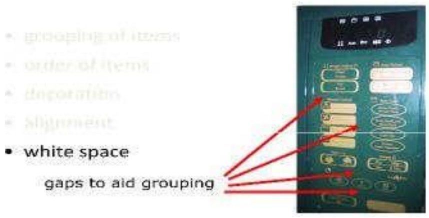
Here items of same functions are gro uped together Here i tems are displayed in order of priority of functions





Here different colours are used to di ferentiate functions here with lines around related buttons

Tex are centered in buttons



Gaps aid grouping here

Use white space within gaps to aid g ouping and proper alignment as in the abov e illustration.

5.2 Forms and dialogue boxes

In designing forms and dialogue box s, the designer should pay attention to presentation of the form and dialogue box. He should also conside r how data would be entered into the form.He should consider the importance of similar layout issues c oncerning the form and dialogue boxes suchas alignment and the label lengths.

In presenting an effective logical layo ut, the designer should use task analysis, ap propriate groupings , a natural order for entering informatio n such as from top to bottom, left to right (depending on the culture adopted) and setting tab order for ke yboard entry.

Look at the illustration below:

The box or form in the middle (The second) presents an acceptable arrangement

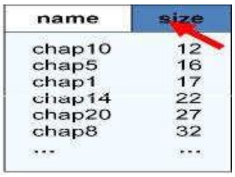
The designer should indicate which a nd what area is active and passive such as w here the user should click and where to type.

gns

The styles used should be consistentsuch as in web underlined links.
The labels and icons should have standards for common actions. The language us ed in the labels should be bold and should represent the curre t state or action.

.5. 3 Creating 'affordances' in desi

The word 'affordances' is a psycholo ical term used for physical objects. The sha pes and sizes of the objects suggest actions to be taken o n the object. Actions such as pick up, twist a nd throw. So in a user interface terminology, one can say th at buttons 'afford' pushing depending on th eir state.
For screen objects, the button-like o ject 'affords' mouse click while the physical -like objects suggest use. There is a culture of computer langu ge use such as icons 'afford' clicking or eve double clicking.



The designer should design appropri te appearance for interface objects to pres nt accurate information. Aesthetics, utility, colour and 3D feat ures could be added for appropriate usability.
In presenting information, purpose atters. The purpose would enable the desi ner determine the sort order, in which column would the da ta be input and whether it is numeric or alp abetic. See the table of figures above.
The designer should consider using t ext or diagram to make his presentation effe ctive. He can also consider presenting the information using graphs such as scatter graph or histog am.
In every decision on presentation, th e designer should use the paper presentatio n principles but add interactivity. An example is a 'dancin g histogram'.

5.4 Aesthetics and Utility

Aesthetically pleasing designs increa e user satisfaction and improve productivity . Beauty and utility may however sometimes conflict.
Mixed up visual styles make present tion easy to distinguish.
Clean design and little differentiation leads to confusion on the part of the user. or example, backgrounds behind text may be good to look at, ut hard to read. Look at the illustration abo ve. Both

can however work together if carefully done, as d monstrated in the design of the counter in onsumer products presentation.

5.5 Using Colour and 3D in pr sentation.

Both Colour and 3D effects are often used very badly!
In using colour, the designer should remember that older monitors have limited alette of colours while modern monitors have millions of colours to manipulate hence colour is over use d because of its abundance.
The designer should also beware of colour blindness as a result of using too man colours. Colours should be used sparingly in order to reinforce other information.
3D effects are good for physical information and some graphs but if over used as in text in perspective and 3D pie charts, it can blur Information .

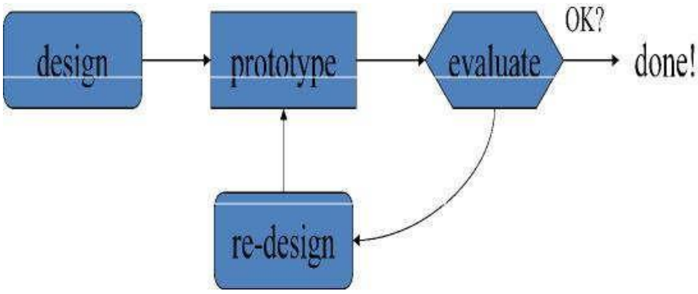
Bad Use of Colour

Example of Bad Use of Colour

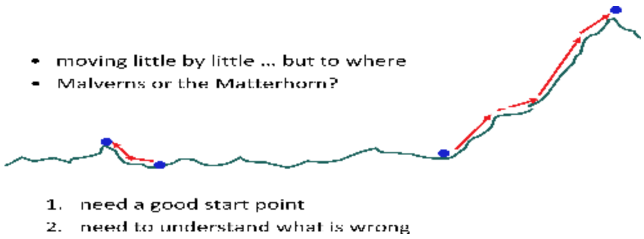
A bad use of colour is an over use of colour without very good reason (e.g. kids' site) which may eventually cause colour blindness .
Poor use of contrast as occurs when you do adjust your set! For example, when you adjust your monitor to greys only, you may not be able to re ad your screen.
Across countries and cultures, there is localisation and internationalisation that require changing interfaces for particular cultures and languages .
In globalisation, when you try to cho ose symbols etc. that work everywhere, you simply change language and use 'resource' database instead of literal text. But changes are required on sizes and left-right order, etc.

3.6 PROTOTYPING

Prototyping is an essential compone t of interactive system design.
Prototypes may take many forms, fr m rough sketches to detailed working proto types. They provide concrete representations of design i eas and give designers, users, developers a d managers an early glimpse into how the new system wil l look and feel. Prototypes increase creativity, allow early evaluation of design ideas, help designers think through, solve design problems, and suppo rt communication within multidisciplinary design teams.
Prototypes are concrete and are not abstract; hence provide a rich medium for e xploring a design space. They suggest alternate design paths and reveal important details about particula r design decisions. They force designers to be creative and to articulate their design decisions. Prototypes embody design ideas and encourage designers to confront the r differences of opinion.
The precise aspects of a prototype o fer specific design solutions and designers c an decide to generate and compare alternatives. The imprecise or incomplete aspects of a prototype highlig ht the areas that must be refined or require additional ideas.



In prototyping you never get it right irst time; if at first you don't succeed then c ontinue ...



Pitfalls of prototyping are
Moving little by little ... but to where . These pitfalls are avoided by Malverns or t e Matterhorn rules that say that
1. the designer needs a good start point and 2. needs to understand what is rong

4.0 CONCLUSION

Design is an active process of workin g with a design space, expanding it by gener ating new ideas and contracting as design choices are ma de. Designing effective interaction is difficult and many interactive systems (including many websites) h ave a good look but a poor feel.
The quality of interaction is tightly lin ked to the end users and a deep understand ing of their work practices. Designers must take the c ntext of use into account when designing th e details of the interaction.
Prototypes are flexible tools that he p designers envision a design space, reflect upon it, and test their design decisions. Prototypes are dive rse and can fit within any part of the design process, from the earliest ideas to the final details of the desig . Prototypes provide one of the most effect ive means for designers to communicate with each other, as we ll as with users, developers, and managers, t hroughout the design process.

5.0 SUMMARY

The processes of design comprise re uirements, analysis, the design itself, Intera ction and prototyping, Implementation and deployment.
The basic principles in screen design and layout are the grouping of data that the screen displays, the structure, order and their alignment.
In designing forms and dialogue box s, the designer should pay attention to pres entation and purpose of presentation.
'Affordances' is a term used to refle t on the shapes and sizes of physical object that suggest actions to be taken on the object.
Aesthetically pleasing designs increa e user satisfaction and improve productivity while mixed up visual styles make presentation easy to distinguish.
The designer should beware of colour blindness resulting from using too many co lours. Colours should be used sparingly in order to reinforce other information.
A bad use of colour is an over use of colour without very good reason and it may eventually cause colour blindness .
3D effects are good for physical infor mation and some graphs but if over used ca n blur Information.
Prototypes provide concrete represe ntations of design ideas and give designers, users, developers and managers an early glimpse into how the new system will look and feel.

6.0 Tutor Marked Assignment

1. Define the term Pattern, articularly as it relates to Human Computer Interaction (HCI)
2. Mention any four of the s ix characteristics of design patterns

. For effective commencement of the design process, the designer should ask himself certain questions as guides, what are those likely areas of question?

4. What are those issues that are likely to be considered by the designer during his design process?
5. Describe the design lifecycle of a typical design pattern
6. What are the major user focuses and cultural probes that can guide the pattern designer in his design process?
7. (i) Produce the hierarchical diagram that relates functional parts of applications with their groups of screens. What are the advantages of this diagram to designer?
   (ii) Draw a network diagram that shows the relationship between applications. What are the significances of the network diagram when considering design style issues?
8. Describe the basic principles governing screen design and layout
9. What are the concepts the designer should consider for effective presentation and physical controls of data? How do you relate these concepts specifically to the design of forms and dialogue boxes?
10. What do you understand by the term "Affordances in designing"? Do you agree that the designer should use "Affordances" concepts in his design process, and why?

(a) Why do designers have to include aesthetics, utilities, and 3D effects in their design patterns?

(b) What are the negative implications of over applying these effects in the designs?

11. Itemize the benefits of prototyping interactive System designs

7.0 Further Readings / References

- Gamma, Erich, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software.* Reading, MA: Addison-Wesley, 1995.
- Alexander, Christopher, Sara Ishikawa, Murray Silverstein, Max Jacobsen, Ingrid Fiksdahl-King, and Shlomo Angel. *A Pattern Language.* New York: Oxford University Press, 1977.
- Norman, Donald. *The Design of Everyday Things.* New York: Basic Books, 1988.
- Cypher, Allen. "Eager: Programming Repetitive Tasks By Example." In *Proceedings of CHI '91*, ACM, 1991.

MODULE 4: DESIGN OF USER AND SYSTEM INTERFACES

UNIT 1: DESIGN OF USER INTERFACE CONCEPTS

Table of contents

1.0 INTRODUCTION

This unit offers guidelines for design of user interface software in six functional areas: data entry, data display, sequence control, user guidance, data transmission, and data protection.

The guidelines are proposed here as a potential tool for designers of user interface software.

Guidelines can help establish rules for coordinating individual design contributions and can also help to make design decisions just once rather than leaving them to be made over and over again by individual designers. They can help define detailed design requirements and to evaluate user interface software in comparison with those requirements.

The design of user interface software will often involve a considerable investment of time and effort. Design guidelines can help ensure the value of that investment.

In designing computer-based information systems, special attention must be given to software supporting the user interface.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Explain the concept of User Interface Design
- Understand User System Interface
- Know the significance of User Interface
- Explain the function of User Interface Software
- Describe the interaction design phases
- Understand the concept of formative evaluation

.0 MAIN CONTENT

3.1 USER INTERFACE DESIGN OF INFORMATION SYSTEMS

Computers today are used for a broad range of applications. User interface design guidelines cannot be applied usefully in every case. Some computers may be embedded as components in larger systems, so that they communicate only with other computers and not directly with human users. When there is no user interface, then no user interface design guidelines are needed.

To the extent that information systems support human users performing defined tasks, careful design of the user-system interface will be needed to ensure effective system operation. The guidelines are intended to improve user interface design for such information systems.

Users of information systems interact with a computer in order to accomplish information handling tasks necessary to get their jobs done. They differ in ability, training and job experience. They may be keenly concerned with task performance, but may have little knowledge of (or interest in) the computers themselves. Design of the user-system interface must take account of those human factors.

3.1.1 USER-SYSTEM INTERFACE

User-system interface is broadly defined to include all aspects of system design that affect system use. Hence we are concerned with the user interface to computer-based information systems, i.e., with those aspects of system design that influence a user's participation in information handling tasks.

It focuses on those design features of the user interface that are implemented via software (i.e., the design of computer program logic) rather than hardware (the design of equipment). The guidelines are worded in terms of the functions that a user must perform, and the functional capabilities that a designer should provide, rather than the particular physical devices that might be used to implement those functions. Thus a particular guideline might deal with "pointing" as a function, with no necessary recommendation whether pointing should be accomplished via touch display or light pen or any other physical device. Software is not the only significant factor influencing user performance. Other aspects of user interface design are important, including workstation design, physical display characteristics, keyboard layout, environmental factors such as illumination and noise, and the design of paper forms and written documentation, user training courses, etc. To achieve a good user interface design, all of those factors must be designed with care.

3.1.2 USER INTERFACE SOFTWARE

What sets data processing systems apart as a special breed is the function of each switch button, the functional arrangement among the buttons. The size and distribution of elements within a display are established not in the design of the equipment but in how the computer is programmed. The 'design' in the programs equally establishes the contents of processed data available to the operator and the visual relationships among the data. In combination with or in place of hardware, it can also establish the sequence of actions which the operator must use and the feedback to the operator concerning those actions.

User interface design cannot be the concern only of the psychologist or the human factors specialist. It is a significant part of information system design that must engage the attention of system developers, designers, and ultimately system users as well.

In designing computer-based information systems, special attention must be given to software supporting the user interface.

A comprehensive set of guidelines for design of user interface software in computer-based information systems exist in another unit of this study pack. Also, the general problems of user interface design and the particular need for guidelines to design user interface software are identified,.

3.1. 3 INFORMATION SYSTEMS AND INTERFACE USERS

Computers today are used for a broad range of applications. User interface design guidelines cannot be applied usefully in every case. Some computers may be embedded as components in larger systems, so that they communicate only with other computers and not directly with human users. When there is no user interface, then no user interface design guidelines are needed.

The particular tasks for which a general-purpose computer might be used are not defined in advance by the designer. Instead, a user must provide exact instructions to program the computer to perform any task at hand. The designer may try to ensure that the computer can process appropriate programming languages, but otherwise is not concerned with explicit design of a user interface.

To the extent that information systems support human users performing defined tasks, careful design of the user-system interface will be needed to ensure effective system operation.

Users of information systems interact with a computer in order to accomplish information handling tasks necessary to get their jobs done. They differ in ability, training and job experience. They may be keenly concerned with task performance, but may have little knowledge of (or interest in) the computers themselves. Design of the user-system interface must take account of those human factors.

3.2 SIGNIFICANCE OF THE USER INTERFACE

The design of user interface software is not only expensive and time-consuming, but it is also critical for effective system performance.

In a constrained environment, such as that of many military and commercial information systems, users may have little choice but to make do with whatever interface design is provided. There the symptoms of poor user interface design may appear in degraded performance. Frequent and/or serious errors in data handling may result from confusing user interface design. Tedious user procedures may slow data processing, resulting in longer queues at the checkout counter, the bank cashier window, the visa office, a company's security check out point, or any other workplace where the potential benefits of computer support are outweighed by an unintended increase in human effort.

In situations where degradation in system performance is not so easily measured, symptoms of poor user interface design may appear as user complaints. The system may be described as hard to learn, or clumsy, tiring and slow to use. The users' view of a system is conditioned chiefly by experience with its interface. If the user interface is unsatisfactory, the users' view of the system will be negative regardless of any niceties of internal computer processing.

A data entry application in which relatively simple improvements to user interface software -- including selection and formatting of displayed data, consistency in wording and procedures, on-line user guidance, explicit error messages, re-entry rather than overtyping for data change, elimination of abbreviations, etc. - - resulted in significantly improved system performance. Data entry was accomplished 25 percent faster, and with 25 percent fewer errors. How can that kind of design improvement be achieved in general practice?

3. 3 INTERFACE DESIGN PRACTICE

User interface software design can be regarded as art rather than science.

As an art, user interface design is best practiced by experts, by specialists experienced in the human engineering of computer systems. Most established information systems, call for a system development sequence starting with requirements analysis, functional specification and verification before any software design begins. The actual course of user interface software development will sometimes depart from that desired sequence. There may be no explicit attempt to determine user interface requirements. Specifications may include only rudimentary references to user interface design, with general statements that the system must be "easy to use". In the absence of effective guidance, both the design and implementation of user interface software may become the responsibility of programmers unfamiliar with operational requirements. Detection and correction of design flaws may occur only after system prototyping, when software changes are difficult to make.

3.4 THE INTERACTION DESIGN PHASES

In this phase, the user interface is specified, sketched, developed, and tested. The goal is to support the identified issues during context, task and user analyses and to meet the HCI evaluation metrics requirements. Design is also based on accepted conventions and experience.

The main activities are interface specification and formative evaluations. Interface specification includes semantic understanding of the information needs to support systems requirements and HCI analysis results.

The syntactical and lexical decisions include metaphors, media, dialogue, and presentation designs. Details of these are given below.

3.4.1 METAPHOR AND VISUALIZATION DESIGN

Metaphor and visualization design helps the user develop a mental model of the system. It is concerned with finding or inventing metaphors or analogies that are appropriate for users to understand the entire system or part of it. Well accepted metaphors include a shopping cart for holding items before checking out in E-Commerce context, and light bulbs for online helps or daily tips in productivity software packages.

3.4.2 MEDIA DESIGN

Media design is concerned with selecting appropriate media types for meeting the specific information presentation needs and human experience needs. Popular media types include text, static images (e.g., painting, drawing or photos), dynamic images (e.g., video clips and animations), and sound. The bandwidth needed for transmitting information depends on the media type. In addition, some media types contain affective qualities that can make pre sentations more interesting and stimulating, or annoying and distasteful.

3.4. 3 DIALOGUE DESIGN

Dialogue design focuses on how information is provided to and captured from users during a specific task. Dialogues are analogous to a converation between two people. Many existing interaction styles can be used such as menus, forms, natural l anguages, dialog boxes, and direct manipula tion.

3.4.4 PRESENTATION DESIGN

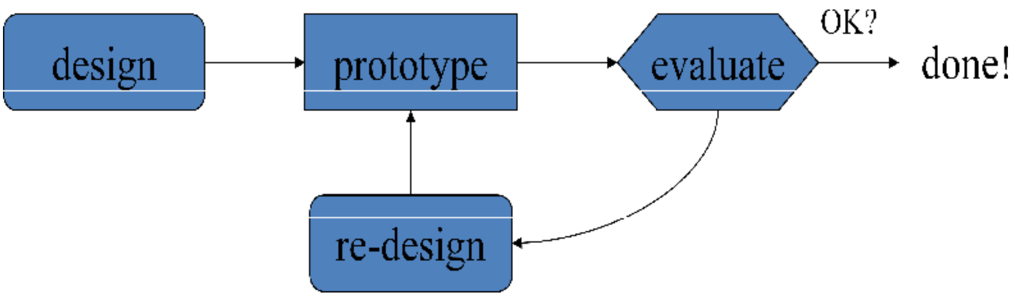
Presentation design concerns the de cisions on information architecture and displ ay layout incorporating metaphors, media, and dialogue desi gns with the rest of the displays.
Commonly established user interface design principles and guidelines may be ap plied during the design stage.
For example, the following presenta tion design principles were suggested by Sut cliffe 1997:

1. Maximize visibility — this means all necessary information should be immediat ly available.
2. Minimize search time with minimu m keystrokes
3. Provide structure and sequence of display
4. Focus on user attention on key da a — here, important information should be s alient and easilycomprehended

S. Provide only relevant information and
6. No overloading of user's working emory.

3.4 FORMATIVE EVALUATION

Formative evaluations identify defects in designs thus inform design iterations an d refinements. A variety of different formative evalua ions can occur several times during the design stage to form final design decisions. In fact, we pro pose that formative evaluations occur during the entire HCI development life cycle, as depicted b elow.



4.0 CONCLUSION

Current user interfaces and their und erlying systems are just too frustrating for ost users. A major effort for software developers should be to increase the reliability and usability of ever system. The system reboots, application crashes, and incompatible file formats that stop user frequency of s in the middle of their tasks must be reduced. Then the inc mprehensible instructions, ambiguous men us, and troubling dialog boxes need to be revised to enable users to complete their work promptly and c nfidently. Long download times of web pages, dropped session s for networked applications, and the disruption caused by unsolicited email (spam), and destructive viruse. As the number of users has grown, all these problems have become more serious. Novice users want the benefits of email or web services, but they are the poorly equipped and motivated to overcome the problems that high-tech early adopters proudly onquered. Improved ns training can help, but improved desi are an important component of making the next generation of ers interfaces more appealing and the u more satisfied.

5.0 SUMMARY

User-system interface includes all aspects of system design that affect system use particularly on those design features of the user interface that are implemented via software.

Special attention must be given to software supporting the user interface and other physical and human factors influencing user performance.

The users' view of a system is conditioned chiefly by experience with its interface.

User interface design is best practiced by experts, by specialists experienced in the human engineering of computer systems.

The interaction design phase is made up of Metaphor and Visualization Design, Media Design, Dialogue Design, Presentation Design, and the Formative Evaluation of the designs.

6.0 Tutor Marked Assignment

1. What do you understand as the user- system interface?
2. 'User interface design cannot be the concern only of the psychologist or the human factors specialist'. What is the significance of this statement?
3. What do you understand as the design formative evaluation?
4. Differentiate between information systems and user interface
5. What is the significance of the user interface to the user and other stakeholders in an organization?
6. What are the implications of the absence of an effective guidance to user interface design?
7. Mention and explain briefly, the four syntactical/lexical phases of interaction design.

7.0 Further Readings / References

- Cooper, Alan. *About Face: The Essentials of User Interface Design.* Foster City, CA: IDG Books Worldwide, 1995.
- Kasik, D.J. "A User Interface Management System," in *Proceedings S IGGRAPH'82: Computer Graphics.* 1982. Boston, MA. 16. pp. 99-106.
- D. Hix, R. Hartson: Developing user interfaces. Wiley, 199 3.
- L. Barfield: The user interface - concepts & design. Addison Wesley, 199 3.
- Monk, P. Wright, J. Haber, L. Davenport: Improving your Human-Computer Interface: a practical technique. Prentice Hall, 199 3.
- Smith, D.C., *et al.* "The Star User Interface: an Overview," in *Proceedings of the 1982 National Computer Conference.* 1982. AFIPS. pp. 515-528.
- Jef Raskin: *The humane interface. New directions for designing interactive systems*. Addison-Wesley, Boston 2000 ISBN 0-201- 379 37-6
- Ben Shneiderman and Catherine Plaisant: *Designing the User Interface: Strategies for Effective Human—Computer Interaction*. 4th ed. Addison Wesley, 2004 ISBN 0- 321-19786-0
- Card, S.K., "Pioneers and Settlers: Methods Used in Successful User Interface Design," in *Human-Computer Interface Design: Success Stories, Emerging Methods, and Real-World Context*, M. Rudisill, *et al.*, Editors. 1996, Morgan Kaufmann Publishers: San Francisco. pp. 122-169.
- Myers, B.A., "A Taxonomy of User Interfaces for Window Managers." *IEEE Computer Graphics and Applications*, 1988. 8(5): pp. 65-84.

UNIT 2: USER INTERFACE DESIGN PRINCIPLES AND CRITERIA/RATIONALE

Table of contents

1.0 INTRODUCTION

The human computer interface can be described as the point of communication between the human user and the computer. The flow of information between the human and the computer is defined as the loop of interaction.

Design criteria and principles are important to designing a new user interface and to evaluate a current user interface.

There are seven principles that may be considered at any time during the design of a user interface and these are: Tolerance, Simplicity, Visibility, Affordance, Consistency, Structure and Feedback. These are briefly discussed in this unit.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Carry out and evaluate simple design using some primary design principles
- Experiment design using some experimental design principles
- Explain the 1 3 principles of display design
- Learn the Norman's 7 design principles
- Know the concepts and types of design rationale

.0 MAIN CONTENT

.1 PRIMARY DESIGN PRINCIPLES

The following seven principles mentioned in the introduction above can be used to guide or evaluate design at any time in the process. The principles can be re-framed or re-structured to suit a particular company or project, or by professional designers.

Principle	Description	Example
Visibility	Clarity	Is the goal obvious? Are icons used?
Feedback	Information sent back to user after their action	Is the feedback in sound? Is there a label showing success or failure?
Affordance	How *clear* is the use of an element to the user?	Label "Push" on one side of a door; a button saying "Click Me"
Simplicity	Utilise the principles of usability	Place an Open File option on a menu, under File tag

Structure	Are the elements set out in a meaningful way from the perspective of the user?	Grouping information within a dialogue box.
Consistency	How easy is it to learn and remember the appearance, positioning and behaviour of the elements?	X to close a window is always on top right hand side of the window; the most important buttons are the same size with only labels indicating different goals.
Tolerance	Prevents user making errors or provides easy recovery or graceful failure	Ignoring of wrong or invalid keyboard input; Hiding options inappropriate in a context.

The loop of interaction has several aspects to it including the:
Task Environment: These are the conditions and goals set upon the user.
Machine Environment: This is the environment that the computer is connected, e.g. a laptop in a college student's dormitory room.
Areas of the Interface: Non-overlapping areas involve processes of the human and computer not pertaining to their interaction. While the overlapping areas only concern themselves with the processes pertaining to their interaction.
Input Flow: This begins in the task environment as the user has some task that requires using their computer.
Output: This is the flow of information that originates in the machine environment.
Feedback: These are loops through the interface that evaluate, moderate, and confirm processes as they pass from the human through the interface to the computer and back.

3.2 EXPERIMENTAL DESIGN PRINCIPLES

Experimental design principles are also important to evaluate a current user interface or to design a new user interface and they are described below:

Early focus on user(s) and task(s):
Establish how many users are needed to perform the task(s) and determine who the appropriate users should be; someone that has never used the interface, and will not use the interface in the future, is most likely not a valid user.
In addition, define the task(s) the users will be performing and how often the task(s) need to be performed.

Empirical measurement:
Test the interface early on with real users who come in contact with the interface on an everyday basis, respectively. Keep in mind that results may be altered if the performance level of the user is not an accurate depiction of the real human-computer interaction.
Establish quantitative usability specifics such as: the number of users performing the task(s), the time to complete the task(s), and the number of errors made during the task(s).

Iterative design:
After determining the users, tasks, and empirical measurements to include, perform the following iterative design steps:
Design the user interface

Test the interface design
Analyze results of using the interface
Repeat the iterative design process until a sensible, user-friendly interface is created.

Design Methodologies

A number of diverse methodologies outlining techniques for human—computer interaction design have emerged since the rise of the field in the 1980s.Most design methodologies stem from a model for how users, designers, and technical systems interact.
Early methodologies, for example, treated users' cognitive processes as predictable and quantifiable and encouraged design practitioners to look to cognitive science results in areas such as memory and attention when designing user interfaces.
Modern models tend to focus on a constant feedback and conversation between users, designers, and engineers and push for technical systems to be wrapped around the types of experiences users want to have, rather than wrapping user experience around a completed system.

User-centered design:

User-centered design (UCD) is a modern, widely practiced design philosophy rooted in the idea that users must take center-stage in the design of any computer system. Users, designers and technical practitioners work together to articulate the wants, needs and limitations of the user and create a system that addresses these elements. Often, user-centered design projects are informed by ethnographic studies of the environments in which users will be interacting with the system.

Display Design

Displays are human-made artifacts designed to support the perception of relevant system variables and to facilitate further processing of that information. Before a display is designed, the task that the display is intended to support must be defined (e.g. navigating, controlling, decision making, learning, entertaining, etc.). A user or operator must be able to process whatever information that a system generates and displays; therefore, the information must be displayed according to principles in a manner that will support perception, situation awareness, and understanding.

3. 3 THIRTEEN PRINCIPLES OF DISPLAY DESIGN

These are principles of human perception and information processing that can be utilized to create an effective display design.
A reduction in errors, a reduction in required training time, an increase in efficiency, and an increase in user satisfaction are a few of the many potential benefits that can be achieved through utilization of these principles.
Certain principles may not be applicable to different displays or situations. Some principles may seem to be conflicting, and there is no simple solution to say that one principle is more important than another. The principles may be tailored to a specific design or situation. Striking a functional balance among the principles is critical for an effective design.

The thirteen principles are:
Perceptual Principles

1. Make displays legible (or audible)

A display's legibility is critical and necessary for designing a usable display. If the characters or objects being displayed cannot be discernible, then the operator cannot effectively make use of them.

2. Avoid absolute judgment limits

Do not ask the user to determine the level of a variable on the basis of a single sensory variable (e.g. color, size, loudness). These sensory variables can contain many possible levels.

3. Top-down processing

Signals are likely perceived and interpreted in accordance with what is expected based on a user's past experience. If a signal is presented contrary to the user's expectation, more physical evidence of that signal may need to be presented to assure that it is understood correctly.

4. Redundancy gain

If a signal is presented more than once, it is more likely that it will be understood correctly. This can be done by presenting the signal in alternative physical forms (e.g. color and shape, voice and print, etc.), as redundancy does not imply repetition. A traffic light is a good example of redundancy, as color and position are redundant.

5. Similarity causes confusion: Use discriminable elements

Signals that appear to be similar will likely be confused. The ratio of similar features to different features causes signals to be similar. For example, A42 3B9 is more similar to A42 3B8 than 92 is to 9 3. Unnecessary similar features should be removed and dissimilar features should be highlighted.

Mental Model Principles

6. Principle of pictorial realism

A display should look like the variable that it represents (e.g. high temperature on a thermometer shown as a higher vertical level). If there are multiple elements, they can be configured in a manner that looks like it would in the represented environment.

7. Principle of the moving part

Moving elements should move in a pattern and direction compatible with the user's mental model of how it actually moves in the system. For example, the moving element on an altimeter should move upward with increasing altitude.

Principles Based on Attention

8. Minimizing information access cost

When the user's attention is averted from one location to another to access necessary information, there is an associated cost in time or effort. A display design should minimize this cost by allowing for frequently accessed sources to be located at the nearest possible position. However, adequate legibility should not be sacrificed to reduce this cost.

9. Proximity compatibility principle

Divided attention between two information sources may be necessary for the completion of one task. These sources must be mentally integrated and are defined to have close mental proximity. Information access costs should be low, which can be achieved in many ways (e.g. close proximity, linkage by common colors, patterns, shapes, etc.). However, close display proximity can be harmful by causing too much clutter.

10. Principle of multiple resources

A user can more easily process information across different resources. For example, visual and auditory information can be presented simultaneously rather than presenting all visual or all auditory information.

Memory Principles

11. Replace memory with visual information: knowledge in the world

A user should not need to retain important information solely in working memory or to retrieve it from long-term memory. A menu, checklist, or another display can aid the user by easing the use of their memory. However, the use of memory may sometimes benefit the user rather than the need for reference to some type of knowledge in the world (e.g. a expert computer operator would rather use direct commands from their memory rather than referring to a manual). The use of knowledge in a user's head and knowledge in the world must be balanced for an effective design.

12. Principle of predictive aiding

Proactive actions are usually more effective than reactive actions. A display should attempt to eliminate resource-demanding cognitive tasks and replace them with simpler perceptual tasks to reduce the use of

the user's mental resources. This will allow the user to not only focus on current conditions, but also think about possible future conditions.
An example of a predictive aid is a road sign displaying the distance from a certain destination.
13. Principle of consistency
Old habits from other displays will easily transfer to support processing of new displays if they are designed in a consistent manner. A user's long-term memory will trigger actions that are expected to be appropriate. A design must accept this fact and utilize consistency among different displays.

3.4 The Norman's 7 Design Principles

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

3.4.1 DESIGN PRINCIPLES FORMULATED TO SUPPORT USABILITY :

Principle of Learnability : This is the ease with which new users can begin effective interaction and achieve maximal performance
Principle of Flexibility: These are the multiplicity of ways the user and system exchange information
Principle of Robustness: This is the level of support provided the user in determining successful achievement and assessment of goal-directed behaviour

The Principles of learnability are broken down into :
Predictability : This is determining effect of future actions based on past interaction history and its operation visibility
Synthesizability: This is assessing the effect of past actions, its immediate and its eventual honesty
Familiarity: This is how prior knowledge applies to new system and how easy one can guess its affordance Generalizability: This is extending specific interaction knowledge to new situations
Consistency: This concerns the likeness in input and output behaviour arising from similar situations or task objectives

Principles of flexibility comprise:
Dialogue initiative : This is the freedom from system imposed constraints on input dialogue and it compares the system against the user pre-emptiveness.
Multithreading: This is expressing the ability of the system to support user interaction for more than one task at a time. It also looks at the concurrent and interleaving multimodality.
Task migratability: This is passing responsibility for task execution between user and system
Substitutivity: This allows equivalent values of input and output to be substituted for each other. It compares representation multiplicity and equal opportunity
Customizability: This is the modifiability and adaptability of the user interface by user or the modifiability and adaptivity of the user interface by the system.

Principles of robustness are made up of:
Observability: This is the ability of the user to evaluate the internal state of the system from its perceivable representation. It considers the browsability, the defaults, the reachability, the persistence , and the operation visibility.
Recoverability: This concerns the ability of the user to take corrective action once an error has been recognized. It looks at the reachability, the forward and backward recovery and the commensurate effort.

Responsiveness: This is how the user perceives the rate of communication with th e system and how stable is the response.

Task conformance: This explains the degree to which system services support all of the user's tasks, the task completeness and its adequacy.

3.5 THE DESIGN RATIONALE

Design rationale is an information that explains why a computer system is the way it is.

Benefits of design rationale are :

Communication exists throughout the life cycle

Reuse of design knowledge is made across products

Design rationale enforces design discipline

It presents arguments for design trade-offs

It organizes potentially large design space

It is used to capture contextual information

3.5.1 Types of Design Rationale:

Process-oriented: this preserves ord r of deliberation and decision-making

Structure-oriented: this emphasizes post hoc structuring of considered design alternatives

Two examples design rationale are:

Issue-based information system (IBIS) and

Design space analysis

Issue-based information system (IBIS)

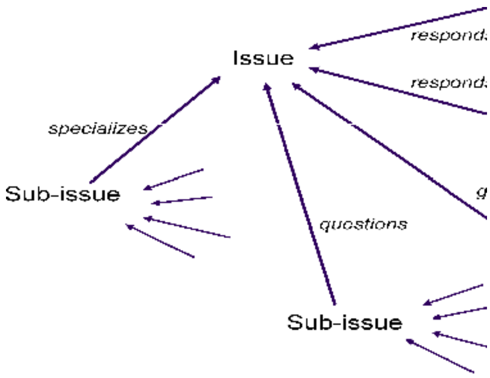
The Issue-based information system (IBIS) provides basis for much of design rationale research and it is process-oriented. The main elements are :

Issues: These describe the hierarchical structure with one 'root' issue

Positions: These contain the potential resolutions of an issue

Arguments: These modify the relationship between positions and issues

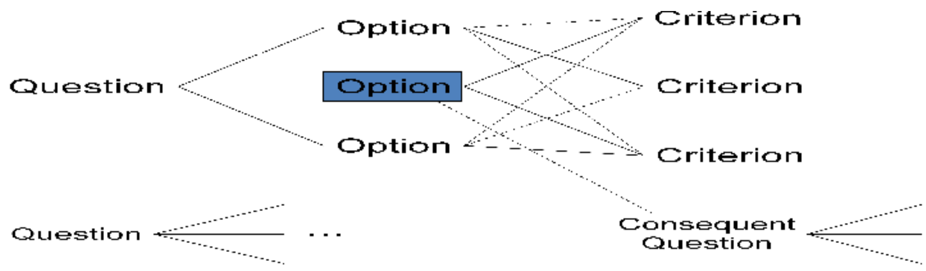
The graphical version or structure of IBIS (gIBIS) is produced below



Design space analysis

This is structure-oriented QOC (Questions, Options and Criteria) is a hierarchical structure made up of questions (and sub-questions) and represent major issues of a design. The options provide alternative solutions to the question while the criteria is the means to assess the options in order to make a choice

The QOC notation (Question, Option and Criterion)



35.2 Characteristics of psychological design rationale

- To support task-artefact cycle in which user tasks are affected by the systems they use
- It aims to make explicit consequences of design for users
- Designers identify tasks the system will support
- Various scenarios are suggested to test task
- Users are observed while using the system
- The psychological claims of the system are made explicit
- The negative aspects of the design can be used to improve next iteration of design

4.0 CONCLUSION

The various design principles mentioned can be used to guide or evaluate design at any time in the process. The principles can be re-framed or re-structured to suit a particular company or project, or by professional designers.

5.0 SUMMARY

Experimental design principles are important to evaluate a current user interface or to design a new user interface. They comprise the empirical measurement and iterative design.

Most design methodologies stem from a model on how users, designers, and technical systems interact. User-centered design (UCD) is a modern, widely practiced design philosophy rooted in the idea that users must take center-stage in the design of any computer system

Displays are human-made artifacts designed to support the perception of relevant system variables and to facilitate further processing of that information.

Principles of display design comprise the Perceptual Principles, the Mental Model Based on Attention, and the Memory Principles

Design rationale is an information that explains why a computer system is the way it is, it could be Process-oriented and/or Structure-oriented.

6.0 Tutor Marked Assignment

1. Briefly describe the seven principles to be considered while designing a user interface
2. Explain the 5 aspects that govern the loop of human Computer interaction.
3. Why is it necessary for an interactive designer to consider experimental design principles during his design? Mention some of the principles
4. The thirteen principles of display design are Principles of human perception and information processing that can be utilised by the designer to create an effective display.
   (a) What are the potential benefits achievable through utilization of these principles?
   (b) Mention any 2 principles under each of the following categories (i) Perceptual (ii) Mental model (iii) Those based on attention and (iv) memory
5. Mention the five benefits of a design rationale
6. (a) What is the objective of a design rationale?
   (b)Produce the graphical versions of an example of a design rationale. You can select either the given examples of issue based information System (IBIS) or the Design space analysis

7. What are the characteristics of the psychological design rationale?

7.0 Further Readings / References

- C. Brown: Human-Computer Interface design guidelines. Ablex, 1989.
- Cooper, Alan. *About Face: The Essentials of User Interface Design.* Foster City, CA: IDG Books Worldwide, 1995.
- Kasik, D.J. "A User Interface Management System," in *Proceedings S IGGRAPH'82: Computer Graphics.* 1982. Boston, MA. 16. pp. 99-106.
- D. Hix, R. Hartson: Developing user interfaces. Wiley, 199 3.
- L. Barfield: The user interface - concepts & design. Addison Wesley, 1993.
- Monk, P. Wright, J. Haber, L. Davenport: Improving your Human-Computer Interface: a practical technique. Prentice Hall, 199 3.
- W. Galitz: Handbook of screen format design. QED, 1989.
- C. Gram, G. Cockton (eds.): Design principles for interactive software. Capman & Hall, 1996.
- ISO 9241 (Part 10: Dialogue principles, Part 12: Presentation of information, Part 14: Menu dialogues, Part 15: Command dialogues, Part 16: Direct manipulation dialogues, Part 17: Form fill-in dialogues)
- D. Mayhew: Principles and guidelines in software user interface design. Prentice, 1992.

UNIT 3: USER INTERFACE DESIGN PROGRAMMING TOOLS

Table of contents

1.0 INTRODUCTION

The programming tools for the design of user interface give implementation support for the levels of services for programmers. These include the windowing systems that provide the core support for separate and simultaneous user-system activity. They enable easy programming of the application and the control of dialogue between the system and the user. The interaction toolkits for example, bring programming closer to the level of user perception while the user interface management systems control the relationship between the presentation and functionality.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Explain the various levels of programming support tools
- Utilize toolkits for programming interaction objects
- Understand the concepts of the User Interface Management Systems(UIMS)

.0 MAIN CONTENT

3.1 How Human Computer Interaction affects the programmer

Advances in coding have elevated programming through hardware that specifically improves upon the programmer's Interaction-technique.

The layers of development tools, as earlier mentioned, also contribute to how human computer interaction affects the programmer. These tools incorporate the windowing systems, the interaction toolkits and the user interface management systems as exemplified in the following:

Levels of programming support tools

- Windowing systems
  - device independence
  - multiple tasks
- Paradigms for programming the application
  - read-evaluation loop
  - notification-based
- Toolkits
  - programming interaction objects
- UIMS
  - conceptual architectures for separation
  - techniques for expressing dialogue

3.1.1 Elements of the Windowing Systems

Device independence

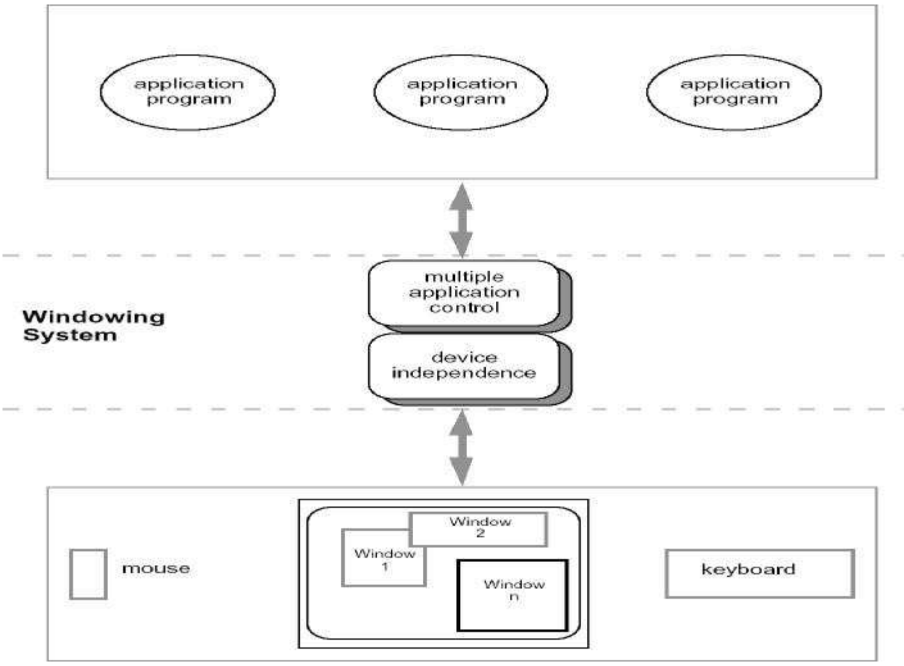
Programming the abstract terminal device drivers using the image models for output and input is device independent. Also device independence is the creation of the image models for output and the input, partially. These image models are the pixels, the PostScript
(as in Macintosh Operating System X and NextStep), the Graphical Kernel System (GKS) and the Programmers' Hierarchical Interface to Graphics (PHIGS)

Resource sharing

Another element of the windowing system is resource sharing. This is the act of achieving simultaneity of user tasks. Resource sharing enables the use of the window system to support independent processes by the isolation of individual applications.

Elements of windowing systems

3.1.2 Role of a windowing system



As shown in the diagram above, the windowing system comprising the multiple application control and the device independent control enables the interface between the application programs and the user.

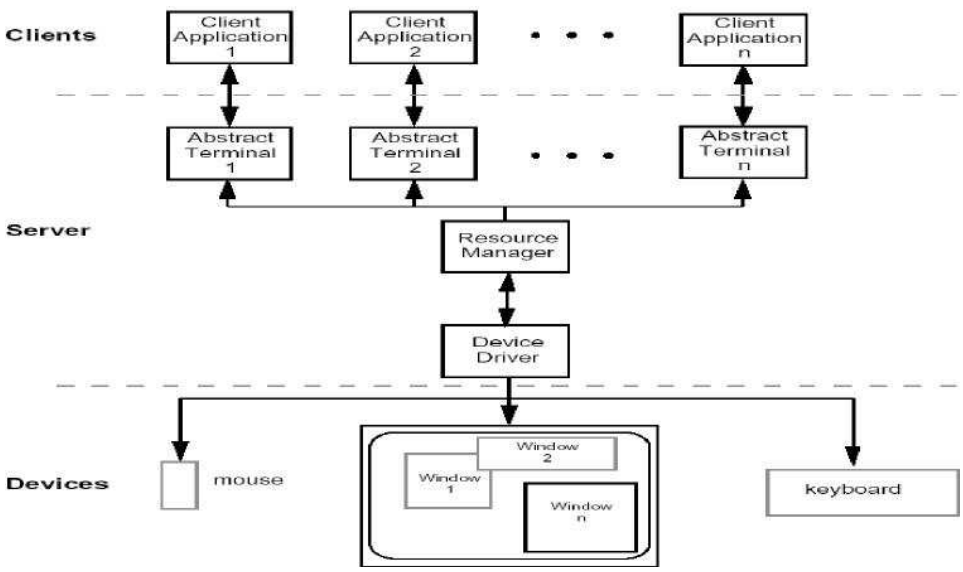
3.1. 3 The Architectures of windowing systems

The Architectures of windowing systems are analysed through three possible software architectures if we all assume device driver is separate and know how they differ and how the multiple application management is implemented.

The three possible software architectures are in the following forms:

1. When each application manages all processes. Here, everyone worries about synchronization and reduces portability of applications
2. When management role within kernel of operating system ensures that applications are tied to operating system, and
3. When management role as separate application ensures maximum portability

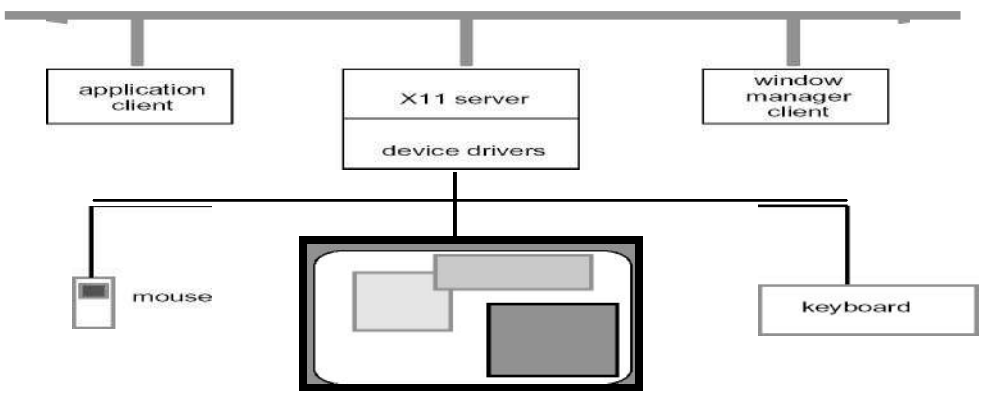
The client-server architecture is illustrated below:



3.1.4 X Windows architecture

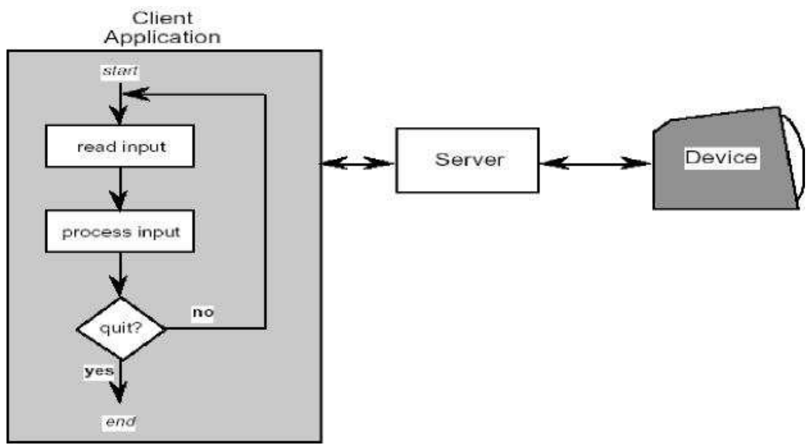
The X Windows architecture comprises the Pixel imaging model with some pointing mechanism and the X protocol that defines the server-client communication.

The architecture also contains a separate window manager client that enforces policies for input and output. Policies on how to change input focus, how the tiled windows compare with overlapping windows and policies on inter-client data transfer. See the pictorial illustration below.



3.1.5 Typical program models of the application:

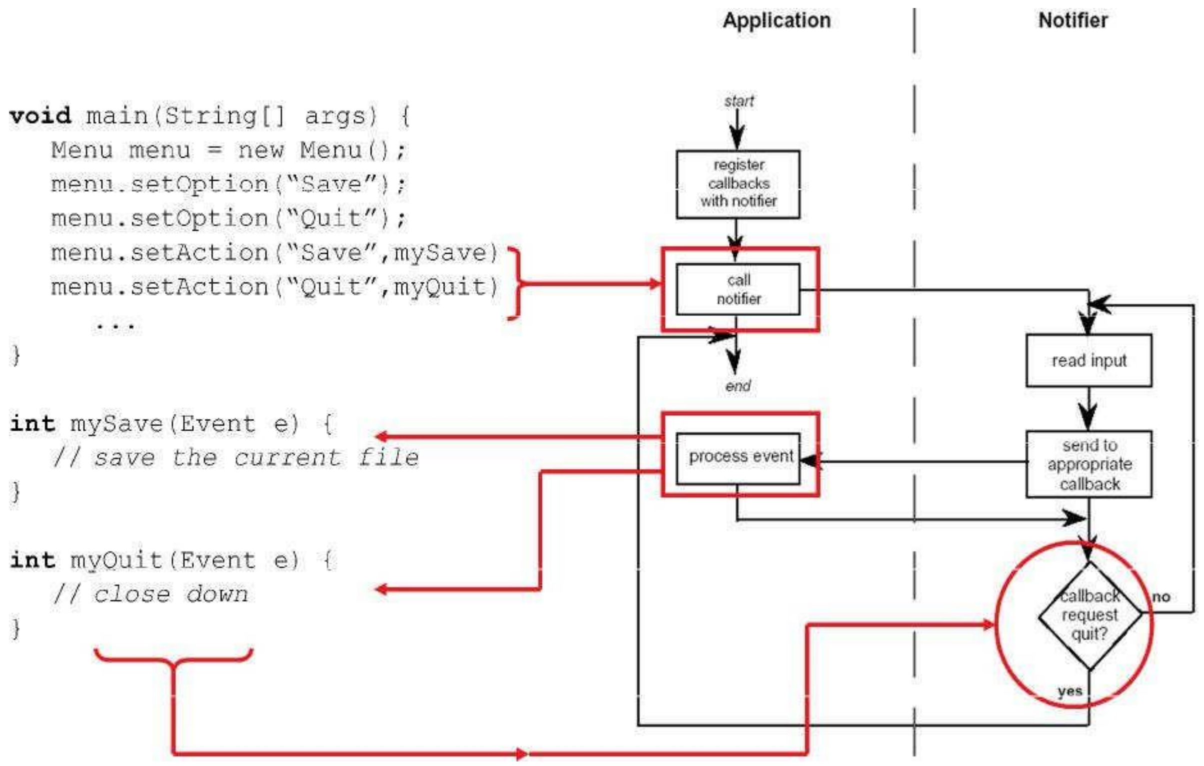
Programming the application - 1



A typical read-evaluation loop is provided below:

```
repeat
        read-event(myevent)
        case myevent.type
                type_1:
                        do typeG1 processing
                type_2:
                        do typeG2 processing
                        type_n:
                        do typeGn processing
        end case
end repeat
```

Programming the application - 2
Notification-based



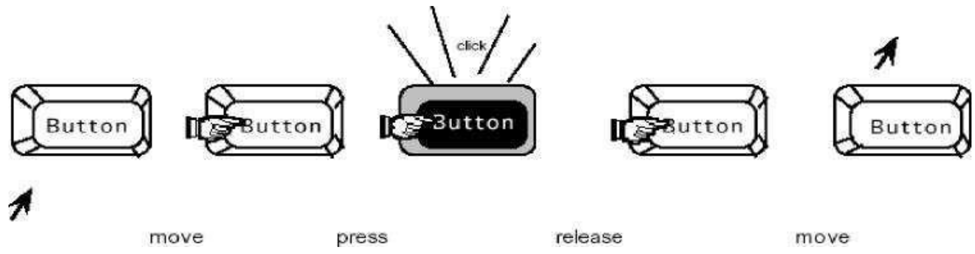
.2 USING TOOLKITS

3.2.1 User Interface Toolkits

User interface toolkits are probably the most widely used tool nowadays to implement applications. All three major platforms (Unix/Linux, MacOS, and Windows) come with at least one standard UI toolkit.

Toolkits are interaction objects with input and output intrinsically linked. They enable programming with interaction techniques using and gadgets. They promote consistency and generalizations through widgets similar look and feel.

A widget is a software object that has three facets that closely match the MVC model: a presentation, a behavior, and an application interface. Sample illustration of widgets is provided below.



The presentation defines the graphical aspect of the widget. The overall presentation of an interface is created by assembling widgets into atree. Widgets, such as buttons are the leaves of the tree. Composite widgets constitute the nodes of the tree and control the layout of their children. The behavior of a widget

defines the interaction methods it supports: a button can be pressed, a scrollbar can be scrolled, and a text field can be edited.
The application interface defines how a widget communicates the results of the user interaction to the rest of the application. It is usually based on a notification mechanism.
One limitation of widgets is that their behaviors are limited to the widget itself. Interaction techniques that involve multiple widgets, such as drag-and-drop, cannot be supported by the widgets' behaviors alone and require separate support in the UI toolkit.

3.2.2 Prototypes and Widgets
In general, prototyping new interaction techniques requires either implementing them within new widget classes, which is not always possible, or not using a toolkit at all. Implementing a new widget class is typically more complicated than implementing the new technique outside the toolkit, (for example, with a graphical library), and is rarely justified for prototyping. Many toolkits provide a blank widget, such as the Canvas in Tk or JFrame in Java Swing, which can be used by the application to implement its own presentation and behavior. This is usually a good alternative to implementing a new widget class, even for production code.

A number of toolkits have also shifted away from the widget model to address other aspects of user interaction. For example, GroupKit was designed for groupware, Jazz for zoomable interfaces, the Visualization and InfoVis toolkits for visualization, Inventor for 3-D graphics, and Metisse for window management.

Creating an application or a prototype with a UI toolkit requires solid knowledge of the toolkit and experience with programming interactive applications. In order to control the complexity of the interrelations between independent pieces of code (creation of widgets, callbacks, global variables, etc.), it is important to use well-known design patterns, otherwise, the code quickly becomes unmanageable and, in the case of a prototype, unsuitable to design models.
Toolkits are amenable to object-oriented programming using Java interfaces that include the AWT (abstract windowing toolkit), a Java toolkit, and some Java classes for buttons and menus, etc.
Some Java interfaces are (i) Notification based such as AWT 1.0 with the need to subclass basic widgets and AWT 1.1 and beyond with call-back objects (ii) Swing toolkit built on top of AWT with higher level features that also uses the MVC architecture.

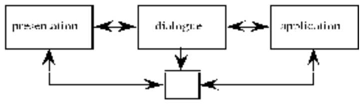
3.2. 3 The User Interface Management Systems (UIMS)
The UIMS add another level above toolkits because toolkits may be too difficult for non-programmers. Concerns of UIMS include the conceptual architecture, the implementation techniques and the support infrastructure

UIMS as conceptual architecture
The conceptual architecture is viewed as the s*eparation between application semantics and presentation. This improves:*

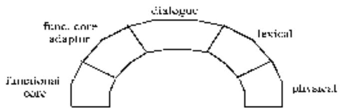
i. Portability which runs on different systems
ii. Reusability having components reused thereby cutting costs
iii. Multiple interfaces that access same functionality
iv. Customizability; here, the system is customised to suit the designer and user.

The User Interface Management System (UIMS) tradition of interface layers and logical components comprise

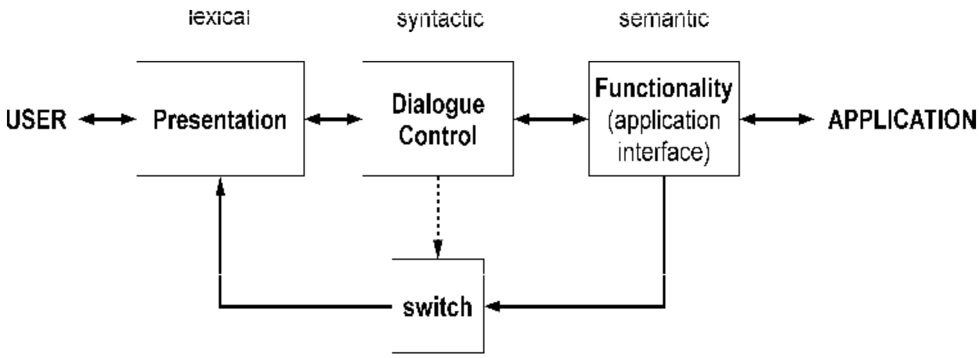
- linguistic: lexical/syntactic/semantic
- Seeheim:



- Arch/Slinky



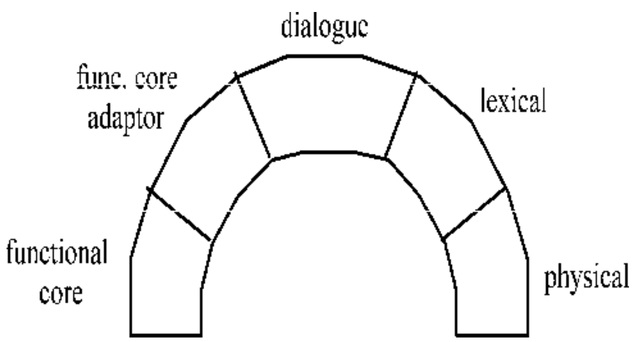
The Seeheim model: Its concept and implementation
Seeheim concept arose as a result of implementation experience with conceptua l approach as principal contribution. The concepts are part f the normal user interface language.



The above depicts different kinds ofeedback. For example, the movement of th e mouse carried out at the presentation interface is known as th lexical feedback, the menu highlights as adialogue control is known as the syntactic feedback while a fun ction carried out at the application interface such as sum of number changing ,is regarded as the semanti c feedback. Because the semantic feedback is oft en slower, programmers prefer to use the r pid lexical and/or the syntactic feedbacks.
The lower box representing the bypa ss switch is needed for implementation. The switch enables a direct communication between application and presentation. Though regulated by a di logue control, it also provides a rapid semantic feedback.

The Arch/Slinky model characteristic s

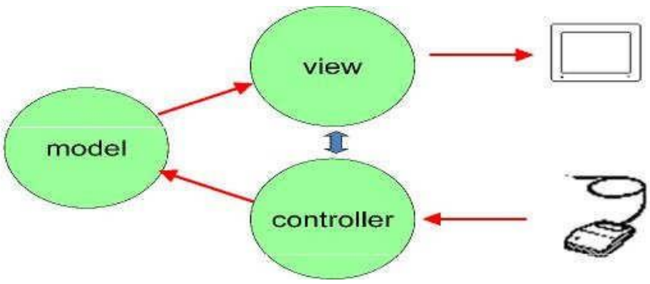
- This model contains more lay ers to distinguish the lexical and the physica l
- Like a 'slinky' spring, differen layers may be thicker (that is more import nt) in different systems or in different components



Monolithic vs. Components
Seeheim has big components and is ften easier to use smaller ones especially. if using object-oriented toolkits .

Smalltalk used the model—view—cont roller (MVC)
Model indicates the internal logical state of component
View shows how it is rendered on sc een
Controller processes user input

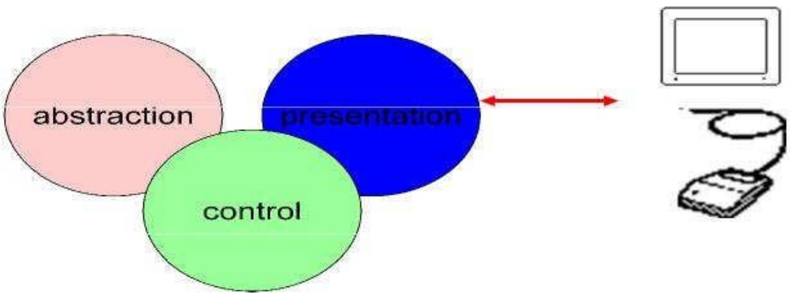
The Model - View - Controller (MVC)



Input ▪ control ▪ model ▪ view▪ outputMVC

issues
MVC is largely pipeline model:
Using the pictorial illustration above, the Input is transmitted to the controller, th e controller processes user input and connects with the mo del. Since the model represents the internal the logical state of the component, a manipulation is carrie out and through the view model, the result of the manipulation is output on screen. In graphical interfa ce, input only has meaning in relation to ou put e.g. a mouse click There is the need to know *what* was clicked and the controller has to decide wha t to do with the click. Using the internal logical state of the component through the model, the view di plays how it is rendered on the screen.
However, in practice, the controller irectly 'talks' to view.

The Presentation, Abstraction and C ntrol (PAC) model
presentation - abstraction - control



The PAC model is closer to the Seehe im model in principle because the term 'Pre sentation' describes input and output are managed, the ' Abstraction' describes the logical state of th component while 'Control' represents the state of me diation between the 'Presentation' and 'Abs raction' .
The PAC model manages the hierarc y and multiple views through the control pa rt of PAC communicate the 'Abstraction'
Though the PAC model is direct, the MVC model is used more in practice as can b e found in the use of Java Swing.

The Implementation of UIMS takes t he following forms:
Implementing the techniques for dia logue controller through the use of the
· Menu networks

- State transition diagrams
- Grammar notations
- Event languages
- Declarative languages
- Constraints
- Graphical specification

4.0 CONCLUSION

Programming tools for the design of user interface enable easy programming of the application and the control of dialogue between the system and the user.

5.0 SUMMARY

Levels of programming support tools comprise Windowing systems that are device independence with multiple tasks, the Paradigms for programming the application with read-evaluation loop that is notification-based , Toolkits containing programming interaction objects, and the User Interface Management System made up of conceptual architectures for separation together with the techniques for expressing the dialogue .

The layers of programming development tools contribute to how human computer interaction affects the programmer.

6.0 Tutor Marked Assignment

1 (a) Explain the two elements of the windowing systems

(b)Using the pictorial representation of a windowing system, describe its role in interactive programming

1. Describe the architecture of a Windowing System using the client server architecture diagram.
2. What are the functions of the User interface toolkits?
3. In what ways are widgets used to support prototyping?
4. What is the alternative to the use of a took-kit? What could require the use of such alternative?
5. Using a suitable pictorial representation, describe the concept and implementation of the Seeheim model
6. What is the concept supporting the model 'View Controller'(MVC)? How useful is this concept to the programmer?
7. Explain how closely is the relationship between the MVC model and the Presentation, Abstraction and Control (PAC) model?
8. In what form can the implementation of the User Interface Management System be carried out?

7.0 Further Readings / References

- Buxton, W., *et al.* "Towards a Comprehensive User Interface Management System," in *Proceedings SIGGRAPH'83: Computer Graphics.* 198 3. Detroit, Mich. 17. pp. 35-42.
- Kasik, D.J. "A User Interface Management System," in *Proceedings S IGGRAPH'82: Computer Graphics.* 1982. Boston, MA. 16. pp. 99-106.
- Scheifler, R.W. and Gettys, J., "The X Window System." *ACM Transactions on Graphics*, 1986. 5(2): pp. 79-109.
- Myers, B.A., "User Interface Software Tools." *ACM Transactions on Computer Human Interaction*, 1995. 2(1): pp. 64-10 3.
- Palay, A.J., *et al.* "The Andrew Toolkit - An Overview," in *Proceedings Winter Usenix Technical Conference.* 1988. Dallas, Tex. pp. 9-21.

UNIT 4: THE SOFTWARE DESIGN PROCESS OF HUMAN COMPUTER INTERACTION

Table of contents

1.0 INTRODUCTION

Software engineering is the discipline for understanding the software design process, or life cycle. Therefore, this unit looks at the software design process of human computer interaction by analysing the pros (linearity) and cons (non linearity) of the water fall model that comprises the design life cycle. The usability engineering process that measures the user's experiences is weighed against the ISO usability standards 9241. For a successful and effective design, management issues concerned with interactive design and prototyping are considered along the relevant design rationale.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Understand the distinct activities that constitute the software engineering life cycle
- Understand the concepts of the water fall model in relationship to the software life cycle
- Differentiate between software engineering life cycle usability engineering
- Explain the concepts of interactive design and prototyping

.0 MAIN CONTENT

3.1 The software process of Human Computer Interaction

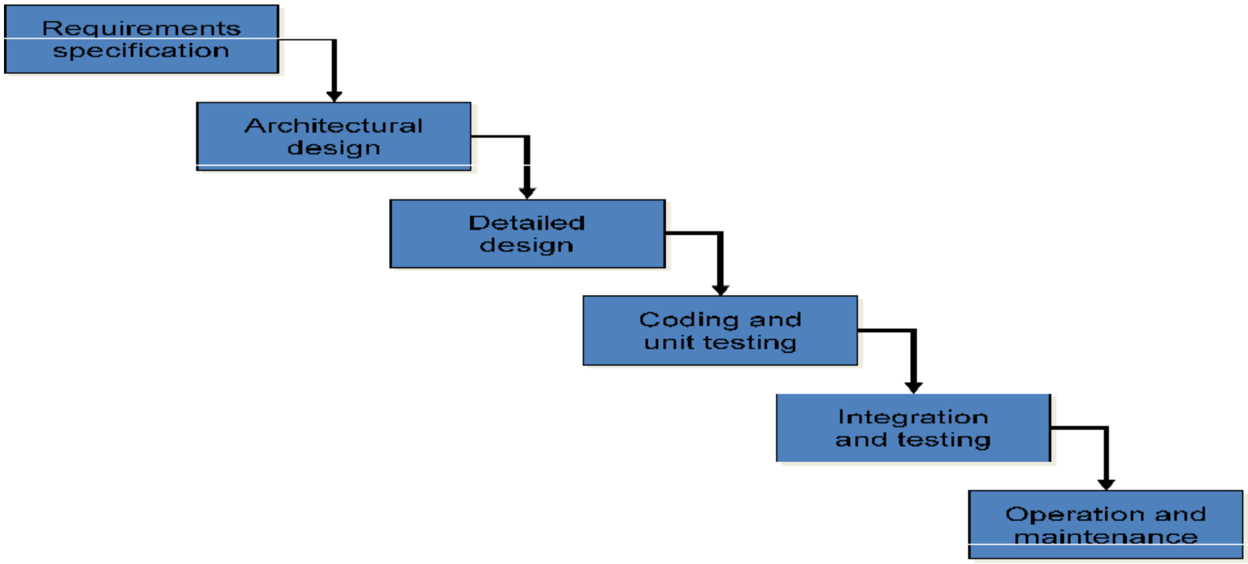
The software process comprises the following:

- Software engineering and the design process for interactive systems
- Usability engineering
- Iterative design and prototyping
- Recording the design knowledge using the design rationale

Software engineering is the discipline for understanding the software design process, or life cycle. The design for usability occurs at all stages of the life cycle, not as a single isolated activity.
Usability engineering is the ultimate test of usability based on measurement of user experience.
Iterative design and prototyping overcomes inherent problems of incomplete requirements
Design rationale is information that explains why a computer system is the way it is.

.2 The waterfall model

The waterfall model depicts the software life cycle.
The pictorial illustration that follows reflects a mountain top from where water f lls towards the bottom of the mountain, hence called a waterf ll. It shows the commencement of the life cycle (the requirements specification) through the design, co ding and testing processes to its ultimate ter mination (the operation and maintenance).



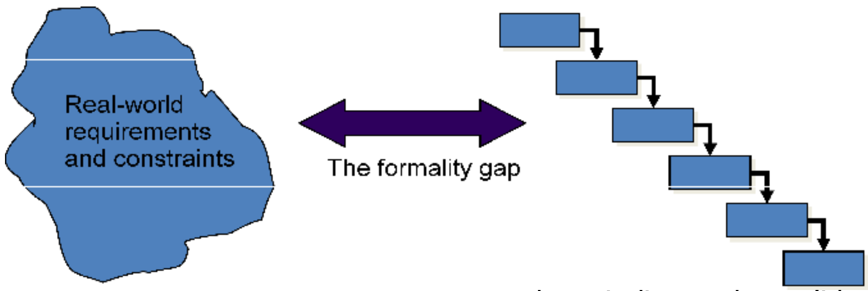
Activities in the software lifecycle ar :

Requirements specification:
Here, the designer and client try to c apture what the system is expected to provi de and can be expressed in natural language or more precise l anguages, such as a task analysis would prov ide.

Architectural design:
This is a high-level description of ho the system will provide the services requir d. It describes how to d. factor system into major component s of the system and how they are interrelate It shows the needs to satisfy both functional and non-funct ional requirements

Detailed design
This concerns a refinement of archit ctural components and their interrelations to identify modules to implemented separately. The refine ent is governed by the non-functional requ be rements.

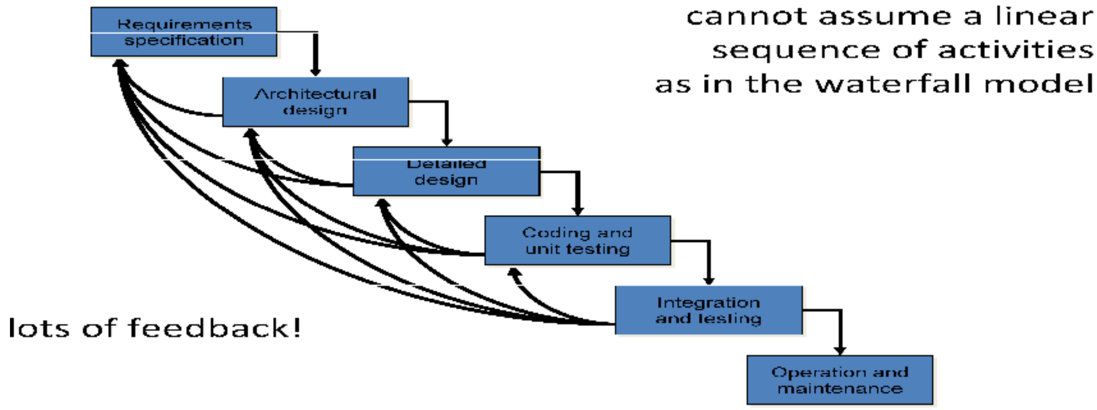
Verification and Validation
Verification: This is ensuring that the product is designed right
Validation: This is ensuring that the right product is designed.



The formality gap shown in the diagr am above indicates that validation will alwa s rely to some extent on subjective means of proof.

Management and contractual issues describes design in commercial and legal co ntexts

. 3 The life cycle for interactive s ystems



The life cycle for interactive systems cannot assume a linear sequence of activitie s as in the waterfall model because there are lots of feed backs occurring within the initial requireme ts specification, the designs, the coding and testing proce sses. See the illustration above.

3.4 Usability engineering:

This is the ultimate test of usability based on measurement of user experience. Usability engineering demands that specific usability measures be made explicit as requirements.

Usability specification comprises the usability attribute and/or principle, the mea suring concept and the measuring method. It also depicts either the present level the worst case level, t e planned level or the best case level.

The problems associated with usability specifications are:

i. Usability specification requires lev el of detail that may not be possible in arly life of the design process.

ii. It does not necessarily satisfy usab the e ility.

Example of a usability specification

Attribute: Backward recoverability

Measuring concept: Undo an errone us programming sequence

Measuring method: Number of expli cit user actions to undo current program

Present level: An undo is not allowe d presently

Worst case: This considers as many a ctions as it takes to program in mistakes

Planned level: A maximum of two ex plicit user actions are allowed

Best case: One explicit cancel action is allowed

3.5 ISO usability standard 9241:

ISO usability standard 9241 adopts the following traditional usability categories:Effectiveness: This is achieving what you want to.

Efficiency: This is doing it without wasting effort.

Satisfaction: This is showing whether or not you enjoy the process.

Some metrics from ISO 9241 are :

Usability objective	Effectiveness measures	Efficiency measures	Satisfaction measures
Suitability for the task	Percentage of goals achieved	Time to complete a task	Rating scale for satisfaction
Appropriate for trained users	Number of power features used	Relative efficiency compared with an expert user	Rating scale for satisfaction with power features
Learnability	Percentage of functions learned	Time to learn criterion	Rating scale for ease of learning
Error tolerance	Percentage of errors corrected successfully	Time spent on correcting errors	Rating scale for error handling

3.6 Iterative design and Prototyping

Iterative design overcomes inherent problems of incomplete requirements while prototypes simulate or animate some features of intended system.

Different types of prototypes can be identified as: (i) throw-away, (ii), incremental and (iii) evolutionary

Management issues concerned with Interactive design and Prototyping:

The management issues are:

- Time allocated to the design,
- Planning the design,
- Non-functional features of the design and
- Contracts of the design.

Prototyping Techniques:

These include the following:

- Storyboards which need not be computer-based but can be animated
- Limited functionality simulations in which some part of the system functionality is provided by designers. Such tools like HyperCard are common.
- Wizard of Oz technique
- Warning about the iterative design. These show concerns about the design inertia in which early bad decisions stay bad. It is better to diagnose the real usability problems in prototypes; and not just the symptoms.

4.0 CONCLUSION

The life cycle for interactive systems cannot assume a linear sequence of activities as in the waterfall model because there are lots of feedbacks!

5.0 SUMMARY

The software engineering life cycle consists of distinct activities and the consequences for interactive system design. The waterfall model depicts the software life cycle.
Usability engineering makes usability measurements explicit as requirements and is the ultimate test of usability based on measurement of user experience.
Iterative design overcomes inherent problems of incomplete requirements.
Prototyping techniques comprise storyboards, limited functionality simulations or animations of intended system, and Wizards.
Design rationale records design knowledge through the process and structure orientations.

6.0 Tutor Marked Assignment

1. Briefly explain the four tasks of the software process of Human Computer Interaction.
2 (a) What do you understand by the term "the Waterfall model" in the software design process of human computer interaction?
(b) With the aid of a related diagram, briefly explain the activities depicted by the "waterfall model"
3. What do you observe as the major departure of the interactive system life cycle from the "waterfall model"? What do you think as the reason behind this major departure?
4. (a) Explain the value of usability engineering in the Software design process
(b) What are the major problems associated with usability specifications? ©
Give any three examples of a usability specification
5. What are the usability objectives of the International Standards Organisation (ISO) 9241? Mention its usability categories
6. (a) Differentiate between the objectives of interactive design and prototyping. (b) What are the management issues concerned with interactive design and prototyping?

7.0 Further Readings / References

- Swinehart, D., *et al.*, "A Structural View of the Cedar Programming Environment." *ACM Transactions on Programming Languages and Systems*, 1986. 8(4): pp. 419-490.
- Swinehart, D.C., *Copilot: A Multiple Process Approach to Interactive Programming Systems.* PhD Thesis, Computer Science Department Stanford University, 1974, SAIL Memo AIM-2 30 and CSD Report STAN-CS-74-412.
- Teitelman, W., "A Display Oriented Programmer's Assistant." *International Journal of Man-Machine Studies*, 1979. 11: pp. 157-187. Also Xerox PARC Technical Report CSL-77- 3, Palo Alto, CA, March 8, 1977.
- Newman, W.M. "A System for Interactive Graphical Programming," in *AFIPS Spring Joint Computer Conference.* 1968. 28. pp. 47-54.
- Shneiderman, B., "Direct Manipulation: A Step Beyond Programming Languages." *IEEE Computer*, 198 3. 16(8): pp. 57-69.
- Smith, D.C., *Pygmalion: A Computer Program to Model and Stimulate Creative Thought.* 1977, Basel, Stuttgart: Birkhauser Verlag. PhD Thesis, Stanford University Computer Science Department, 1975.

UNIT 5: INTERACTIONS IN HYPERTEXT, MULTIMEDIA AND THE WORLD WIDE WEB

Table of contents

HUMAN COMPUTER INTERACTION IN HYPERTEXT, MULTIMEDIA AND THE WORLD-WIDE WEB

1.0 INTRODUCTION

Understanding hypertext

It enables you to find information by navigating the hyperspace using the web technology

Hypermedia is not just text but hypertext systems containing additional media such as illustrations, photographs, video and sound.

The web contains protocols, browsers, web servers, clients and a lot of networking. The challenges remain a loss in hyperspace and information overload.

The advantage of this option is an interactive Data Base access. The availability of bandwidth and the security of data are problem issues to be resolved.

2.0 OBJECTIVES

Understanding hypertext

Multimedia or Hypermedia

Animation

Video and Audio effects.

Web technology issues:

Network issues

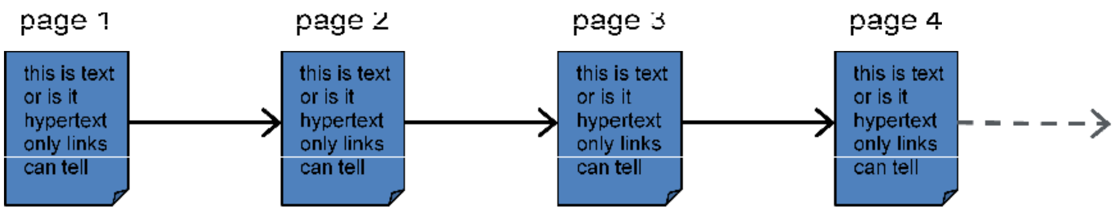
The web content could be made static by unchanging pictures and text or made dynamic with interaction and applications on the web.

.0 MAIN CONTENT

3.1 Understanding hypertext

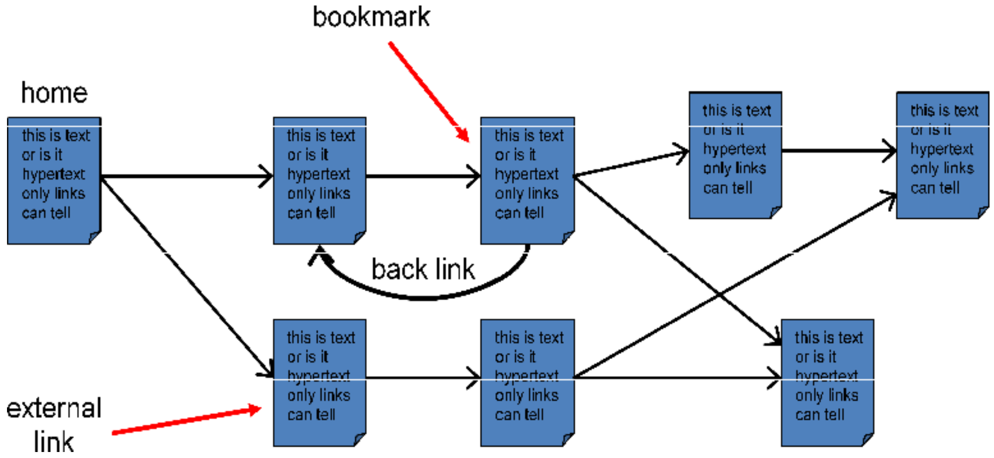
Hypertext enables you to find inform ation by navigating the hyperspace using th web technology Hypertext is made up of rich content of graphics, audio, video, computation and nteraction.

Text



In some cases hypertext imposes stri ct linear progression on the reader but in m st cases it is not linear as shown in the diagram below:

Hypertext - not just linear



From the above, we can see that the non-linear structure contains blocks of text (pages) with links between pages that create a mesh o r network. The users follow their own path f r the information desired.

3.2 Multimedia or Hypermedia

Multimedia is sometimes also called Hypermedia. The term can also be used forimple audio and video effects.

Hypermedia is not just text but hype rtext systems containing additional media su ch as illustrations, photographs, video and sound.

Links and hotspots may be in media with their pictures, the times and locations i video.

Animation

Animation is adding motion to image s particularly images on things that change i n time. Examples are digital faces that take seconds to tick past or warp into the next and analogue fac e with hands swept around the clock face.

Animation comprises live displays fo r showing status and progress, flashing carat at text entry location, busy cursors (in form of hour-glass, c lock, and spinning disc) and progress bars

Animations are used for education a nd training making students see things happ en through introduction of interesting and entertaining images.

Used for data visualisation by creatin
visualisation is done using animated
In science, complex molecules and th
and viewed on the screen.
Animated characters are useful in wi

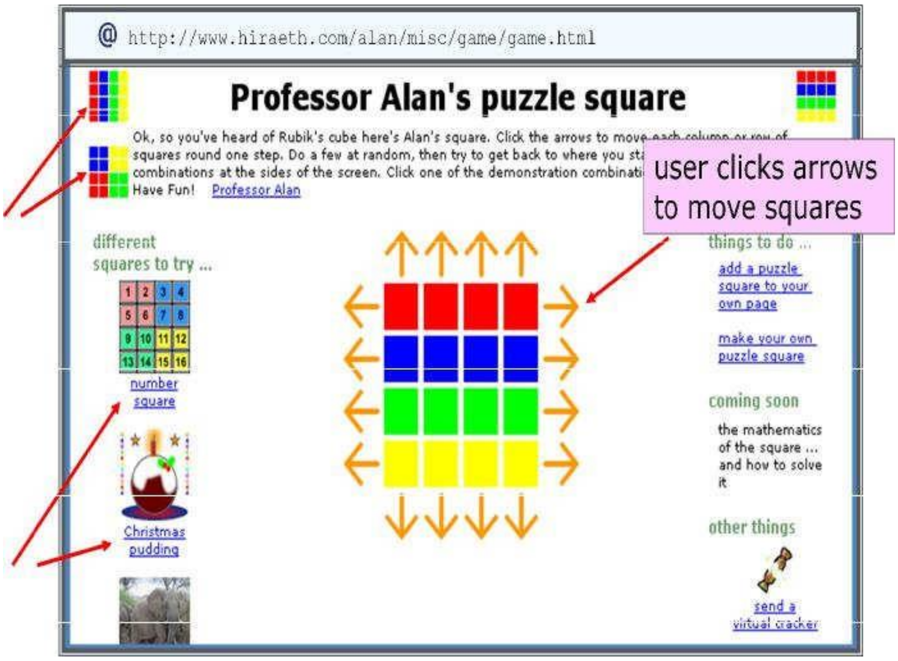
Video and Audio effects.
The current technology on improved
Tools are now available to edit soun
They are easy to embed in web page
The memory occupied could be man
It can however affect the download t
and sometimes frustrating.
It may be hard to add 'links' in hyper

Using animation and video
Animations and videos are potentiall
arcade games.
But how do we harness the full possi
In order to gain more experience fro
theorists, cartoonists, artists, and wr

3. 3 Interacting in hypertext.
Using the computer for processes, h
Illustrative interactions:

i. We search for a particula books, look for the books' puzzle square). The e-com

ii. Professor Alan's puzzle sq



iii. Delivery technology

On the computer, the help systems a re installed on hard disk with applications in CD-ROM or DVD based hypermedia

The same information can be obtain d from the web since many applications have web-based documentation. These applications c an as well be delivered on the move as long as you are connected to the internet using mobile platforms such as mobile phones, PDAs (Personal Digit al Assistants), or laptop computers.

WiFi access points or mobile phone networks containing tiny web-like pages can be used for whom and where

iv. Tourist guides and directe d advertising.

v. Rapid prototyping, creati on of live storyboards and mock-up interacti ons are done using links.

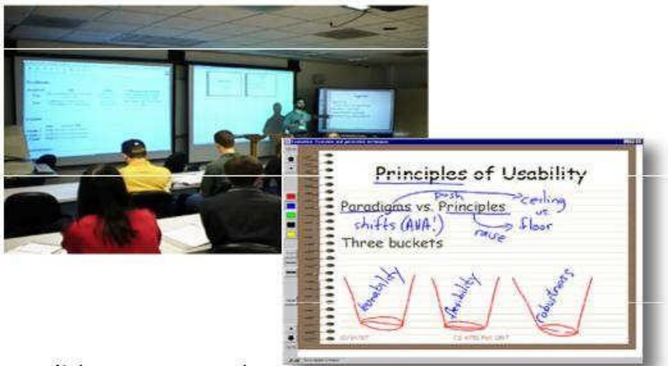
vi. The help and documentat ion allow hierarchical contents and keyword search or browsing.

vii. 'Just in time learning' (wh at you want when you want it) such as obtai ning a technical manual for a photocopier with te hnical words linked to their definition lossary are done on the web. You can as well obtain a g in links between similar photocopiers!

viii. In education, animation a nd graphics allow students to see things happen .Sound effects add atmosphere, and diagrams can be looked at while listening to explanation. The non-linear structure of the web allo s students to explore at their own pace. tion

ix. E-learning provides educa out of the classroom!!

An e-Class is shown in the picture be low



slides, pen marks, video are 'captured'

Lost ness in hyperspace

To avoid getting lost in hyperspace w hile finding information, use structure and n avigation with the aids of history, bookmarks, indices, directories and searching algorithms.

The non-linear structure of the web i s very powerful but potentially confusing.

There are two aspects of lost ness: C ognition and Content that create fragmentary information. There is also a lack of information integration thereby creating confusion.

Good design helps navigation and str ucture where hyperlinks move across struct

3.4 Designing structure:

Designing good structure entails task analysis for activities and processes in existi ng organisational structures. In making navigation easi er, create maps to give an overview of the st ructure and to show current location.

Also provide recommended routes a s guided tour or bus tour metaphor and as a linear path through a non-linear structure.

To support printing your navigated in formation, you need a linearised content th

History and bookmarks also ease nav igation because they allow 'hub and spoke' access with lots of revisiting of pages. Bookmarks and fa vourites are good for longer term revisiting. Frames are difficult to bookmark, search and link to, except there are good reasons for its use, it is not r ecommended.

Using indices, directories and search

Indexes are often found on help, doc umentation and books. Selective but non-ex haustive list of words are used in index. Directories on web ind ex would be very huge; so, manually choose site of navigation e.g. an open directory project or Yahoo. Usi g web search engines make you 'crawl' the web by following links from page to page. Search engines b uild full word index but ignore common 'sto ' words to carry out its search. It looks up your request in in dex when you enter keywords to find the pa ges.

3.5 Conducting complex search:

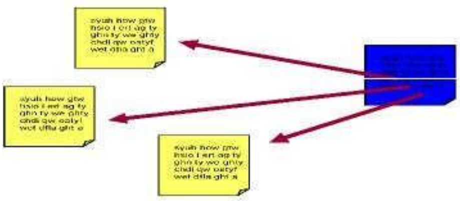
Conducting a complex search involvetoo many pages for a single word search, th ere is a need to be more selective. You can use a Boolean seach method that combines words with logic e .g. if you want to find facts about engine only, write 'enginAND NOT car' . 'AND NOT 'is the Boolean xpression.

In creating a link structure, Goggle, a nks to rank pages.

To be search engine friendly, add 'M eta' tags, relevant title, keywords and descri ption. Note that it is hard search engine, uses richness of in and out li

to index generated pages for a hidden web.

Finding research literature involves s pecial portals and search sites such as citese er<citeseer.nj.nec.com>. Searches for literature papers requir e scanning the papers for bibliography to bui ld up citation index, such as in the diagrams below.



bibliography backwards in time

es:

3.6 Web technology and web iss s, web servers, clients and a lot of networki g.

The web contains protocols, browsers and standards. Protocols such as HTTP(Hy pertext Transmission Web activities involve using protocolr the internet, HTML (Hyper-Text Mark-up anguage) , XML Protocol) that carry information ov aphics formats for content browsers to view the results, and a lot of (Extended Mark-up Language) and g plug-ins.

Changing use of the Internet: research by CERN for their high energy phys ics research. But all over Initially, the internet was created for e, government, commerce, entertainment, a nd the advertising the world, it is now used by corporat community. rspace and information overload.

The challenges remain a loss in hype

Web servers and clients:

The web is distributed with different machines far across the world. Pages are st red on servers, the browsers (the clients) ask for pages t hat are sent to and from across the internet as illustrated in the picture below:

.7 Network issues — Timing and volume of data transmission

QOS (quality of service): This term describes the quality of service provided by th enetwork.

The following comprise the quality oservice:-

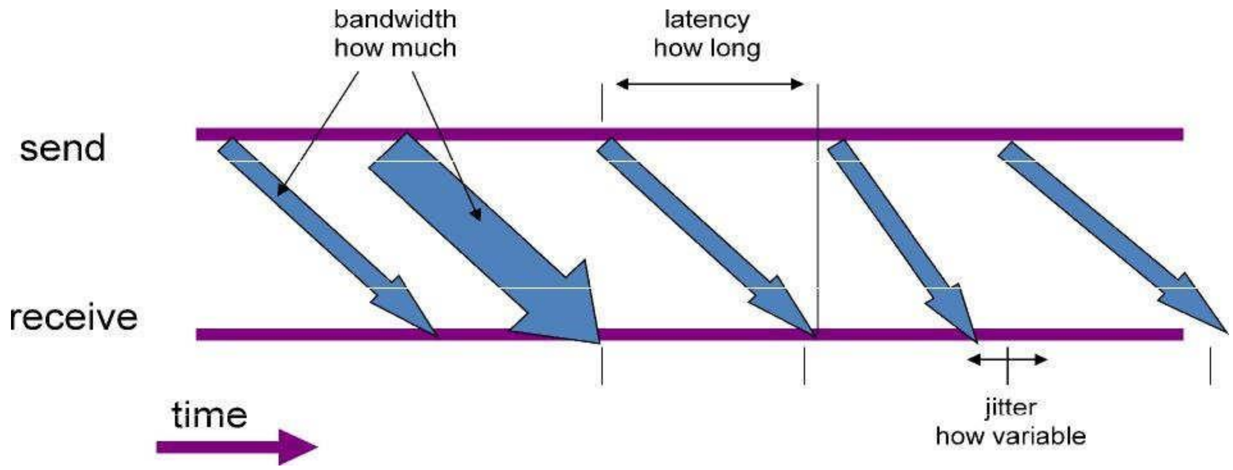
Bandwidth: This is how much of infomation transmitted per second

Latency: This is how long the transmission takes (otherwise called delay)

Jitter: This is how consistent the delay is

Reliability: Here some messages may be lost and needed to be resent. This increa ses jitter and the connection set-up, hence a need to ' handshake' to start.

The illustration of bandwidth, latenc and jitter is given in the accompanying diag ram below.



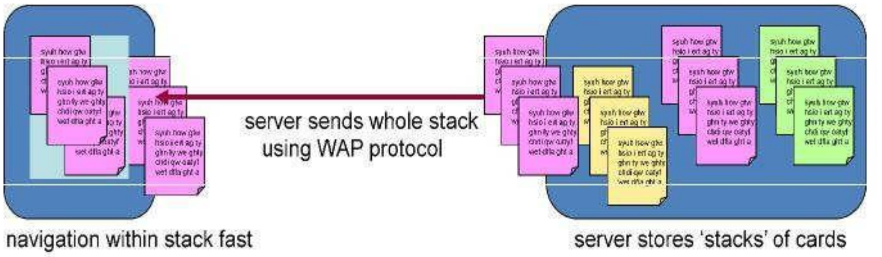
3.8 Design implications of the W b:

While designing, you should considethe bandwidth and hence about the downl oad time, e.g. an image of 100K bytes may be transmitted in 1 s ec (also called broadband), and a 56K bytes modem may transmit in 18 seconds. Hence the need to save graphics in appropriate format and size at the same graphics could be reused in the browser cach so th e after first load.

To reduce the connection time, use ne big file at a time in a data transfer than u sing several small ones. Beware of 'fit on one screen' rule be cause scrolling is fast! Think before breaking big graphic into bits.

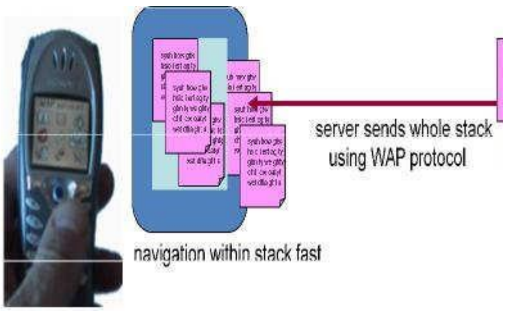
In latency, think about feedback.

3.8.1 WAP (Wireless Advance Prot col): This describes the web activities on th e phone as illustrated in the diagram below:



errata: book says stacks of 'notes', cards is correct

The phone is made up of very small screen and the scrolling of data could be very painful because the screen displays small 'pages' at a time. A GSM connection is considerably slow with big chunks. Operation is carried out using a WML (wirelessmark-up language) whose content is deliver d in 'stacks' of 'cards'. The cards are the 'pages' the user viws but navigation within the stack is fast.
N.B. With larger screens and faster connections, WML (Wireless Mark-up Language) is giving way to small HTML pages. See illustration below.



3.8.2 Static web content:
This is a medium and a message comprising text, graphics, movies and sound.
The message and the medium "cont nt is king" because it is the catch phrase of he [dot.com](dot.com) era but widely ignored.
The message content should be appropriate to the audience, should be timely, r liable, and generally worth reading!
The medium page and site design should be a good design that is essential to att act readers. A bad design may mean good material that is nev r seen.
The content should be printable!

3.8. 3 Text:
The text style should be universal generic styles such as serif, sans, fixed, bold and italic. You can use specific fonts too, but these should vary between platforms.
Use cascading style sheets (CSS) for fine control but beware of older browsers and fixed font sizes for compatibility.
Because colours are often abused, b careful about your choice of colour.
Text positioning should be easy; it could be left, right justified or centred. You should remember that DHTML (Dynamic Hypertext Mark-up Language) requires precise positioning so beware of platforms.
You should also take note of the screen size.
Remember that mathematical oriented texts require special fonts and layout.

8.4 Graphics:

Graphics should be used with care considering for example, the file size and download time. For example, the image above is made up of 1000 words of text and is affected by size, number of colours, and file format.
Add little backgrounds because too any backgrounds often make text hard to read.
Speeding up transmission of graphic require caching and to be able to reuse the same graphics.

Using progressive formats make the image to appear in low resolution and the image also gets clearer as shown above.

8.5 Formats:

Use JPEG for photographs as shown bove, for higher but 'lossy' compression.
Use GIF (Graphics Image Format) for sharp edges and lossless compression.
PNG formats are supported by current web browsers.
For action, use animated gifs for simple animations and image maps for images you can click on.

3.8.6 Icons:
Sample icons are displayed below

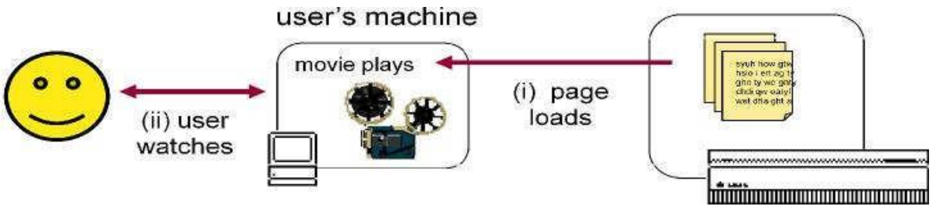
They are just small images on the web used for bullets and decoration or to link to other pages.
There are lots available!
The design of icons is just like any other interface that needs to be understood. So icons should be designed as collection to fit the web. A web site under construction is a sign of the inherent incompleteness of the web.

3.8.7 Web colour:
Decide from the beginning how many colours your web site is to contain. The PCmonitor is made to display millions of colours each comrising 24 bits per pixel but the 'same' colour may look very different.
Web colours are usually expressed a dips per inch —dpi. Between 72 to 96 dpi a e common.
Older computers, PDAs and phones can tolerate perhaps only 16 bits or 8 bits per pixel in 256 colours or even in greyscale.
From the colour palettes, you can choose up to 256 different useful colours, alth ugh Netscape 'web safe' 216 colours are common. Each GIF (raphics Image Format) used for fast download has its own palette.

.8.8 Movies and sound

The problems of size and download a re worse in movies and sound compared wi th graphics. So they require special plug-ins. The problems are not prominent with audio how ever, because some have some compact formats such as MIDI.

With streaming video, you can play hile downloading hence can be used for 'br oadcast' radio or TV (see illustration below).



Dynamic web content

This content shows what happens w ere, with its technology and security, it ena bles a local interaction, and effective search through remote access and batch generation of pages of dat a. It is of dynamic content.

The active web

In the early days of the web, pages t at contained text were static pages. Some gateways such as FTP, GOPHER enabled usability thereby m aking interaction easy on one simple model.

A dynamic content has a model or m etaphor with passive pages but active interf ce. Each leads to different user understanding.

The architectural design of the web is about what happens where. The design has (i) a feedback in which a user can see results of his own actios, (ii) a feed through effect in which effects of other people's actions are seen. Note that the effect of the design is reflected on the complexity of imp lementation and its maintenance.

The concerns of the designer from th e user's point of view are the changes to be made in the design during use in terms of the media stream, th e presentation, and the content. He should be guided by the following questions:

i. 'Are these done automatically?', ii. ' By whom would the changes be made?' - The site author, the user, or other users through a feed through? and iii. How often is the pace of change: in days, in months, or in seconds?

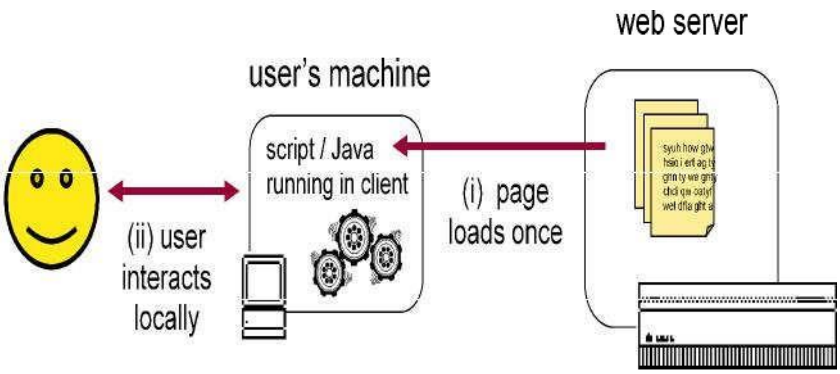
The technology of design changes:

Where does the change happen? Is i through the client using toolkits such as the applets , Flash, JavaScript and DHTML ,or through the server using server toolkits such as the CGI scripts, J va servlets , JSP, ASP, PHP, etc,? Or through another machine such as the author's machine, the databa se server, the proxy server, or through people adopting s ocio-technical solutions?

Security of the web:

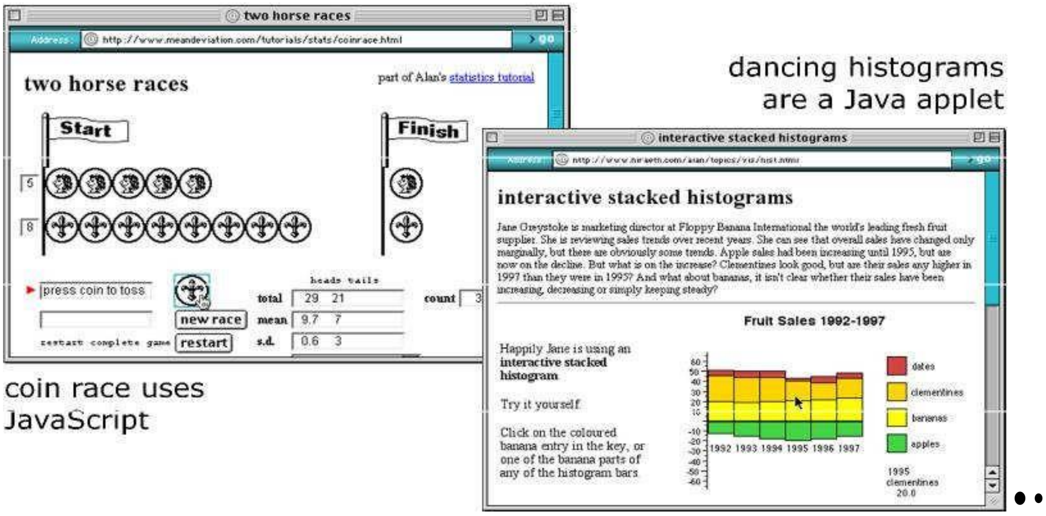
For computational functions, the codes and data should not be at the same place! The problems that need to be addressed on security include the security of data, the safety of the web-server and the client machine that is the most vulnerable, and of course the entire networks.

Local interaction at the client side: S e the illustration below



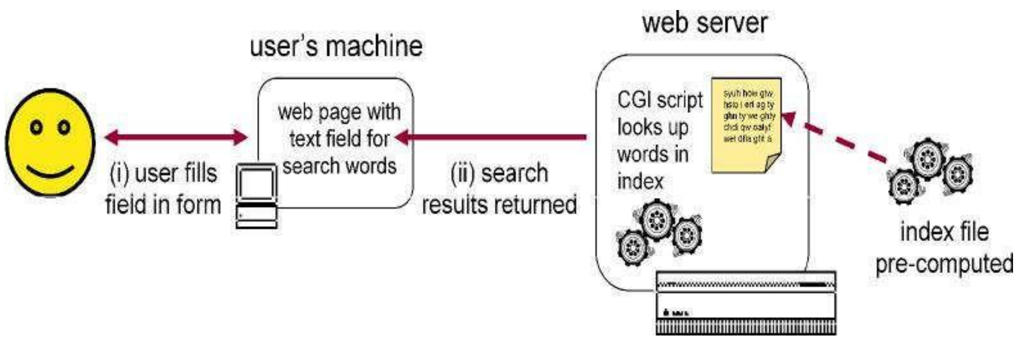
For a fixed content interaction, the u ser interacts locally through his machine by he use of Java applets, Flash, JavaScript plus DHTML (Dynam ic Hyper-Text Mark-up Language).
In fixed content interaction, a rapid f eedback occurs but interaction is local and t here is no feed through.

examples



The picture above shows that the 'co in race' uses Java Script while the 'dancing' istograms (set of interactive stacked histograms) that depict the sales trends for each fruit type, ar e Java applets.

The picture below demonstrates the processes of the user conducting a web sea ch.

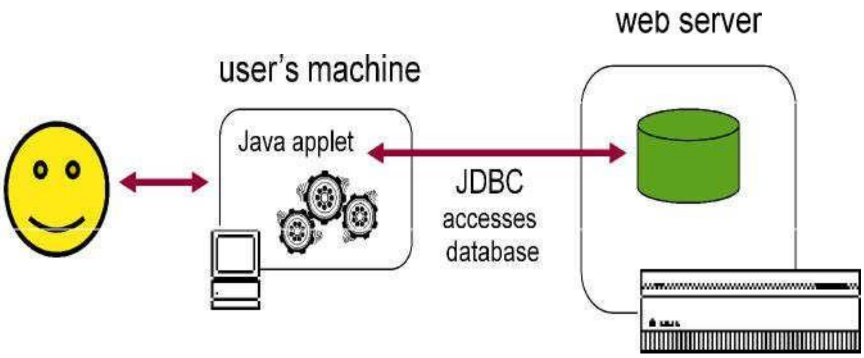


Indices are created off-line before co mmencement of search.

For automatic generation of data, yo u need database driven sites. The available ptions of tools that can be used are the client-end applet or Flash access remote DB using the server-end CGI that is driven by web forms with limited user interface.

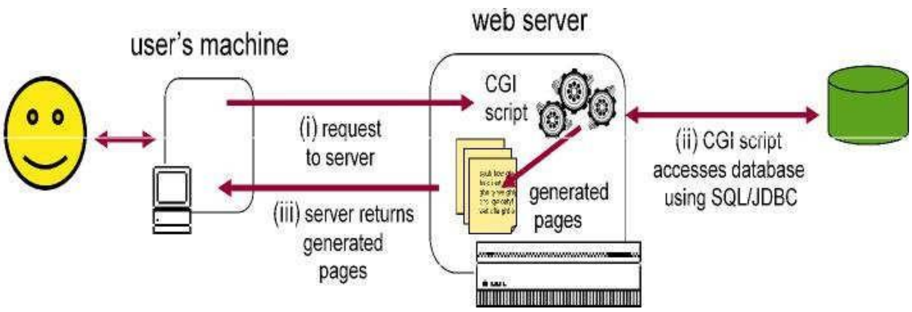
However, hybrid solutions that could be optionally utilised are the CGI generated pages that can contain JavaScript etc. The JavaScript can 'wr ite' web pages dynamically.

Look at the following picture: A situa tion of the Java applet and JDBC accessing t e database.
Java applet & JDBC



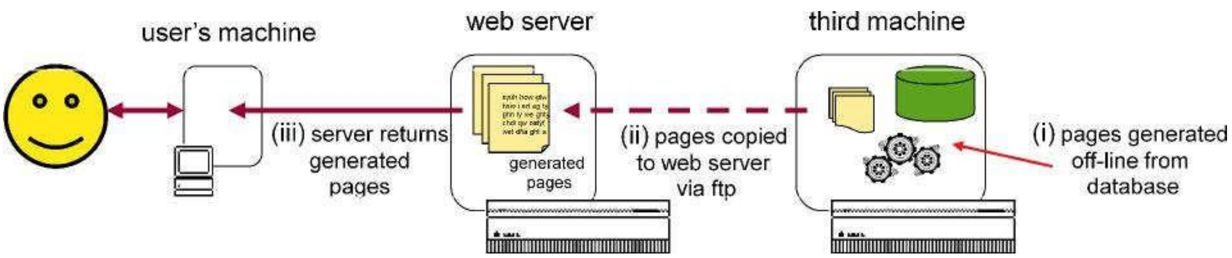
The advantage of this option is an int eractive Data Base access. However, the ava ilability of bandwidth and the security of data are problem issu es to be resolved.

The picture below depicts the situati on in which the CGI script accesses the data ase.



The advantage here is that the datab ase is always current with up to date inform ation.
Its disadvantage is the non proximity of the web server and that access is not ind ex friendly.

Batch generation of web pages of data.
Batch generation of data is for slow a nd varying data that is updated through a lo cal database. Here, pages are periodically generated before up load.
Many technologies are involved in b tch generation; and they include the use of object oriented languages such as C, Java, HyperCard and Visua l Basic.
Illustration of batch generation of w b pages



Advantages of this option are that th e generated batches are index able and secu re.
The disadvantage is its slower turnar ound.

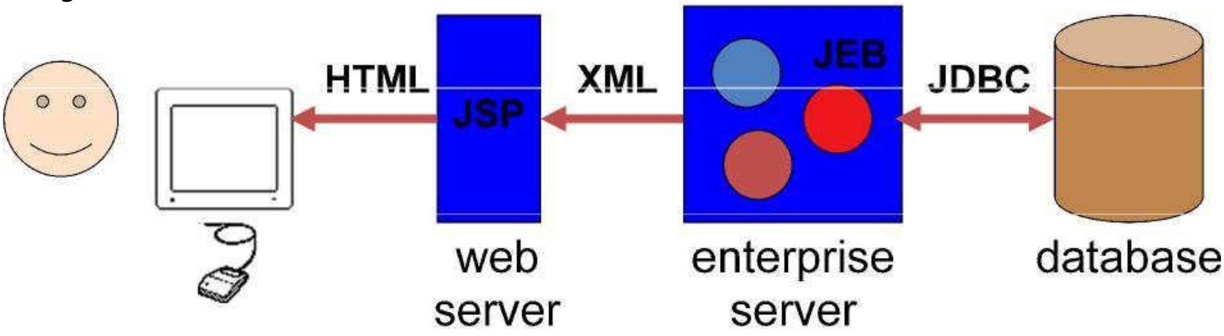
Dynamic content:

Dynamic contents are really 'active'web pages in which data is updated as well a s presented on the web. The data presentation could be made in any of the means discussed previously u sing the CGI, applet-JDBC etc.

The update is done through the webform interface using the server script that updates the data base, as in the example of that carried out in th e updating of book theatre seats.

The issues of concern in the design of this type of interaction are the authenticat on and security problems due to multiple transactions of data.

Using the n-tier architecture



As illustrated, the picture contains oe or more intermediate layers with theiness logic' in layers. The web is made up of standard com 'bus ponents and protocols.

4.0 CONCLUSION

The challenges of human computer i nteraction in hypertext, multimedia and the world-wide web remain a loss in hyperspace and information overload.

Also, the availability of bandwidth an d the security of data are problem issues to be resolved

5.0 SUMMARY

Hypertext enables you to find inform ation by navigating the hyperspace using th web technology

Multimedia is sometimes called Hypermedia and can be used for simple audio an d video effects. ch

Hypermedia is not just text but hype rtext systems containing additional media su as illustrations, photographs, video and sound.

Animation is adding motion to image s, particularly images on things that change in time.

History and bookmarks ease navigati on because they allow 'hub and spoke' acce s with lots of revisiting of pages. Bookmarks and favourites aregood for longer term revisiting

The web contains protocols, browser s, web servers, clients and a lot of networki g.

Web activities involve using protocols and standards. Protocols such as HTTP(Hy pertext Transmission Protocol) that carry information ovr the internet, HTML (Hyper-Text Mark-up Language) , XML (Extended Mark-up Language) and gaphics formats for content browsers to view the results; and a lot of plug-ins.

Wireless Advance Protocol (WAP) de scribes the web activities on the phone

Web icons are small images on the w eb used for bullets and decoration or to link to other pages.

Dynamic content are 'active' webges in which data is updated as well as prese nted on the web.

Batch generation of web pages ofta is for slow and varying data that is update d through a local da database

6.0 Tutor Marked Assignment

1 Distinguish between: (i) Hypertext and Hyper media, (ii) Hypermedia and Ani mation effects

2 What do you understand by "Video and Audio'' effects in human computer in teraction? What are their advantages and challenges arising from their application in web browsing and hypertext?

3 Give 5 examples of typical Human Computer interactions in the web.
4 What is the function of the Boolean search used in conducting a web search? Give 2 examples of Boolean expressions that can be used for a web search.
5 Describe the major components of the web technology
6 The quality of Service (QOS) of a network is reflected in the bandwidth, latency, the jitter, and reliability of the network. What do you understand of the underlined terms?
7 Mention 2 design implications of the web. Describe how these affect the web activities using the GSM phone
8 What are the design implications to be considered in the following:
(i) Static web content, (ii)Text, (iii) Graphics, (iv)Picture formats, (v)Icons, (vi)Web colour, (vii) Movies and sound,(vii) the active web.
9. Distinguish between a fixed content interaction in the web and a dynamic interaction.
What are the resource software applications utilized to carry out these interactions?
10. Using a pictorial illustration how is the batch generation of web pages carried out? What are the technologies utilized to carry out this function? What are the advantage(s) and disadvantage(s) of this option of generating data?

7.0 Further Readings / References

- Catledge, Lara D. and James E. Pitkow. "Characterizing Browsing Strategies in the World-Wide Web." Unpublished (?), 1994. [give URL?]
- Nielsen, Jakob. "Using link titles to help users predict where they are going." *Alertbox* column of January 11, 1998.
- Nielsen, Jakob. "The difference between Web design and GUI design." *Alertbox* column of May 1, 1997.
- Nielsen, Jakob. "The Tyranny of the Page: continued lack of decent navigation support in Version 4 browsers." *Alertbox* column of November 1, 1997.
- Nielsen, J., *Multimedia and Hypertext: the Internet and Beyond.* 1995, Boston: Academic Press Professional.
- van Dam, A., *et al.* "A Hypertext Editing System for the 360," in *Proceedings Conference in Computer Graphics.* 1969. University of Illinois.

COURSE CODE CIT 353
COURSE TITLE INTRODUCTION TO HUMAN COMPUTER INTERACTION

COURSE DEVELOPER JOSHUA ADEKUNLE AKINJOBI (CHF)
CRAWFORD UNIVERSITY, IGBESA,OGUN STATE, NIGERIA

TABLE OF CONTENTS

UNIT 4: DEVICES TECHNOLOGICAL PERSPECTIVE-INTERACTION STYLES AND DEVICES TECHNOLOGICAL PERSPECTIVE

MODULE 3: DESIGNS OF HUMAN-COMPUTER INTERACTION
UNIT 1: DESIGN GUIDELINES, RULES AND PRINCIPLES
UNIT 2: EVALUATION METHODS
UNIT 3: PARTICIPATORY DESIGN
UNIT 4: SYSTEM INTERACTIVE DESIGN PATTERNS

MODULE 4: DESIGN OF USER AND SYSTEM INTERFACES
UNIT 1: DESIGN OF USER INTERFACE CONCEPTS
UNIT 2: USER INTERFACE DESIGN PRINCIPLES AND CRITERIA/RATIONALE
UNIT 3: USER INTERFACE DESIGN PROGRAMMING TOOLS
UNIT 4: THE SOFTWARE DESIGN PROCESS OF HUMAN COMPUTER INTERACTION
UNIT 5: INTERACTIONS IN HYPERTEXT, MULTIMEDIA AND THE WORLD WIDE WEB

MODULE 1: SURVEY OF HUMAN-COMPUTER INTERACTION CONCEPTS, THEORIES AND PRACTICE

UNIT 1 CONCEPTS, THEORIES AND HISTORY

Table of contents

UNIT 2: SURVEY OF HUMAN COMPUTER INTERACTION PRACTICES

TABLE OF CONTENTS

UNIT 3: BASIC COMPONENTS OF HUMAN COMPUTER INTERACTION Table of contents

UNIT 4: CRITICAL EVALUATION OF COMPUTER BASED TECHNOLOGY Table of contents

MODULE 2: USER PERSPECTIVES OF HUMAN-COMPUTER INTERACTION

UNIT 1: USER ORIENTED PERSPECTIVE OF HUMAN COMPUTER INTERACTION : SOCIAL HUMAN THRUST

Table of contents

UNIT 2: USER ORIENTED PERSPECTIVE OF HUMAN COMPUTER INTERACTION : COGNITIVE HUMAN THRUST

Table of contents

UNIT 3: SYSTEM ORIENTED PERSPECTIVE OF HUMAN COMPUTER INTERACTION-IMPROVING THE USER'S TECHNOLOGICAL PERSPECTIVE

Table of contents

UNIT 4: DEVICES TECHNOLOGICAL PERSPECTIVE

INTERACTION STYLES AND DEVICES TECHNOLOGICAL PERSPECTIVE

Table of contents

MODULE 3: DESIGNS OF HUMAN-COMPUTER INTERACTION

UNIT 1: DESIGN GUIDELINES, RULES AND PRINCIPLES

Table of contents

UNIT 2: EVALUATION METHODS

Table of contents

UNIT 3: PARTICIPATORY DESIGN

Table of contents

UNIT 4: SYSTEM INTERACTIVE DESIGN PATTERNS

Table of contents

MODULE 4: DESIGN OF USER AND SYSTEM INTERFACES

UNIT 1: DESIGN OF USER INTERFACE CONCEPTS

Table of contents

UNIT 2: USER INTERFACE DESIGN PRINCIPLES AND CRITERIA/RATIONALE

Table of contents

UNIT 3: USER INTERFACE DESIGN PROGRAMMING TOOLS

Table of contents

UNIT 4: THE SOFTWARE DESIGN PROCESS OF HUMAN COMPUTER INTERACTION

Table of contents

UNIT 5: INTERACTIONS IN HYPERTEXT, MULTIMEDIA AND THE WORLD WIDE WEB

Table of contents

7.0 Further Reading/References

MODULE 1: SURVEY OF HUMAN-COMPUTER INTERACTION CONCEPTS, THEORIES AND PRACTICE

UNIT 1 CONCEPTS, THEORIES AND HISTORY

Table of contents

.0 INTRODUCTION

This unit introduces you to the basic concepts of Human-computer interaction and the theories driving it. It is meant as an overview towards appreciating the early efforts made to improve upon human computer interaction. It therefore discusses the history, and the paradigm shifts. It is meant to provide a general background for the understanding and design of Human-computer interaction.

4.0 OBJECTIVES

By the end of this unit, you should be able to

- Understand the concepts of Human-computer interaction
- Express the goals of Human-computer interaction research and study
- Explain Human-computer interaction technique
- Understand the history and paradigms of Human-computer interaction

.0 MAIN CONTENT

3.1 Definition

The following definition is given by the Association for Computing Machinery

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

Because of the interaction that occurs between users and computers at the interface of software and hardware such as between computer peripherals and large-scale mechanical systems in aircraft and power plants, human-computer interaction is the study of that interaction between people (otherwise called users) and computers. It can also be regarded as the intersection of computer science, behavioral sciences, design and several other fields of study.

3.2 Overview

Since human-computer interaction studies a human and a machine in conjunction, it draws from supporting knowledge on both the machine and the human side. On the machine side, techniques in computer graphics, operating systems, programming languages, and development environments are relevant. On the human side, communication theory, graphic and industrial design disciplines, linguistics, social sciences, cognitive psychology, and human performance are relevant. Engineering and design methods are also relevant.

The multidisciplinary nature of HCI enables people with different backgrounds contribute to its success. HCI is also sometimes referred to as man—machine interaction (MMI) or computer—human interaction (CHI).

2.1 The goals of HCI Studies:

A basic goal of HCI study is to improve the interactions between users and computers by making computers more usable and receptive to the user's needs in the following ways:

- Methodologies and processes for designing interfaces in their related styles (i.e., given a task and a class of users, design the best possible interface within given constraints, optimizing for a desired property such as learn ability or efficiency of use)
- Methods or techniques for implementing interfaces (e.g. software toolkits and libraries; efficient algorithms)
- Techniques for evaluating and comparing interfaces
- Developing new interfaces and interaction techniques
- Developing descriptive and predictive models and theories of interaction
- Design systems that minimize the barrier between the human's cognitive model of what they want to accomplish and the computer's understanding of the user's task.

2.2 Research

Part of research in human-computer interaction involves exploring easier-to-learn or more efficient interaction techniques for common computing tasks. This includes inventing new techniques and comparing existing techniques using the scientific method as follows:

1. Designing graphical user interfaces and web interfaces.
2. Developing new design methodologies,
3. Experimenting with new hardware devices,
4. Prototyping new software systems,
5. Exploring new paradigms for interaction, and
6. Developing models and theories of interaction.

. 3 Interaction technique

An interaction technique or user interface technique is a combination of input and output consisting of hardware and software elements that provides a way for computer users to accomplish a simple task. For example, one can go back to the previously visited page on a Web browser by either clicking a button, hitting a key, performing a mouse gesture or uttering a speech command.

The computing perspective of interaction technique:
Here, an interaction technique involves one or several physical input devices, including a piece of code which interprets user input into higher-level commands, possibly producing user feedback and one or several physical output devices.
Consider for example, the process of deleting a file using a contextual menu. This first requires a mouse and a screen (input/output devices). Then, a piece of code needs to paint the contextual menu on the screen and animate the selection when the mouse moves (user feedback). The software also needs to send a command to the file system when the user clicks on the "delete" item (interpretation).

The user view of interaction technique:
Here, an interaction technique is a way to perform a simple computing task and can be described by the way of instructions or usage scenarios. For example "right-click on the file you want to delete, then click on the delete item".

The conceptual view of interaction technique:
Here, an interaction technique is an idea and a way to solve a particular user interface design problem. It does not always need to be bound to a specific input or output device. For example, menus can be controlled with many sorts of pointing devices.
Interaction techniques as conceptual ideas can be refined, extended, modified and combined. For example, pie menus are a radial variant of contextual menus. Marking menus combine pie menus with gestures. In general, a user interface can be seen as a combination of many interaction techniques, some of which are not necessarily widgets.

3.4 Interaction styles

Interaction techniques that share the same metaphor or design principles can be seen as belonging to the same interaction style. Examples are command line and direct manipulation user interfaces.
More details are provided in subsequent chapter of this guide.

3.5 Paradigms and History

Paradigms are predominant theoretical frameworks or scientific world views such as the Aristotelian, Newtonian, and Einsteinian (relativistic) paradigms in physics
Understanding HCI history is largely about understanding a series of paradigm shifts
The study of paradigms is concerned about how an interactive system is developed to ensure its usability and how that usability can be demonstrated or measured.
The history of interactive system design also provides paradigms for usable designs

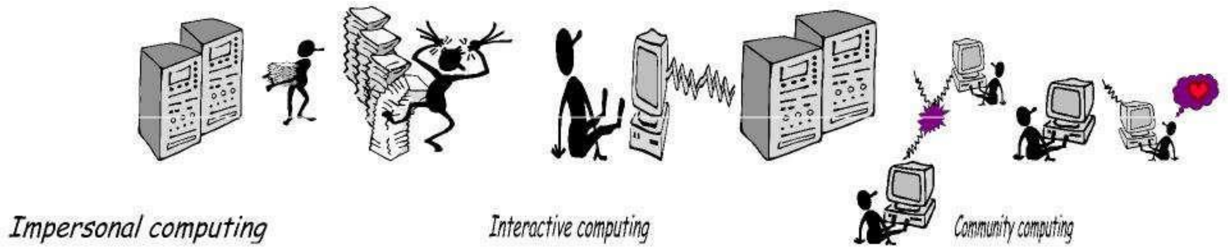
3.5.1 Paradigms of interaction

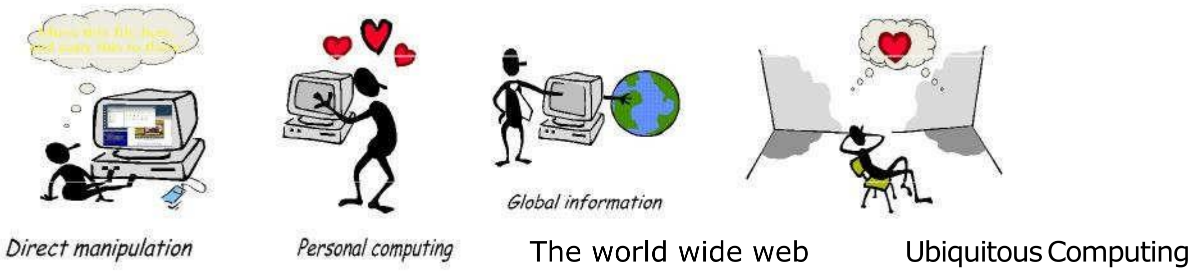
Paradigms of interaction conceptually outline the arrival of new technologies creating a new perception of the human-computer relationship.
Some of these paradigms shifts can be traced in the history of interactive technologies as follows:.
Batch processing
Timesharing
Networking
Graphical display
Microprocessor
World Wide Web (WWW)

Ubiquitous computing

The initial paradigm started with batch processing that signified impersonal com uting.
The paradigm shifts commenced from timesharing processing system that signified an interactive computing.
This was followed by another paradigm shift in networking that represented a community computing.
The graphical display was an innovation whose era indicated a paradigm shift to direct manipulation of devices.
The Micro processor innovation provided opportunity for personal computing as an example of another paradigm shift.
The World Wide Web (www) turned the world into a global village by creating environment for a global information access and transmission This represents a significant paradigm shift in the life of educated humans all over the world.
Ubiquitous computing can be regarded as another paradigm shift as it is presently a symbiosis of physical and electronic worlds in service of everyday activities.

Pictorial representations of the shifts are illustrated below:





The linked image cannot be displayed. The file may have been moved, renamed, or deleted. Verify that the link points to the correct file and location.

Figure 1: Approximate time lines showing where work was performed on some major technologies. .5.2 The History of Paradigm Shifts

Time-sharing:
1940s and 1950s witnessed explosive technological growth in computing and in 1960s , there was the need to channel the power.
Hence J.C.R. Licklider at ARPA introduced the single computer that supported multiple users.

Video Display Units:
Video Display Units provided more suitable medium than paper and so in 1962, Sutherland introduced the Sketchpad computers for visualizing and manipulating data. So, one person's contribution drastically changed the history of computing.

Programming toolkits
Engelbart at Stanford Research Institute in 1968 augmented man's intellect by demonstrating the NLS Augment system. This became the right programming toolkit that provided the building blocks to produce complex interactive systems.

Personal computing:
The era of personal computing came on board in 1970s with the introduction of the Papert's LOGO language for simple graphics programming by children. This system became popular as it became easier to use.
The era of computing in small but powerful machines dedicated to the individual was witnessed such as was demonstrated by Kay at Xerox PARC with the Dyna-book as the ultimate personal computer.

The Window systems and the WIMP interface:
The Window systems and the WIMP interface enabled humans to pursue more than one task at a time such as in the windows used for dialogue partitioning, to "change a topic". This became a reality in 1981

with the Xerox Star as the first commercial windowing system comprising windows, icons, menus and pointers (WIMPs) as familiar interaction mechanisms.

Direct manipulation

In 1982, Shneiderman improved upon direct manipulation of objects on the computer by introducing a graphically-based interaction of visibility of objects. This provided incremental action and rapid feedback, its reversibility facility encouraged exploration and syntactic correctness of all actions. It replaced language with action.

In 1984 using the Apple Macintosh , the model-world metaphor 'What You See Is What You Get (WYSIWYG) ' became popular.

This related Language with action to confirm that actions do not always speak louder than words! The direct manipulation interface replaced underlying system of language paradigm and interface as mediator. The interface acted as the intelligent agent since programming by example is both action and language.

Hypertext

In 1945, Vannevar Bush and the Memex gave the computing world the key to success in managing explosion of information by introducing the hypertext. In mid 1960s, Ted Nelson described hypertext as non-linear browsing structure.

Within the same period, Nelson Xanadu started a project on hypermedia and multimedia; this gave bedrock for research in this area.

In the World Wide Web, the Hypertext, as originally realized, was a closed system. It comprises simple, universal protocols (e.g. HTTP) and mark-up languages (e.g. HTML) that made publishing and accessing easy. It allowed emancipation of critical mass of users that led to a complete transformation of our information economy.

Applying hypertext technology to browsers allows one to traverse a link across the world with a click of the mouse.

.5. 3 History of Basic Interactions

The Mouse: The mouse was developed at Stanford Research Laboratory in 1965 as part of the NLS project to be a cheap replacement for light-pens, which had been used at least since 1954. Many of the current uses of the mouse were demonstrated by Doug Engelbart as part of NLS in a movie created in 1968

Drawing programs: Much of the current technology was demonstrated in Sutherland's 196 3 Sketchpad system. The use of a mouse for graphics was demonstrated in NLS (1965). In 1968 Ken Pulfer and Grant Bechthold at the National Research Council of Canada built a mouse out of wood patterned after Engelbart's and used it with a key-frame animation system to draw all the frames of a movie.

Text Editing: In 1962 at the Stanford Research Lab, Engelbart proposed, and later implemented a word processor with automatic word wrap, search and replace, user-definable macros, scrolling text, and commands to move, copy, and delete characters, words, or blocks of text. Xerox PARC's Bravo was the first WYSIWYG editor-formatter developed in 1974. It was designed by Butler Lampson and Charles Simonyi who had started working on these concepts around 1970 while at Berkeley. The first commercial WYSIWYG editors were the Star, LisaWrite and then MacWrite.

Spreadsheets: The initial spreadsheet was VisiCalc which was developed by Frankston and Bricklin between 1977 and 1978 for the Apple II while they were students at MIT and the Harvard Business School.

The solver was based on a dependency-directed backtracking algorithm by Sussman and Stallman at the MIT AI Lab.

Computer Aided Design (CAD): The same 196 3 IFIPS conference at which Sketchpad was presented also contained a number of CAD systems, including Doug Ross's Computer-Aided Design Project at MIT in the Electronic Systems Lab and Coons' work at MIT with SketchPad. Timothy Johnson's pioneering work on the interactive 3D CAD system Sketchpad 3 was his 196 3 MIT MS thesis.

Video Games: The first graphical video game was probably SpaceWar by Slug Russel of MIT in 1962 for the PDP-1 including the first computer joysticks. The early computer Adventure game was created by Will Crowther at BBN, and Don Woods developed this into a more sophisticated Adventure game at Stanford in 1966 Conway's game of LIFE was implemented on computers at MIT and Stanford in 1970.

Gesture Recognition: The first pen-based input device, the RAND tablet, was funded by ARPA. Sketchpad used light-pen gestures (196 3). Teitelman in 1964 developed the first trainable gesture recognizer. A very early demonstration of gesture recognition was Tom Ellis' GRAIL system on the RAND tablet in 1964,. It was quite common in light-pen-based systems to include some gesture recognition. A gesture-based text editor using proof-reading symbols was developed at CMU by Michael Coleman in 1969. Gesture recognition has been used in commercial CAD systems since the 1970s
.
Multi-Media: The FRESS project at Brown used multiple windows and integrated text and graphics in 1968. The Interactive Graphical Documents project at Brown was the first hypermedia (as opposed to hypertext) system, and used raster graphics and text, but not video between 1979 and 198 3.. The Movie Manual at the Architecture Machine Group (MIT) was one of the first to demonstrate mixed video and computer graphics in 198 3

3-D: The first 3-D system was probably Timothy Johnson's 3-D CAD system in 196 3. The "Lincoln Wand" by Larry Roberts was an ultrasonic 3D location sensing system, developed at Lincoln Labs in 1966. That system also had the first interactive 3-D hidden line elimination. An early use was for molecular modeling. Also, the military-industrial flight simulation work of between the 60's and the 70's led the way to making 3-D real-time with commercial systems from some firms.

Virtual Reality and "Augmented Reality": The original work on VR was performed by Ivan Sutherland when he was at Harvard between 1965 and 1968.

Computer Supported Cooperative Work. Doug Engelbart's 1968 demonstration of NLS included the remote participation of multiple people at various sites. Electronic mail, still the most widespread multi-user software, was enabled by the ARPAnet, which became operational in 1969, and by the Ethernet from Xerox PARC in 197 3. An early computer conferencing system was Turoff's EIES system at the New Jersey Institute of Technology in 1975.

Natural language and speech: The fundamental research for speech and natural language understanding and generation has been performed at CMU, MIT, SRI, BBN, IBM, AT&T Bell Labs and BellCore, much of it government funded.

Software Tools and Architectures
The area of user interface software tools is quite active now, and many companies are selling tools. Most of today's applications are implemented using various forms of software tools.

UIMSs and Toolkits: The first User Interface Management System (UIMS) was William Newman's Reaction Handler created at Imperial College, London between 1966 and 1967. Most of the early work was done at the university of Toronto, George Washington University and Brigham Young University. Early window managers such as Smalltalk developed in 1974 and InterLisp, both from Xerox PARC, came with a few widgets, such as popup menus and scrollbars. The Xerox Star of 1981 was the first commercial system to have a large collection of widgets. The Apple Macintosh (1984) was the first to actively promote its toolkit for use by other developers to enforce a consistent interface.

Interface Builders: These are interactive tools that allow interfaces composed of widgets such as buttons, menus and scrollbars to be placed using a mouse. The Steamer project at BBN carried out between 1979 and 1985 was probably the first object-oriented graphics system. Trillium was developed at Xerox PARC in 1981. Another early interface builder was the MenuLay system and was developed by Bill Buxton at the University of Toronto in 198 3. The Macintosh in 1984 included a "Resource Editor" which allowed widgets to be placed and edited. Jean-Marie Hullot created "SOS Interface" in Lisp for the Macintosh while working at INRIA in 1984.

Component Architectures: The idea of creating interfaces by connecting separately written components was first demonstrated in the Andrew project by Carnegie Mellon University's Information Technology Center in 198 3. It is widely popularized by Microsoft's OLE and Apple's OpenDoc architectures.

Multimodality

A mode is a human communication channel. Hence multimodality places emphasis on simultaneous use of multiple channels for input and output.

Computer Supported Cooperative Work (CSCW)

The Computer Supported Cooperative Work (CSCW) removes bias of single user with single computer system but one can not neglect the social aspects.

Electronic mail is most prominent success of Computer Supported Cooperative Work.

Agent-based Interfaces

Agent-based Interfaces are original interfaces with commands given to computer and it is language-based. It involves direct manipulation using the WIMP interface. Commands are performed on "world" representation

and it is aaction based.

The agents return to the language by instilling proactively and "intelligence" in command processor. Example is found in Avatars, a natural language processor.

Ubiquitous Computing:

One example of ubiquitous computing is the ubiquitous graphical interface used by Microsoft Windows 95, which is based on the Macintosh, which is based on work at Xerox PARC, which in turn is based on early research at the Stanford Research Laboratory (now SRI) and at the Massachusetts Institute of Technology. Virtually all software written today employs user interface toolkits and interface builders, concepts which were developed first at universities. The spectacular growth of the World-Wide Web is also a direct result of HCI research. Interface improvements more than anything else has triggered this explosive growth. Computers will communicate through high speed local networks, nationally over wide-area networks, and portably via infrared, ultrasonic, cellular, and other technologies. Data and computational services will be portably accessible from many if not most locations to which a user travels.

Sensor-based and Context-aware Interaction

Humans are good at recognizing the "context" of a situation and reacting appropriately. They are also good at automatically sensing physical phenomena (e.g., light, temperature, location, identity) becoming easier Sensors utilized the concept of senses of physical measures to interactions that behave as if made "aware" of the surroundings.

Metaphor

The LOGO's turtle dragging its tail enabled an effective teaching technique with file management on an office desktop, word processing for typing, and financial analysis using the spreadsheets

The problems with metaphors are that some tasks do not fit into a given metaphor while some can be culturally biased.

4.0 Conclusion

The study of paradigms is concerned about how an interactive system is developed to ensure its usability and how that usability can be demonstrated or measured.

The history of interactive system design also provides paradigms for usable designs

Paradigms of interaction conceptually outline the arrival of new technologies creating a new perception of the human-computer relationship. Understanding Human-computer interaction history is largely about understanding a series of paradigm shifts.

Some of these paradigms shifts can be traced in the history of interactive technologies as outlined.

5.0 Summary

Human Computer Interaction is the interaction between computer users and its interface of software and hardware.

Its study requires the knowledge of computer graphics, operating systems, programming languages, cognitive psychology, and human performance among others.

The study of HCI paradigms concerns its development while the history concerns the understanding of the paradigm shifts. Paradigms are predominant theoretical frameworks or scientific world views.

The unit has looked at the concepts and techniques of HCI, the history of paradigm shifts, the history of basic interactions and the personalities behind the innovations of HCI.

6.0 Tutor Marked Assignment

6. What do you understand by the expression Human Computer Interaction?
7. Mention any 5 scientific methods of conducting research in human-computer interaction.
8. Distinguish between an interaction technique and an interaction style.
9. Explain what you understand as the paradigms of human Computer interaction
10. Explain any 5 innovations relating to the history of paradigm shifts

7.0 Further Reading/References

- Andrew Sears and Julie A. Jacko (Eds.). (2007). Handbook for Human Computer Interaction (2nd Edition). CRC Press. ISBN 0-8058-5870-9
- Julie A. Jacko and Andrew Sears (Eds.). (200 3). Handbook for Human Computer Interaction. Mahwah: Lawrence Erlbaum & Associates. ISBN 0-8058-4468-6
- Stuart K. Card, Thomas P. Moran, Allen Newell (198 3): *The Psychology of Human—Computer Interaction*. Erlbaum, Hillsdale 198 3 ISBN 0-89859-24 3-7
- Brad A. Myers: *A brief history of human—computer interaction technology.* Interactions 5(2):44-54, 1998, ISSN 1072-5520 ACM Press. http://doi.acm.org/10.1145/2744 30.2744 36

UNIT 2: SURVEY OF HUMAN COMPUTER INTERACTION PRACTICES

TABLE OF CONTENTS

1.0 INTRODUCTION

Nature of Interactivity

Long ago going down memory lane, 'remote computer interaction' was done through *batch* processing involving punched card stacks or large data files prepared with long wait for the line printer output. And if it is not right the wait continued indefinitely ...

But now most computing is 'truly' interactive with rapid feedback and the user is in control most of the time with the thinking taken over by the computer.

A typical computer system interaction is carried out through input devices such as: the screen or monitor, keyboard, mouse or track pad.

The input devices exist in variations of desktop, laptop, mainframe computers and Personal Digital Assistants (PDAs). The devices dictate the styles of interaction that the system supports.

If we use different devices, then the interface will support a different style of interaction.

This unit sets out to remind you of the various nature of human-computer interaction that you may have normally come across.

2.0 OBJECTIVES

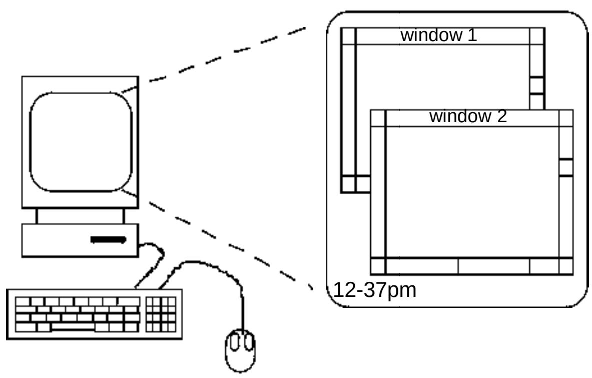
By the end of this unit, you should be able to:

- Understand handwriting and speech recognition
- Identify positioning, printing and drawing devices
- Know display devices
- Recognise physical controls and sensors
- Differentiate between printers and scanners
- Know memory and storage formats

.0 MAIN CONTENT

.1 Overview of the Computer System

The computer system is made up of various elements and each of these elements affects the interaction in the following manner:
Input devices are used for text entry and pointing.
Output devices are used for display and print of processed data on Visual Display Unit(screen) and printer on digital paper
Virtual reality affected using special interaction and display devices
Physical interaction carried out through sound, haptic, and bio-sensing devices
Paper is used for printing output and for scanning inputs.
Storage and memory utilised through accessing large capacity Random Access Memory (RAM) and permanent storage media
Processing carried out using high speed processing units and networks
See a pictorial representation below:



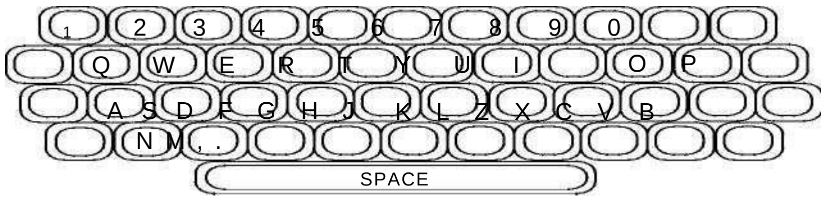
You can therefore consider the types and variations of computers that help operate your devices as follows:
In your house, these include
Personal Computers
TV, VCR, DVD, HiFi, Cable/satellite TV
Microwave, Cooker, Washing Machine
Central heating system
Security system
can you think of more variations ?

And then portable ones that can be put in your pockets
Personal Digital Assistants
Phone, Camera
Smart card and card with magnetic strip
Electronic car key with automatic opening and closing of doors
USB memory

.2 TEXT ENTRY DEVICES

Most of these common text input devices allow rapid entry of text by experienced users and usually connected by cable but can also be wireless.
The richer interaction is enabled through faster text entry devices using 'QWERTY' keyboards , chord keyboards and phone pads

QWERTY layout



Standardised layout — QWERTY keyboard with non-alphanumeric keys are placed differently because accented symbols are needed for different scripts.
An example of QWERTY arrangement is shown above:
This QWERTY arrangement is not optimal for typing because initial layout was to prevent typewriters jamming! Alternative designs however allow faster typing but large social base of QWERTY typists produces reluctance to change.
Alternative keyboard layouts introduced later, have the following characteristics:
Alphabetic:
Here the keys are arranged in alphabetic order but are however neither faster for trained typists nor for beginners too!

Dvorak:
This has common letters under dominant fingers with bias towards right hand.
Common combinations of letters alternate between hands resulting in 10-15% improvement in speed and reduction in fatigue.
Expectedly, large social base of QWERTY typists produce market pressures not to change.

Special keyboards:
These are designed to reduce fatigue for RSI and also for one handed use.

Example is the Maltron left-handed keyboard shown above.

Chord keyboards:
These have only a few keys - four or five in number. Letters are typed as combination of key presses; the key presses reflect the letter shape.

Its compact size makes it ideal for po rtable applications.
It has a short learning time, and it is ast once you have trained.
However, social resistance and fatigu e creep in after extended use.

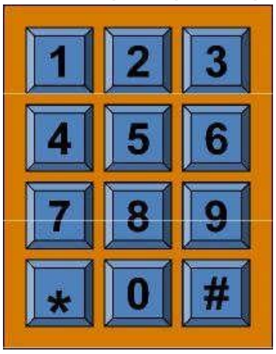
Phone pad and T9 entry:

These use numeric keys with multipl e presses as shown above and keys extracte and shown below

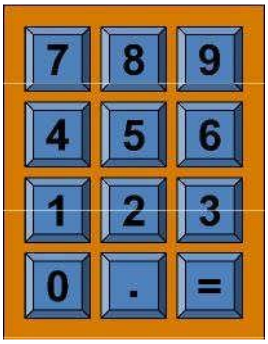
2 — a b c 6 - m n o
3- d e f 7 - p q r s
4 - g h i 8 - t u v
5 - j k l 9 - w x y z

T9 predictive entry:
This allows typing as if single key for each letter. It uses dictionary to 'guess' the r ight word

Numeric keypads
These are also developed for easier uman computer interaction since they prov ide the means of entering numbers quickly. They could be foun d in calculators, PC keyboards and telephon s.



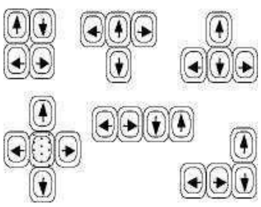
Telephone



Calculator

If you compare your phone keyboard with your calculator keyboard, you would n otice the difference as shown above.

Cursor keys



There are four keys (up, down, left, right) on keyboard.
The keys are very, very cheap, but slow and provide basic motion for text-editing tasks.
There is no standardised layout, but inverted "T" that is most common with discrete positioning controls.

. 3 Handwriting, Speech Recognition and other Devices

3. 3.1 Handwriting Recognition.

Another initiative that improves human interaction is Handwriting recognition.
Here text can be input into the computer, using a pen and a digesting tablet thereby producing natural interaction
However, this generates the following technical problems:
Difficulties may be experienced while capturing all useful information in a natural manner and segmenting joined up writing into individual letters.
Also difficult is interpreting individual letters and coping with different styles of handwriting.
They are commonly used in PDAs (Personal Digital Assistants), and tablet computers.

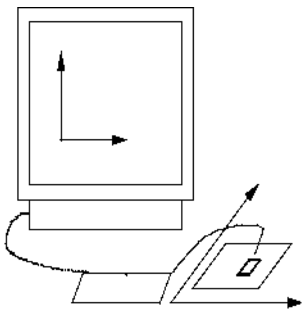
3. 3.2 Speech Recognition:

This provides the advantage of leaving the keyboard on the desk, doing some other thing and talking to the computer!
The speech recognition is improving rapidly and is most successful when a single user has initial training and learns the peculiarities of limited vocabulary systems
The speech recognition system may have problems with
External noise interfering, imprecision of pronunciation, large vocabularies, and variation effects due to different speakers.

Other devices:
Other classes of human interaction devices are the positioning, pointing and drawing devices that include the mouse, touchpad, trackballs, joysticks, touch screens, eye gaze and cursors.

.4 Positioning, Pointing and Drawing Devices

The Mouse
This is a human computer interaction medium and is a handheld pointing device that is very common and easy to use. It has two characteristics, the planar movement and the buttons. Usually from 1 to 3 buttons on top, used for making a selection, indicating an option, or to initiate drawing etc. The mouse is located on desktop and requires no large physical space and no arm fatigue.



Relative movement only is detectable which moves the screen cursor. The screen cursor is oriented in x and y plane while the mouse movement is in x and z plane ...

It is an *indirect* manipulation device that does not obscure screen, it is accurate and fast.
There are two methods for detecting its motion
Mechanical

- Ball on underside of mouse turns as mouse is moved rotating orthogonal potentiometers
- It can be used on almost any flat surface

Optical

- Here light emitting diode is located on underside of mouse
- It may use special grid-like pad placed on desk
- It is less susceptible to dust and dirt
- It detects fluctuating alterations in reflected light intensity to calculate relative motion in (x, z) plane
- It can be used even by foot using foot *mouse similar to* car pedals, sewing machine speed control, organ and piano pedals

Trackball and thumbwheels
A Trackball has a ball rotated inside a static housing like an upside down mouse!
Its relative motion moves the cursor. It is an indirect device and fairly accurate using separate buttons for picking. It is very fast for gaming and used in some portable and notebook computers.
Thumbwheels are used for accurate Computer Aided Design (CAD). It has two dials for X-Y cursor positioning. For fast scrolling , a single dial is used on mouse.

Joystick and keyboard nipple
Joystick has an indirect pressure of stick that equals the <u>velocity</u> of movement.
It has buttons for selection on top or on front like a trigger.
It is often used for computer games, aircraft controls and 3D navigation
A Keyboard nipple is useful for laptop computers. It has a miniature joystick in the middle of the keyboard.

Touchpad
These are small touch sensitive tablets with 'stroke' to move mouse pointer used mainly in laptop computers. It has 'acceleration' settings in form of fast stroke with lots of pixels per inch moved and initial movement to the target
Other types have slow stroke with less pixels per inch for accurate positioning.

Touch-sensitive screen
A touch-sensitive screen detects the presence of finger or stylus on the screen. It works by interrupting matrix of light beams, capacitance changes or ultrasonic reflections. It is a *direct* pointing device.

- Advantages are:
  - It is fast, and requires no specialised pointer
  - It is good for menu selection
  - It is suitable for use in hostile environment; being clean and safe from damage.
- Disadvantages:
  - Finger can mark the screen
  - It could be imprecise because the finger is a fairly blunt instrument!, hence difficult to select small regions or perform accurate drawing
  - The user lifting his arm can be tiring

Stylus and light pen
Stylus is a small pen-like pointer to draw directly on screen which may use touch sensitive surface or magnetic detection.
It is used in PDA, tablets PCs and drawing tables
Light Pen is now rarely used but uses light from screen to detect location.

Both stylus and light pen are very direct and obvious to use but they can obscure the screen.

Digitizing tablet
This is a mouse like-device with cross hairs used on special surface and it is rather like stylus.
It is very accurate and used for digitizing maps

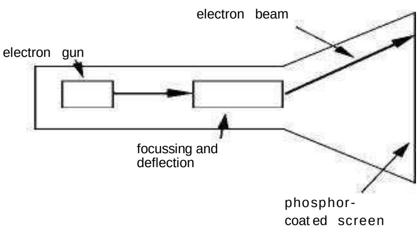
Eye gaze
This controls interface by eye gaze direction such as looking at a menu item to select it.
It uses laser beam reflected off retina at a very low power laser!
It is mainly used for evaluation and has potential for hands-free control.
Its high accuracy requires headset. The cheaper and lower accuracy devices are available under the screen like a small web cam.

Discrete positioning controls
These can be found in phones, TV controls etc.
They have cursor pads or mini-joysticks with discrete left-right and up-down movement used mainly for menu selection. See below.

3.5 Display devices
These are bitmap screens Cathode Ray Tube (CRT) and Liquid Crystal Displays (LCD).There are also large and situated displays.

Bitmap Screen displays
The screen contains vast number of coloured dots with the following resolution and colour depth:
Resolution: This is the number of pixels on screen (width x height), for example SVGA has 1024 x 768, PDA has around 240 x 400.
Density of pixels (in pixels or dots per inch - dpi) is typically between 72 and 96 dpi
Aspect ratio: This is the ratio between width and height, for example ratio 4 to 3 for most screens, ratio 16 to 9 for wide-screen TV.
Colour depth: This expresses the number of different colours for each pixel. e.g black and white or greys only, 256 from a palette, 8 bits each for red, green, and blue contains millions of colours.

Cathode Ray Tube (CRT)



Stream of electrons emitted from electron gun, focused and directed by magnetic fields, hit phosphor-coated screen which glows and used in TVs and computer monitors.

Health hazards of CRT!
X-rays are largely absorbed by the screen though not from the rear.
Ultra Violet and Intra Red radiations emanate from phosphors in the tube but at insignificant levels.

Radio frequency emissions, plus ultrasound about 16kHz, are common.
Electrostatic field leaks out through tube to the user. The Intensity is dependent on distance and humidity and this can cause rashes to the user.
Electromagnetic fields between 50Hz and 0.5MHz create induction currents in conductive materials, including the human body. Two types of effects are attributed to this: effect on visual system with high incidence of cataracts in VDU operators, and concern over reproductive disorders such as miscarriages and birth defects.

Health hints ...
Do not sit too close to the screen
Do not use very small fonts
Do not look at the screen for long periods without a break
Do not place the screen directly in front of a bright window
Work in well-lit surroundings
Take extra care if pregnant.
Also watch your posture, ergonomics and stress while using the system.

Liquid Crystal Displays (LCD)
This is smaller, lighter, and without radiation problems.
They are found on PDAs (Personal Digital Assistants), portables and notebooks, and increasingly on desktop and even home TV.
It is also used in dedicated displays such as in digital watches, mobile phones, High Fidelity (HiFi) controls etc.
How it works ...
Top plate is transparent and polarised. Bottom plate reflects the light that passes through the top plate and crystal, and reflects back to user's eye. The voltage applied to the crystal changes polarisation and hence the colour.
The light from the display is reflected and not emitted and so causes less eye strain.

Special displays of LCD
Random Scan comprising directed-beam refresh and vector display
These draw the lines to be displayed directly without jaggies . The lines need to be constantly redrawn.
They are however rarely used except in special instruments

Direct view storage tube (DVST)
- Similar to random scan but persistent => no flicker
- Can be incrementally updated but not selectively erased
- Used in analogue storage oscilloscopes

large displays
- used for meetings, lectures, etc.
- technology
  - plasma — usually wide screen
  - video walls — lots of small screens together
  - projected — RGB lights or LCD projector
    - hand/body obscures screen
    - may be solved by 2 projectors + clever software
  - back-projected
    - frosted glass + projector behind

Direct view storage tube (DVST)

Similar to random scan but persistent and has no flicker. It can be incrementally updated but not selectively erased
It is used in analogue storage oscilloscopes .
Has large displays used for meetings, lectures, etc.
The technology comprises plasma that is usually a wide screen. The video walls contain lots of small screens together and has projected RGB lights or LCD projector.

Situated displays
These are displays in 'public' places and they could be large or small. They are used for a large audience or for a small group.
The display is only for information relevant to location.
The interactive one uses stylus and touch sensitive screen. But in all cases, the location matters and meaning of information or interaction is related to the location.

3.6 Physical Controls, Sensors etc.
These are special displays and gauges, sound, touch, feel and smell sensors. They also include physical controls, environmental and bio-sensing devices.

Dedicated displays like physical controls and sensors, comprise the following:
Analogue representations that include dials, gauges, lights, etc.
Digital displays as identified in small LCD screens and LED lights, etc.
Head-up displays found in aircraft cockpits. These show most important controls depending on context.

Sounds
Sounds are beeps, bongs, clonks, whistles and whirrs that are used for error indications and for confirmation of actions e.g. key click

Touch, feel and smell devices otherwise called haptic devices:
Here touch and feeling are important such as in games that involve vibration, and force feedback
Also important in simulation that requires the feel of surgical instruments.
However, current technology in texture, smell and taste is very limited

BMW iDrive
This is used for controlling menus with a feel of small 'bumps' for each item making it easier to select options by feel.
It uses haptic technology from Immersion Corporation. See below

Physical Controls:
These are specialist controls that exist in industries and those that are built into consumer products.
See examples below.

Environment and bio-sensing
These are sensors all around us suchas can be found in the following:
Car courtesy light — small switch ondoor
Ultrasound detectors — as in securityand washbasins.
RFID security tags in shops
Bio-sensors are you for personal health as applicable in the use of iris scanners, easurement of body temperature, heart rate, galvanic ski n response and blink rate.

3.7 Print Technology:
The following elements of interactio are provided by the print technology
Fonts, Page description, What You S e Is What You Get (WYSIWYG), Scanning, 0 ptical Character Reading (OCR) etc

Printing
This is an image made from small do s. It allows any character set or graphic to b e printed.
Critical features are resolution expre ssed in size, spacing of the dots in dots per i ch (dpi).
Speed is usually measured in pages per minute.
Types of dot-based printers
i. Dot-matrix printers: These use inke d ribbon (like a typewriter). They have line o f pins that can strike the ribbon hence dotting the paper.
Typical resolution is between 80 to 1 20 dots per inch (dpi)
ii. Ink-jet and bubble-jet printers These work by sending tiny blobs of i nk from print head to paper. Resolution is ty pically 300 dpi or more.
iii. Laser printers
These are like photocopiers: Here do ts of electrostatic charge are deposited on d rum, which picks up toner (black powder form of ink) rolled ont o paper which is then fixed with heat.
Typically 600 dpi or more.

Printing (aspect of human computer interaction) in the workplace
Shop tills:
Dot matrix printer uses same print h ad for several paper rolls; may also print ch e q u e s
Thermal printers use special heat-se sitive paper, paper heated by pins makes a d o t .
Though of poor quality, printing is si ple and maintenance is low.
They are used in some fax machines.

Fonts
Font is a particular style of text accor ding to some examples given below;
Courier font, Helvetica font, Palatino font, Times Roman font, CICICICICICICICICICICI (spe cial symbol)
Size of a font is measured in points (point is about 1/72"), about related to its h eight.

Pitch
Fixed-pitch : In this case, every character has the same width,e.g. Courier
Variable-pitched — some characters are wider e.g. Times Roman — compare the 'i' and the "m"
Sans-Serif or Serif :
Sans-serif contain square-ended strokes such as in in Helvetica

Serif These are characters with splayed ends such as in in Times Roman or Palatino

Readability of text
Lowercase: easy to read shape of words
UPPERCASE: better for individual letters and non-words e.g. flight numbers: BA79 3 vs. ba79 3
Serif fonts: helps your eye on long lines of printed text but sans serif is often better on screen

Page Description Languages
Useful when pages become very complex with different fonts, bitmaps, lines, digitised photos, etc.
Description languages can convert all into a bitmap and send to the printer.
Using a page description language sends a *description* of the page for example, instructions for curves, lines, text in different styles, etc. can be sent.
It is like a programming language for printing!
PostScript is the most common form of Page description language

3.8 Scanners
These accept paper and convert it into a bitmap
Two sorts of scanner exist:
Flat-bed: Here paper is placed on a glass plate with whole page converted into bitmap
Hand-held: Here scanner is passed over paper, digitising strip typically 3-4" wide
Shines light at paper and note intensity of reflection with colour or greyscale
Typical resolutions are from 600—2400 dpi
Scanners are used in desktop publishing for incorporating photographs and other images.
They are used in document storage and retrieval systems thereby doing away with paper storage.
Special scanners exist for slides and photographic negatives

Optical Character Recognition (OCR):
Optical character recognition (OCR) converts bitmap back into text with different fonts creating problems for simple "template matching" algorithms.
However, more complex systems segment text, decompose it into lines and arcs, and decipher characters that way.
Page formatting is done on columns, pictures, headers and footers.

Paper-based interaction
Paper is usually regarded as *output* only but can be *input* too in OCR, scanning, etc. operations.
Xerox Paper Works is a paper based interaction that involve glyphs containing small patterns of #\\##\\\
Used to identify forms etc. and also used with scanner and fax to control applications
More recently, papers are micro printed - like watermarks. Watermarks identify *which* sheet and *where* you are.
Special 'pen' can read locations to know where they are writing.

.9 Memory Interaction

Exists in form of short term and long term
One needs to have knowledge of the characteristics of a particular memory for an effective and valuable interaction. Such include the speed, capacity, the compression formats and mode of access.
Short-term Memory
Short-term Memory otherwise called Random access memory (RAM) are made of silicon chips. Most RAMs have 100 nano-second access time and are usually volatile, losing information if power is turned off.
Data transfer rate is around 100 Mbytes/sec
Some *non-volatile RAMs(ROMs) are* used to store basic set-up of information
Typical desktop computers have between 64 to 256 Mbytes of RAM.
Long-term Memory
These include magnetic disks comprising:
i. Floppy disks that can store up to 1.4 Mbytes and more; hard disks that can store between 40 Gbytes to hundreds of Gbytes
Access time approximate to 10 micro seconds while transfer rate is around 100kbytes per second.
ii. Optical disks that use lasers to read and sometimes write. They are more robust that magnetic media.
They include CD-ROMs that have same technology as home audio, with storage capacity approximating 600 Gbytes and DVDs used for AV applications and for very large files.
iii. PDAs (Personal Digital Assistants) that often use RAM for their main memory.
iv. Flash-Memories used in PDAs, cameras etc.. They are silicon based but persistent. Flash memories are Plug-in USB devices used for data transfer.
Virtual memory
Problems calling for the use of virtual memory include running lots of programs. Each program could be very large but with insufficient RAM size to run it, hence the need for the use of Virtual memory.
The solution provided by Virtual memory is to store some programs temporarily on disk thereby making RAM appear bigger. But the program on disk that needs to run again and copied from disk to RAM slows down processing.
Compression
This is the reduction in the amount of storage required. The two types that could be identified are:
Lossless compression:
i. Here, the exact text or image is recovered e.g. as in GIF, ZIP formats.
ii. By looking for commonalities in text as demonstrated below:
iii. If text= AAAAAAAAAABBBBBCCCCCCCC then there are 10 As,5 Bs, and 8 Cs and is written as 10A5B8C
iv. Video: Here successive frames are compared and changes are stored.
Lossy compression:
This recovers something like the original — e.g. as in JPEG and MP 3 formats.
Exploited perception:
In JPEG, perception is exploited by losing rapid changes and some colour while in MP 3, perception is exploited by
reducing accuracy of drowned out notes

3.10 Storage formats
Text formats:
ASCII : This is a 7-bit binary code that represents each letter and character
UTF-8 : This is an 8-bit encoding of 16 bit character set
RTF (rich text format): This is a text plus formatting and layout information
SGML (standardized generalised mark-up language): These are documents regarded as structured objects
XML (extended mark-up language): This is a simpler version of SGML for web applications
Media formats:

Many storage formats exist for images : Such formats include PostScript, GIFF, JPEG, TIFF, PICT, etc. in addition to different compression techniques that reduce their storage requirements.
For Audio/Video, there are lots of formats as well. Such include QuickTime, MPEG, WAV, etc.. Compression is even more important here to optimise available storage space.
For network delivery of data, 'streaming' formats are also available.
Methods of access:
For any of the data type above, a large information store takes long time to search. Therefore, an index storage technique is used. Whatever is indexed is what can be accessed. Simple index needs exact match. Accessing without structure involves free text indexing all the words in a document hence requires lots of space!!

3.11 Processing and Networks Interactions
There is finite speed applicable to processing which is also governed by Moore's law.
There are also limits of interaction both in single user and networked computing.

Finite processing speed
Designers tend to assume fast processors and therefore make interfaces more and more complicated. But problems do occur, because processing cannot keep up with all the tasks it needs to do for the following reasons:
i. Cursor overshooting because system has buffered key presses.
ii. Icon wars Here, the user clicks on an icon and nothing happens, clicks on another, then system responds and windows fly everywhere.
iii. Also problems do occur if system is too fast. For example, help screens may scroll through text much too rapidly to be read.

Moore's law
The Moore's law observes that computers get faster and faster!
In 1965, Gordon Moore, co-founder of Intel, noticed a pattern that processor speed doubles every 18 months e.g
PC in1987, speed was 1.5 Mhz, up till 2002 when the speed became 1.5 GHz.
Similar pattern also occurs for memory and storage. But this doubles every 12 months!! e.g.
Maximum Hard disk requirement in 1991 was 20 Mega bytes but rose in 2002 to 30 Giga bytes

The myth of the infinitely fast machine:
The implicit assumption here is that there are no delays in processing on an infinitely fast machine.
This takes us to the question: What is a good design for real machines?

Networked computing
Networks allow access to large memory and processing, access to other people (such as in groupware and email) and other shared resources especially the web
Issues relating to interaction in network computing are network delays which cause slow feedback, and unexpected processing delay as a result of many people updating data simultaneously.
Lastly is the unpredictability nature of networks.

The internet as an example of a network, (international network of computers)
Short History:
In 1969: DARPANET US Department of Defence had 4 sites. In 1971, the sites increased to 2 3. In 1984, it became 1,000 and in 1989, it increased to 10,000 .
Common language protocols used are:

i. TCP — Transmission Control protocol: This operates at lower level on packets (like letters) between machines.
ii. IP — Internet Protocol: This provides a reliable channel (like phone call) between programs on machines. Email and HTTP (Hypertext Transmission Protocol) build on top of these.

4.0 Conclusion

In spite of all the discussions about the various interactions above, some limitations are placed on interactive performance as follows:
Computation bound: Computation may take a long time, causing frustration for the user.
Storage channel bound: Bottlenecks may occur in the transference of data from disk to memory.
Graphics bound: Here common bottleneck is that updating displays requires a lot of effort, effort sometimes helped by adding a graphics co-processor optimised to take on the burden
Network capacity: Though many computers are networked with sharable resources and files and access to printers etc. yet still have interactive performance reduced by slow network speed.

5.0 Summary

The knowledge of the computer systems and the types of user interface devices such as the text entry, hand writing and speech recognition, pointing and drawing, display and storage together with the kind of network facilities have their individual impacts on human computer interactions.

6.0 Tutor Marked Assignment

7. What is the reason for the difference in the arrangement of Numeric keypads existing between your phone keyboard and the calculator keyboard
8. What are the specific difficulties experienced in handwriting recognition as a human interaction medium
9. Mention 2 advantages and 2 disadvantages of the touch sensitive screen interface
10. Explain the characteristics of the eye gaze interaction.
11. Mention 5 health hazards that may likely exist as a result of interacting with the CRT
12. (a) What do you know of haptic devices and biosensors?
    (b) Describe the function and value of Page Description language

7. What is Virtual memory and its benefit in Computer processing?

7.0 Further Readings / References

- Card, S.K., "Pioneers and Settlers: Methods Used in Successful User Interface Design," in *Human-Computer Interface Design: Success Stories, Emerging Methods, and Real-World Context*, M. Rudisill, *et al.*, Editors. 1996, Morgan Kaufmann Publishers: San Francisco. pp. 122-169.
- Baecker, R., *et al.*, "A Historical and Intellectual Perspective," in *Readings in Human-Computer Interaction: Toward the Year 2000, Second Edition*, R. Baecker, *et al.*, Editors. 1995, Morgan Kaufmann Publishers, Inc.: San Francisco. pp. 35-47.
- Brooks, F. "The Computer "Scientist" as Toolsmith--Studies in Interactive Computer Graphics," in *IFIP Conference Proceedings.* 1977. pp. 625-6 34.
- Buxton, W., *et al.* "Towards a Comprehensive User Interface Management System," in *Proceedings S IGGRAPH'83: Computer Graphics.* 198 3. Detroit, Mich. 17. pp. 35-42.
- Engelbart, D. and English, W., "A Research Center for Augmenting Human Intellect." *Reprinted in ACM S IGGRAPH Video Review, 1994.*, 1968. 106
- Goldberg, A., ed. *A History of Personal Workstations*. 1988, Addison-Wesley Publishing Company: New York, NY. 5 37.

UNIT 3: BASIC COMPONENTS OF HUMAN COMPUTER INTERACTION

Table of contents

1.0 Introduction

The components of Human Computer Interaction comprise the Interaction models that concern translation between the user and the computer system, Ergonomics that describe the physical characteristics of interaction, the Interaction styles that express the nature of user and system dialog and finally the context of the social, organizational and the motivational aspect of interaction.

This unit briefly describes each of these components with a desire of giving an overview of the general requirements for the design of human computer interaction systems.

2.0 Objectives

By the end of this unit, you should b able to:

- Understand the Interaction m odels of interface between the user and sys tem
- Describe the cycle of the exe ution and evaluation loop
- Explain the concepts of ergon omics
- Know the common interactio n styles
- Understand the context of th e social, organizational and the motivational aspects of interaction
- The usefulness of the WIMP I nterface

.0 Main Content

.1 The Interaction Models

The interaction models comprise
The terms of Interaction,
The Donald Norman Model
The Interaction framework

3.1.1 Terms of interaction
Domain: This is the area of work under study e.g. a graphic design
Goal : This is what you want to achieve e.g. to create a solid red triangle
Task : Concerns how you go about d ing it, ultimately in terms of operations or a ctions e.g select the fill tool, click over the triangle

3.1.2 Donald Norman's model:
These are in seven stages as follow:

- The user establishes the goal
- The user formulates intention
- The user specifies actions at i nterface
- The user executes the action
- The user perceives the syste state
- The user interprets the syste state
- The user evaluates the syste state with respect to goal

Norman's model concentrates on th user's view of the interface

Execution and evaluation loop

Interpretation:
Goal:
The user establishes the goal

Execution:
The user formulates intention
The user specifies actions at interface
The user executes the action

Evaluation:
The user perceives the system state
The user interprets the system state
The user evaluates the system state with respect to goal

Donald Norman's model
Norman's model can be applied through:
Gulf of Execution that evaluates the user's formulation of actions where actions are allowed by the system
Gulf of Evaluation where the user's expectation of changed system state represent actual presentation of this state

Interaction could harbour some human errors which may be slips and mistakes.
Slips may include lack of understanding the system and goal, incorrect formulation of action, incorrect action and mistake of not even having the right goal!
To fix slips, better interface design should be carried out while to avoid mistakes, one should better understand the system

To avoid some of the Human errors, Abowd and Beale framework is adopted. Abowd and Beale framework is an extension of Norman model and it has 4 parts namely: i. the user, ii. the input, iii. the system, and iv. the output while each framework has its own unique language.
If interaction is the translation between languages, and if there are problems in interaction, then there would be problems in translation

Using Abowd & Beale's model
The user intentions could be translated into actions at the interface, translated into alterations of system state, reflected in the output display or interpreted by the user himself.
The general framework for understanding interaction are that interaction is not restricted to electronic computer systems alone, all major components involved in interaction should be identified. The comparative assessment of systems should be allowed. The framework also considers an abstraction.

3.2 Ergonomics
This considers both the physical aspects of interfaces and the industrial interfaces.
Ergonomics is the study of the physical characteristics of interaction. It is known as human factors.
Ergonomics is good at defining standards and guidelines for constraining the way we design certain aspects of systems
Examples of Ergonomics include:
Arrangement of controls and displays such as the controls grouped according to function, frequency and sequence of use.
Surrounding environment such as the seating arrangements adaptable to cope with all sizes of user, health issues such as the physical position, environmental conditions (temperature, humidity), lighting, and noise.

Use of colour such as the use of red for warning, green for okay, and awareness of colour-blindness etc. The user interacts *with* real world *through* interface issues, feedback and delays.

. 3 Common Interaction styles

Two major classes of interaction styles will be considered, they are:-
Dialogue Style of Interaction between computer and user
Distinct styles of interaction
Both are expressed in the following common forms of interfaces:

- Command line interface
- Menus
- Natural language
- Question and answer, and query dialogue
- Form-fills and spreadsheets
- WIMP
- Point and click
- Three—dimensional interfaces

3. 3.1 Command line interface

This is the way of expressing instructions to the computer directly through the function keys, single characters, short abbreviations, whole words, or a combination suitable for repetitive tasks.
The interface is better designed for expert users than novices because it offers direct access to system functionality. However, the command names and abbreviations used should be meaningful!
A typical example is the Unix system command line interface.

3. 3.2 Menus

Menus is a set of options displayed on the screen. The Menu Options are visible, it has a less recall characteristic that make it easier to use.
The visible options rely on recognition so the names should be meaningful. The selection is done through numbers, letters, arrow keys, mouse and/or combination of any of them e.g. mouse plus accelerators .Often, the options are hierarchically grouped. But sensible grouping is needed.
Menus are restricted form of full WIMP system .

3. 3. 3 Natural language

This is the language familiar to the user. It may be in form of speech recognition or a typed natural language.
Problems with in this kind of interaction are that the language may be vague, ambiguous, and hard to be recognised.
Design solutions to language interface problems are for the user to try to understand a subset and pick on key words .

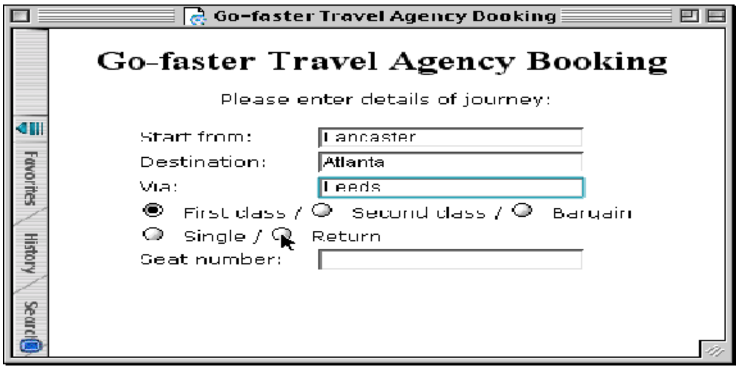
3. 3.4 Query interfaces

These comprise question and answer interfaces in which the user is led through interaction via series of questions. Though with restricted functionality, this kind of interface is suitable for novice users. It is often used in information systems.
Query languages (e.g. SQL)
This is used to retrieve information from database. It requires understanding of the database structure and language syntax, hence requires some expertise

3. 3.5 Form-fills and Spreadsheets

Form-fills are primarily designed for data entry or data retrieval. It is a screen like paper form to which data is put in relevant place.

It requires a good design and obvious correction facilities.
See illustration below



Example of a form-fill

Spreadsheets
Spreadsheets are sophisticated variations of form-filling in which grid of cells contain a value or a formula. The formula can involve values of other cells e.g. sum of all cells in this column.
The user can enter and alter data in spreadsheet to maintain consistency.
The first spreadsheet introduced was VISICALC, followed by Lotus 1-2- 3. Micro Soft Excel is the most common today

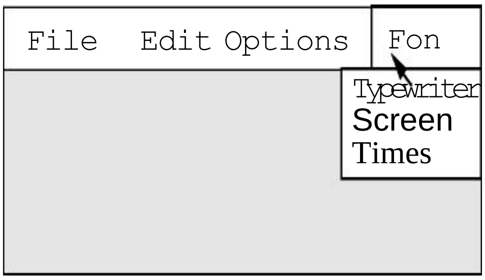
3. 3.6 WIMP Interface
This interface comprises Windows, Icons, Menus, and Pointers or Windows, Icons, Mice, and Pull-down menus!
The interface is the default style for majority of interactive computer systems, especially PCs and desktop machines.
Elements of the WIMP interface
The elements include windows, icons, menus, and pointers. In some other cases they may be buttons, toolbars, palettes, and dialog boxes.
Understanding the concept of 'Look and feel'
WIMP systems have the same elements: as windows, icons., menus, pointers, buttons, etc. but have different window systems that *behave* differently . For example, Macintosh Operating System (MacOS) compared with Windows menus.
The combination of the appearance and the behaviour is the 'look and feel'

Windows
Windows are areas of the screen that behave as if they were independent. They can contain text or graphics and can be moved or resized.
They can overlap and obscure each other, or can be laid out next to one another (tiled)

Icons
Icons are small pictures or images that represent some object in the interface. They appear often as windows or as actions.
Windows can be 'iconised' that is closed down. They are small representations that fit many accessible windows.
Icons can be many and various. They can be highly stylized with realistic representations.

Menus
These are choice of operations or services offered on the screen
The required option is selected with the pointer. However, this takes a lot of screen space

This problem is partly solved when a pop-up menu appears when needed



Kinds of Menus

Menu Bar at top of screen (normally), menu drags down

i Pull-down menu - mouse hold and drag down me

ii Drop-down menu - mouse click reveals men

v. Fall-down menus - mouse just moves over bar!

Contextual menu appears where you are

Pop-up menus take actions for selected object

Pie menus are arranged in a circle such that it is easier to select item over larger target area. Selection is also quicker because it can move same distance to any option. Pie menus are not widely used!

Cascading menus

This has a hierarchical menu structure in which a menu selection opens new menu and so in ad infinitum

Keyboard accelerators

This comprises key combinations with same effect as menu item.

They operate in two modes

– active when menu open — usually first letter and

– active when menu closed — usually Ctrl + letter

Menus design issues

In order to design an effective menu, the following issues should be considered:

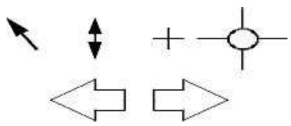
- which kind to use
- what to include in menus at all
- words to use (in action or description)
- how to group items
- choice of keyboard accelerators

Palettes and tear-off menus

- Palettes are little windows of actions shown or hidden via menu option in available shapes in drawing package
- In tear-off and pin-up menus, menu 'tears off' to become palette

Pointers

Pointers are important WIMP style components that point on and select. They are activated by the use of mouse, track pad, joystick, trackball, cursor keys or keyboard shortcuts. They are in wide variety of graphical images. See examples below.



Point and click interfaces

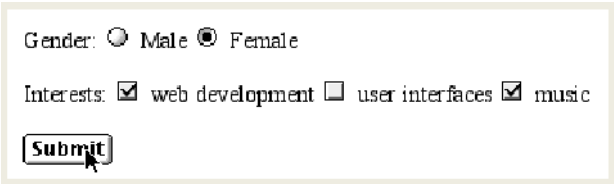
Point and click interfaces are used in multimedia, web browsers, and hypertext. You just click something such as icons, text links or location on map. It requires minimal typing.

Scrollbars

Scrollbars allow the user to move the contents of the window up and down or from side to side.

Title bars
Title bars describe the name of the window

Buttons
This is an individual and isolated region within a display that can be selected to invoke an action
The Special kinds that exist are
The radio buttons with a set of mutually exclusive choices and the check boxes with a set of non-exclusive choices.



Toolbars
These are long lines of icons with fast access to common actions and are often customizable:
You can choose *which* toolbars to see and choose *what* options are on it

Dialogue boxes
These are information windows that pop up to inform of an important event or requested information, for example when saving a file, a dialogue box is displayed to allow the user to specify the filename and location. Once the file is saved, the box disappears.

The interactivity of dialogue boxes
They are easy to focus on look and feel.

Other types of interaction styles are speech—driven interfaces
The development of this kind of interface is yet to be perfect and accurate; though it is rapidly improving. Example of speech driven interface dialogue on an airline reservation:

reliable "yes" and "no"?
+ System reflects back its understanding
"you want a ticket from New York to Boston?"

3. 3.7 Three dimensional interfaces
These are virtual reality 'ordinary' window systems highlighting visual affordance. The indiscriminate use can however be confusing!
There are also three dimensional (3D) workspaces used for extra virtual space with light and occlusion that give deep distance effects.
For typical computer displays, *t*hree-dimensional images are projected on them in two dimensions. Three-dimensional graphics are currently mostly used in computer games, art and computer-aided design (CAD). There have been several attempts at making three-dimensional desktop environments like Sun's Project Looking Glass. A three-dimensional computing environment could be used for collaborative work. For example, scientists could study three-dimensional models of molecules in a virtual reality environment, or engineers could work on assembling a three-dimensional model of an airplane.] The Technologies The use of three-dimensional graphics has become increasingly common in mainstream operating systems, but mainly been confined to creating attractive interfaces—eye candy—rather than for functional purposes only possible using three dimensions. For example, user switching is represented by rotating a cube whose faces are each user's workspace, and window management is represented in the form of

Exposé on Mac OS X. In both cases, the operating system transforms windows on-the-fly while continuing to update the content of those windows.
Interfaces for the X Window System have also implemented advanced three-dimensional user interfaces through compositing window managers such as Beryl and Compiz using the AIGLX or XGL architectures, allowing for the usage of OpenGL to animate the user's interactions with the desktop.
Another branch in the three-dimensional desktop environment is the three-dimensional GUIs that take the desktop metaphor a step further, like the BumpTop, where a user can manipulate documents and windows as if they were "real world" documents, with realistic movement and physics.
The Zooming User Interface (ZUI) is a related technology that promises to deliver the representation benefits of 3D environments without their usability drawbacks of orientation problems and hidden objects. It is a logical advancement on the GUI, blending some three-dimensional movement with two-dimensional or "2.5D" vector objects.

3.4 Context: Social and Organisational

These issues and concerns involve all possible interactions between a user and a system during its lifecycle, including the development stage, use in context, and the impact of such use on individuals, organizations, society, and future systems development.

Context Analysis

Context analysis includes understanding the technical, environmental and social settings where the information systems will be used. It examines whether and how the interaction between physical and social environment and the physiological and psychological characteristics of the user would impact users interacting with the system.
There are four aspects in Context Analysis: physical context, technical context, organizational context, and social and cultural context. Overall, context analysis can provide ideas for design factors such as metaphor creation, selection and patterns of communications between users and the system.

Physical context: This considers where the tasks carried out, what entities and resources are implicated in task operation, What physical structures and entities are necessary to understand observed task action. For example, an ATM machine can be used in a mall outside a bank office, or in a night club. These environments provide different levels of lighting, crowdedness, and noisiness. Thus legibility of the screen, use of audible devices
for input or output, or even the size of the working space to prevent people nearly to see the screen could be designed differently.

Technical context: This considers the technology infrastructure, platforms, hardware and system software, wired or wireless network connection? For example, an E-commerce website may be designed to allow access only to people with certain browser versions. The website may also be designed to allow small screen devices such as PDA or mobile phone to access.

Organizational context: Organizational context may play different roles in internal and external situations. For an organizational information system to be used by the organization's own employees, organizational context analysis answers questions such as:

- What is the larger system where this information system is embedded?,
- What are the interactions with other entities in the organization?
- What are the organizational policies or practice that may affect individual's attitude and behavior towards using the system?

For example, assuming that Lotus Note is used by an organization as a communication and collaboration tool, management may depend on using the tool to set up meetings by checking employees' calendars on mutually available time slots. The effectiveness of setting up meetings depends on whether employees use the tool, and how they use it. The whether and how questions can be enforced by organizational policies.

Social and cultural context: What are the social or cultural factors that may affect user attitudes and eventual use of the information system? In an E-Commerce website example, the website can be accessed from all over the world. It thus is a design consideration that the website allows access by people with any language and cultural
background that can provide credit cards with the foreign currency exchange, or it is only accessible to people who speak certain languages and are from certain cultures.

Interactions are also affected by other social and organizational context as follow:

- By other people: A desire to impress, competition among stakeholders, and fear of failure from management
- Motivation from management as against fear, allegiance, ambition, self-satisfaction that exist among employees
- Existing inadequate systems that may cause frustration and lack of motivation

The organizational, social and cultural context in which humans interact with IT is largely the result of the broad adoption of IT by organizations and society to support organizational functions and goals and to enhance society's development. For example, organizational efficiency may be expected due to redesign of workflows among critical business units that is affected by the implemented IT; satisfaction and retention of customers/clients are anticipated
due to accurate and fast information gathering and presentations, to name a few. Some of the organizational or societal impacts may not be tangible or directly attributed to HCI considerations. This assertion is in line with the issues of determining IT values in organizations and societies.
While each of these HCI concerns may have its own importance in different situations in relation to human motivation, it would be helpful for designers to see an overview picture of the potential HCI concerns and goals. The purpose of this picture is not to force every IT to be compliant with all the HCI concerns, but to provide an overall framework so that designers can use it as a roadmap and to apply it according to different situations.

4.0 Conclusion

The knowledge of the basic components of human computer interaction aids in giving direction, focus and human considerations pertaining to interactive design.

5.0 Summary

There are variations of interaction models of the interface established between the user and the computer system. These models are presented as a loop in the execution and evaluation of an interactive design. These interaction models together with the human ergonomics, the interaction styles, and the social and organizational contexts are basic components of human computer interaction.

6.0 Tutor Marked Assignment

9. Explain the three terms of Interaction
10. Mention any 4 of the 7 stages of the Donald Normans model and briefly describe each of the stages.
11. What do you understand as the execution and evaluation loop and how is the loop useful in the user's participation in the design of interactive systems
12. What are slips and mistakes in human Computer interaction and how do you avoid such slips and mistakes before they occur?
13. Briefly express your understanding of the term Ergonomics

14. Describe any four common interaction styles
15. What are Cascading menus and Keyboard accelerators?
16. What are the four constituent aspects of Context analysis? How are they of benefits to the user interacting with the computer system?

7.0 Further Readings / References

- Licklider, J.C.R. and Taylor, R.W., "The computer as Communication Device." *Sci. Tech.*, 1968. April: pp. 21- 31.
- Linton, M.A., Vlissides, J.M., and Calder, P.R., "Composing user interfaces with InterViews." *IEEE Computer*, 1989. 22(2): pp. 8-22.
- Meyrowitz, N. and Van Dam, A., "Interactive Editing Systems: Part 1 and 2." *ACM Computing Surveys*, 1982. 14(3): pp. 321- 352.
- Myers, B.A., "The User Interface for Sapphire." *IEEE Computer Graphics and Applications*, 1984. 4(12): pp. 1 3-2 3.
- Myers, B.A., "A Taxonomy of User Interfaces for Window Managers." *IEEE Computer Graphics and Applications*, 1988. 8(5): pp. 65-84.

UNIT 4: CRITICAL EVALUATION OF COMPUTER BASED TECHNOLOGY

Table of contents

1.0 Introduction

This unit briefly describes some uncommon technologies associated with human computer interaction. These are innovations that improve upon the user interface, particularly those innovations benefiting the disabled. Technologies such as the phonetic typewriter, the ear cons, the auditory icons, the recognition and gesture devices for the disabled and the elderly are described.

2.0 Objectives

By the end of this unit, you should be able to:

- Describe multi-modal, multi-media and multi-sensory systems
- Appreciate the speech and the Phonetic typewriter interfaces
- Understand the Ear cons and Auditory Icons as important components of multi-modal systems
- Know that Recognition and Gestures Devices are essential for the Elderly and Disabled

.0 Main Content

3.1 Multi-Sensory Systems
Here, more than one sensory channel are involved in interaction as in sounds, text, hypertext, animation, video, gestures and vision.
They are used in a range of applications particularly for users with special needs and virtual reality.
The components of Multi-Sensory systems are: Speech, Non-speech sounds, Handwriting, together with their applications and principles.

Usable Senses
The five senses: sight, sound, touch, taste and smell are used by us every day and each is important on its own Together, they provide a fuller interaction with the natural world
We can ideally utilise the Computers to use all the available senses but this becomes practically impossible because computers rarely offer such a rich interaction.
We can use the sight, sound, and sometimes the touch senses, but we cannot yet use the taste and smell.

3.2 Multi-modal and Multi-media systems:
Multi-modal systems
These use more than one sense (or mode) of interaction as in visual and aural senses. For example, a text processor may speak the words as well as echo them to the screen
Multi-media systems
These use a number of different media to communicate information. For example, a computer-based teaching system may use video, animation, text and still images; different media all using the visual mode of interaction. These may also use sounds, both speech and non-speech. Of course two or more media now use different modes.

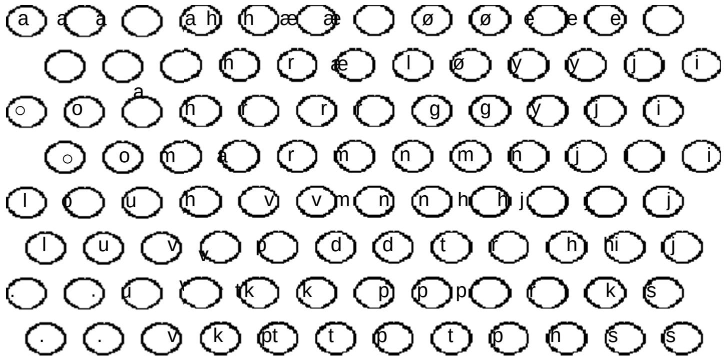
3.2.1 Speech
Human beings have a great and natural mastery of speech which makes it difficult to appreciate the complexities but it is an easy medium for communication
Simple terminologies used to describe speech:
The structure of speech is called phonemes and there are 40 of them. The phonemes as basic atomic units that sound slightly different depending on the context they are in, the larger units of phonemes are the Allophones. Allophones are the sounds in the language between 120 and 1 30 and are formed into morphemes. The morphemes are the smallest units of language that have meaning.
Prosody is the alteration in tone and quality. They are also variations in emphasis, stress, pauses and pitch. They impart more meaning to sentences.
Co-articulation is the effect of context on the sound. It transforms the phonemes into allophones.
Syntax is the term used for the structure of sentences while semantics is the collective term used for the meaning of sentences.

3.2.2 Problems in Speech Recognition.
Different people speak differently because accent, intonation, stress, idiom, volume, etc. differ. The syntax of semantically similar sentences may also vary while background noises can interfere.
People often "ummm....." and "errr….. " but words are not enough - semantics are also needed. It requires intelligence to understand a sentence because context of the utterance often has to be known as well as information about the subject and speaker. For example, even if "Errr……I, um, don't like this" is recognised, it is a fairly useless piece of information on its own.

3.2. 3 Speech Related Human-Interaction Technologies.
The Phonetic Typewriter

This is developed for Finnish - a phonetic language. This machine trained on one speaker, will generalise the training to others.
A neural network is trained to cluster together similar sounds, which are then labelled with the corresponding character.
When recognising speech, the sounds uttered are allocated to the closest corresponding output, and the character for that output is printed.
It requires large dictionary of minor variations to correct general mechanism.

The Phonetic Typewriter



Usefulness of Speech Recognition:
It is useful for a single user or in a situation in which limited vocabulary systems exist, for example in a computer dictation.
In the public and open places of limited vocabulary systems, it can work satisfactorily e.g. in some voice activated telephone systems.
For the general user with wide vocabulary systems, problems do occur.
Its great potential value however manifests when users hands are already occupied as in driving or during manufacturing particularly for users with physical disabilities.
Another advantage is its lightweight and its use as a mobile device.

Speech Synthesis
This is a generation of speech. It is useful because of its natural and familiar way of receiving information. It is successful in certain constrained applications when the user has few alternatives and is particularly motivated to overcome problems.
However, it has its own problems similar to speech recognition particularly in prosody. Additional problems can arise from intrusion calling the need for headphones particularly due to noise in the workplace .Its transient nature is a problem when it becomes harder to review and browse.
Few examples occur in screen readers that read the textual display to the user e.g. as utilised by visually impaired people. Also in warning signals of spoken information sometimes presented to pilots whose visual and haptic skills are already fully occupied while flying.

. 3 Sounds

Non-Speech Sounds:
These are bongs, bangs, squeaks, clicks etc. that are commonly used for warnings and alarms. Fewer typing mistakes occur here with key clicks. It is also useful in video games that become uninteresting without sound.
Unlike speech, it is language and culture independent.

Non-Speech Sounds provide dual mo de displays in information presented along two different sensory channels. It provides redundant pres entation of information in the resolution of mbiguity in one mode through information in another.

It is good for providing both transien t and background status information e.g. So nd can be used as a redundant mode in the Apple Macin osh. Also, almost any user action (file select ion, window active, disk insert, search error, copy complete, tc.) can have a different sound associated with it.

Auditory Icons

These use natural sounds to represe t different types of object or action. Natura l sounds have associated semantics which can be mapped ont o similar meanings in the interaction e.g. thr owing something away such as the sound of smashing glass.

Problem sometimes arise because n t all things have associated meanings.

Additional information can also be p esented on muffled sounds if object is obsc red or action is in the background. The use of stereo allow positional information to be added.

Examples:

SonicFinder for the Macintosh: Here, items and actions on the deskt op have associated sounds. For example, fol ders have a papery noise and moving files produce a dragging sound.

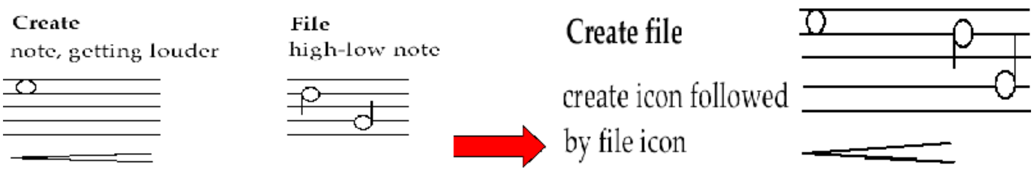
Copying sound gives a sound of a lig id being poured into a receptacle while risin g pitch indicates the progress of the copy.

Big files have louder sound than sma ller ones.

Earcons

These are synthetic sounds used to c onvey information. They comprise structure d combinations of notes or motives that represent actions an d objects. Motives are combined to provide rich information.

Compound earcons are multiple m earcon



Family ear cons

Here, similar types of earcons repres ent similar classes of action or similar object s .The family of "errors" would contain syntax and operating system errors. Earcons are easily grouped an d refined due to compositional and hierarchical natur e.

It is harder to associate with the inte rface task since there is no natural mapping.

3.4 Recognition and Gestures

Touch recognition:

Comprises:

i. Haptic interaction made up of cuta neous perception that provide tactile sensa ion and vibrations on the skin

ii. Kinaesthetic comprising movemen t and position and force feedback.

Touch recognition also include infor ation on shape, texture, resistance, temper ature, and comparative spatial factors.

Examples of technologies on touch recognition include electronic Braille displays and force feedback devices e.g. Phantom that recognises resistance and texture.

Handwriting recognition
Handwriting is another communication mechanism which we are used to in day-to-day life
The technology of handwriting consists of complex strokes and spaces.
The handwriting is captured by digitising tablet through strokes transformed to sequence of dots.
Large tablets available are suitable for digitising maps and technical drawings.
Smaller devices, some incorporating thin screens are used to display the information. Such include PDAs such as Palm Pilot and tablet PCs.
The problems associated with handwriting recognition are personal differences in letter formation and co-articulation effects.
The breakthroughs in this technology is the creation of stroke not just bitmap found in special 'alphabet' like Graffeti on PalmOS
The technology is usable even without training though many people prefer to use the keyboards!

Gesture technology;
This can be found in its various applications such as in gestural input - e.g. "put that there" and sign language.
The technology comprises data glove and position sensing devices such as MIT Media Room.
Gesture provides the benefits of natural form of interaction by pointing.
It enhances communication between signing and non-signing users
The problems with gesture interaction are that it is user dependent due to the variable nature of each user. Issues of co articulation are also considered as problems.

3.5 Devices for the Elderly and Disabled
The development of Technology on Human Computer Interaction has helped users with disabilities as follow:
Visual impairment: Use of screen readers and Sonic Finder
Hearing impairment: Use of text communication, gesture and captions
Physical impairment: Use of speech input and output, eye gaze, gesture, predictive systems (e.g. reactive keyboard)
Speech impairment: Use of speech synthesis and text communication
Dyslexia: Use of speech input and output
Autism: Use of communication and education devices
Older people use disability aids, memory aids, and communication tools to prevent social isolation

Others:
Children use appropriate input and output devices for education, games and fun.
In solving cultural differences, the influence of nationality, generation, gender, race, sexuality, class, religion, political persuasion etc. are affected by the interpretation of interface features. e.g. interpretation and acceptability of language, cultural symbols, gesture and colour.

4.0 Conclusion
Since the basic goal of HCI study is to improve the interactions between users and computers by making computers more usable and receptive to the user's needs, there is continuous research in human-computer interaction that involves exploring easier-to-learn or more efficient interaction techniques for common computing tasks. This includes inventing new techniques and comparing existing techniques using scientific methods.

5.0 Summary

Uncommon technologies associated with human computer interaction include consideration of the multi-modal and multi-media systems that incorporate speech recognition and synthesis, the phonetic typewriter, sound interface facilities, recognition and gestures mechanisms. These facilities particularly aid the elderly and the disabled to effectively and comfortably interact with the computer system.

6.0 Tutor Marked Assignment

6. What are multisensory systems, their components, and their relevance in the design of interactive systems?
7. What is speech Synthesis and how is it valuable to the computer user? Give two examples of its application.
8. Distinguish between an auditory icon and an earcon. Explain the limitation of their applications.
9. Explain the three types of Recognition and Gestures mode of interaction. Mention areas where each is effectively applied.
10. Mention any 3 devices that aid the elderly and the disabled in human computer interaction.

7.0 Further Readings / References

- Coons, S. "An Outline of the Requirements for a Computer-Aided Design System," in *AFIPS Spring Joint Computer Conference.* 196 3. 2 3. pp. 299- 304.
- Engelbart, D. and English, W., "A Research Center for Augmenting Human Intellect." *Reprinted in ACM S IGGRAPH Video Review, 1994.*, 1968. 106
- English, W.K., Engelbart, D.C., and Berman, M.L., "Display Selection Techniques for Text Manipulation." *IEEE Transactions on Human Factors in Electronics*, 1967. HFE-8(1)

MODULE 2: USER PERSPECTIVES OF HUMAN-COMPUTER INTERACTION

UNIT 1: USER ORIENTED PERSPECTIVE OF HUMAN COMPUTER INTERACTION : SOCIAL HUMAN THRUST

Table of contents

1.0 Introduction

This unit concerns socio-organizational issues and stakeholder requirements

Organizational issues affect acceptance of new computer systems because where there are conflicts and power, the question arises on who benefits and who encourages the use.

Stakeholders of a new computer system identify their requirements in organizational context.

Organizational context may play different roles in internal and external situations. For an organizational information system to be used by the organization's own employees, organizational context analysis answers questions such as: What is the larger system where this information system is embedded? What are the interactions with other entities in the organization? What are the organizational policies or practice that may affect individual's attitude and behavior towards using the system?

This event is the sociological, organizational, and cultural impact of computing. In other words, the organizational, social and cultural context in which humans interact with IT. This context is largely the result of the broad adoption of IT by organizations and society to support organizational functions and goals and to enhance society's development.

2.0 Objectives

By the end of this unit, you should be able to understand:

v. The socio-technical models that look at human and technical requirements
vi. The soft systems methodology that considers the broader view of human and organizational issues
vii. The participatory design that includes the user directly in the design process
viii. The ethnographic methods that study users in context with unbiased perspective

.0 MAIN CONTENT

3.1 Organisational issues

Organisational factors

Organisational factors can make or break a system. Studying the work group is not sufficient since any system is used within a wider context and the crucial people need not be the direct users.

Therefore, before installing a new system, one must understand:

- Who benefits?
- Who puts in effort?
- The balance of power in the organisation and how the system will be affected.

Even when a system is successful, it may be difficult to measure that success

Conflict and power

In computer supported *cooperative* work (CSCW), people and groups have conflicting goals hence the systems design assumes cooperation will fail! For example, in a computerised stock control, the store man looses control of information and may decide to subvert the system before or after it becomes operational. Therefore, it is important to identify stakeholders — not just the users.

Organisational structures

Groupware affects organisational structures so also communication structures reflect line management.
For example, email is a cross-organisational communication.
Organisation structure can also disenfranchise lower management and disaffected staff could 'sabotage'.
Because technology *can* be used to change management style and power structures, the implementation of such technology would improve upon organizational efficiency.
For example, organizational efficiency may be expected due to redesign of workflows among critical business units that is affected by the implemented IT; satisfaction and retention of customers/clients are anticipated due to accurate and fast information gathering and presentations, to name a few. It is noteworthy that some of the organizational or societal impacts may not be tangible or directly attributed to HCI considerations. This assertion is in line with the issues of determining IT values in organizations and societies.

3.2 Invisible workers

Telecommunications improvements allow neighbourhood work centres and home-based tele-working.
Many ecological and economic benefits arise from tele-working such as reduced travel and flexible family commitments.
But 'management by presence' is absent. The presence in the office increases perceived worth and reduces problems for promotion.
Barriers to tele-working are both managerial and social but *not* technological.
The new system should benefit all. Disproportionate effort should be avoided. Examine who puts in the effort and who gets the benefit.
It is possible to get benefit without doing work, if everyone does it, system falls into disuse
To get started, look for cliques to form core user base and design to benefit an initial small user base

Evaluating the benefits

Assuming we have avoided the pitfalls!
How do we measure our success?
Job satisfaction and information flow and economic benefit should diffuse throughout the organisation
There is the need to identify requirements within context of use and the need to take account of stakeholders. Work groups and practices should be identified in organisational context
Many approaches including socio-technical modelling, soft system modelling, participatory design and contextual inquiry are used.

Who are the stakeholders?

The system will have many stakeholders with potentially conflicting interests. Stakeholder is anyone affected by success or failure of system. The primary stakeholders actually use the system, the secondary

receive output or provide input, while the tertiary have no direct involvement but are those affected by the success or failure of the new system.
Facilitators are those involved in the development or deployment of the system.

Example: Classifying stakeholders — an airline booking system
An international airline is considering introducing a new booking system for use by associated travel agents to sell flights directly to the public.
Primary stakeholders: The travel agency staff and the airline booking staff
Secondary stakeholders: Customers and the airline management
Tertiary stakeholders: Competitors, civil aviation authorities, customers' travelling companions and, airline shareholders
Facilitating stakeholders: The design team and the Information Technology department staff

Designers need to meet as many stakeholder needs as possible. Usually needs may be in conflict so they have to prioritise, often priority decreases as one moves down the categories

3. 3 Socio-technical modelling

This is a response to *technological determinism* and it is concerned with the technical, social, organizational and human aspects of design. It also describes the impact of specific technology on organization. It is concerned with information gathering such as interviews, observation, discussion with focus groups and document analysis.
Several approaches to be considered are the Stakeholders' Focus and the ESTA (Eight Stage Task Analysis).

3. 3.1 Stakeholders' Focus

This comprises six stage processes that focus on stakeholders
The first describes the organizational context, including primary goals, physical characteristics, political and economic background
The second identifies and describes the stakeholders including personal issues, role in the organization and their job.
The third identifies and describes the work-groups whether formally constituted or not
The fourth identifies and describe task—object pairs; these are tasks to be performed and objects used
The fifth identifies the stakeholder needs: Stages 2—4 described above are in terms of both current and proposed system, the stakeholder needs are identified from the differences between the two.
Lastly, we consolidate and check the stakeholder requirements against earlier criteria.

3. 3.2 ESTA (Eight Stage Task Analysis)

This is an eight stage model that focuses on task
The primary task is identified in terms of users' goals
Secondly, task inputs to the system are identified
Thirdly, the external environment into which the system will be introduced is described, including physical, economic and political aspects
Fourthly, the transformation processes within the system are described in terms of actions performed on or with objects
In the fifth stage, the social system is analyzed by considering existing internal and external work-groups and relationships
At the sixth stage, the technical system is described in terms of configuration and integration with other systems
At the seventh stage, the performance satisfaction criteria are established, indicating social and technical requirements of the system.

The last stage specifies the new technical system.

3.5 Soft systems methodology

The soft systems methodology considers the broader view of human and organizational issues
There is no assumption of technological solution here — emphasis is on understanding the situation fully
This methodology was developed by Checkland.
The seven stages involved here are:

i. Recognition of problem and initiation of analysis
ii. Detailed description of problem situation; it is a rich picture stage
iii. Generation of the root definitions of system: this is known as CATWOE (see definition below)
iv. Conceptual model - this identifies transformations
v. This compares real world to conceptual model
vi. Identifies necessary changes
vii. Determines actions to effect changes

CATWOE (Clients, Actors, Transformation, World View, Owner, Environment) further defines and explains the soft systems methodology.
Clients: those who receive output or benefit from the system
Actors: those who perform activities within the system
Transformations: the changes that are affected by the system
World View - how the system is perceived in a particular root definition
Owner: those to whom the system belongs, to whom it is answerable and who can authorize changes to it
Environment: the world in which the system operates and by which it is influenced

3.5 Participatory design

In participatory design, workers enter into design context while in ethnography (as used for design), the designer enters into work context. Both make workers feel valued in design and encourages workers to 'own' the products. The user is an active member of the design team.

Characteristics

Participatory design is context and work oriented rather than system oriented.
It is collaborative and iterative
Methods involved are: brain-storming, storyboarding, workshops, pencil and paper exercises.

Ethics

The ethics involved the participatory socio-technical approach devised by Mumford.
It states that the system development is about managing change and that non-participants are more likely to be dissatisfied.
There are three levels of participation: consultative, representative, and consensus.
Design groups including stakeholder representatives make design decisions and job satisfaction is the key to solution
(See the unit on Participatory Design for more details)

3.6 Ethnography

This is very influential in CSCW
It is a form of anthropological study with special focus on social relationships and does *not* enter actively into situation.
It seeks to understand social culture, it is unbiased and open ended.

.7 Contextual inquiry

Here inquiry is conducted in ethnographic tradition but acknowledges and challenges investigator focus.
It creates a model of investigator being apprenticed to the user in order to learn about the work.
The investigation takes place in the workplace with detailed interviews, observation, and analysis of communications, physical workplace, and artefacts.
Number of models created is according to sequence, physical, flow, cultural, and artefact
The models are consolidated across users, while the output indicates task sequences, the artefacts and communication channels needed, the physical and cultural constraints

4.0 Conclusion

Work groups and practices should be identified in organisational context
Many approaches including socio-technical modelling, soft system modelling, participatory design and contextual inquiry have been explained.
In concluding, there is the need to identify requirements within context of use and the need to take account of stakeholders.

5.0 Summary

The socio human thrust analysis explains the socio-technical models of human and technical requirements, the systems methodology that considers human and organizational issues, the participatory design that includes the user directly in the design process and the ethnographic methods that study users in context with unbiased perspective.

6.0 Tutor Marked Assignment

1. "Organizational factors can make or break a system". Explain the concepts of this expression as it affects users in an organization when designing a new interactive computer system.
8. What do you understand by a Computer Supported cooperative Work (CSCW) in an organization. Why is it important to identify and consider stakeholders when designing interactive systems in an organization?
9. What are the factors used in evaluating whether a designed and implemented system is successful in an organization?
10. Who is a stakeholder in a human computer interaction? Differentiate between a primary stakeholder, a secondary stakeholder and a facilitator of a newly designed interactive system.
11. Differentiate between the different goals of the six stage model of Stakeholders' focus and the eight stage model of ESTA within the human aspects of interactive design.
12. What is the full meaning of the acronym "CATWOE"?
1 3. Explain what you understand by Participatory Design of Human Computer Interaction. What are its characteristics and the Ethics involved?

7.0 Further Readings / References

- Reddy, R., "To Dream the Possible Dream (Turing Award Lecture)." *Communications of the ACM*, 1996. 39(5): pp. 105-112.
- Robertson, G., Newell, A., and Ramakrishna, K., *ZOG: A Man-Machine Communication Philosophy* . Carnegie Mellon University Technical Report Report, Number, August, 1977.

UNIT 2: USER ORIENTED PERSPECTIVE OF HUMAN COMPUTER INTERACTION : COGNITIVE HUMAN THRUST

Table of contents

1.0 Introduction

Cognitive Human Thrust incorporates cognitive psychology and cognitive modeling. Cognitive psychology plays an important role in Human-computer Interaction.

Perception, memory, mental models and metaphors, knowledge representations, problem solving, errors and learning, are all topics under cognitive psychology that have direct implications to HCI design.

Cognitive modeling involves creating a computational model to estimate how long it takes people to perform a given task. Models are based on psychological principles and experimental studies to determine times for cognitive processing and motor movements. Cognitive models can be used to improve user interfaces or predict problem errors and pitfalls during the design process.

2.0 OBJECTIVES

By the end of this unit, the student should be able to:

- Be familiar with different types of Cognitive models
- Know available inspection methods
- Know how to apply inquiry methods on problems of cognition
- Explain prototyping methods
- Carry out evaluation using the tests and metrics methods

.0 MAIN CONTENT

3.1 Cognitive Models

3.1.1 Parallel Design

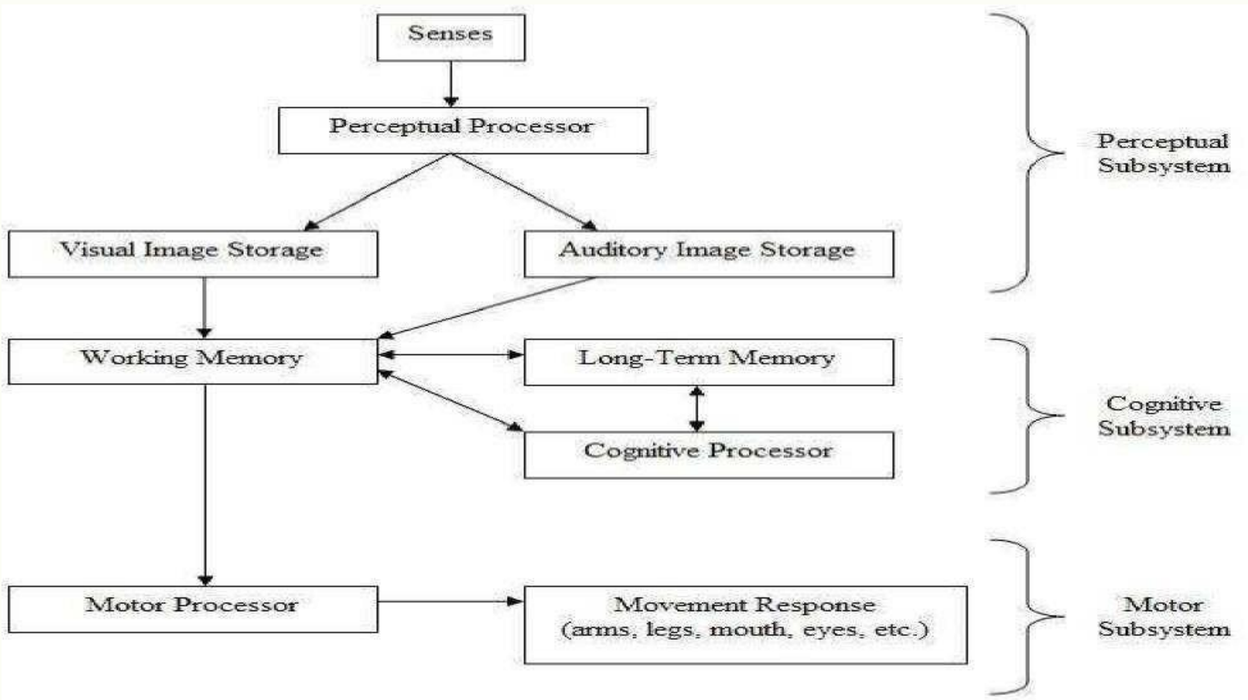
With parallel design, several people create an initial design from the same set of requirements. Each person works independently, and when finished, shares his/her concepts with the group. The design team considers each solution, and each designer uses the best ideas to further improve their own solution. This process helps to generate many different, diverse ideas and ensures that the best ideas from each design are integrated into the final concept. This process can be repeated several times until the team is satisfied with the final concept.

3.1.2 .2 GOMS

GOMS is an acronym that stands for Goals, Operator, Methods, and Selection Rules. It is a family of techniques that analyzes the user complexity of interactive systems. Goals are what the user has to accomplish. An operator is an action performed in service of a goal. A method is a sequence of operations that accomplish a goal. Selection rules specify which method should be used to satisfy a given goal, based on the context.

3.1. 3 Human Processor Model

Sometimes it is useful to break a task down and analyze each individual aspect separately. This allows the tester to locate specific areas for improvement. To do this, it is necessary to understand how the human brain processes information. A model of the human processor is shown below.



Many studies have been done to estimate the cycle times, decay times, and capacities of each of these processors. Variables that affect these can include subject age, ability, and the surrounding environment. For a younger adult, reasonable estimates are:

Parameter	Mean	Range
Eye movement time	230 ms	70-700 ms
Decay half-life of visual image storage200		ms90-1000 ms
Perceptual processor cycle time	100 ms	50-200 ms
Cognitive processor cycle time	70 ms	25-170 ms
Motor processor cycle time	70 ms	30-100 ms
Effective working memory capacity	7 items	5-9 items

Long-term memory is believed to have an infinite capacity and decay time.

Keystroke level modeling

Keystroke level modeling is essentially a less comprehensive version of GOMS that makes simplifying assumptions in order to reduce calculation time and complexity.

3.2 Inspection methods

These usability evaluation methods involve observation of users by an experimenter, or the testing and evaluation of a program by an expert reviewer. They provide more quantitative data, as tasks can be timed and recorded.

3.2.1 Card Sorting

Card sorting is a way that involves users in grouping information for a website's usability review. Participants in a card sorting session are asked to organize the content from a Web site in a way that makes sense to them. Participants review items from a Web site and then group these items into categories. Card sorting helps to learn how users think about the content and how they would organize the information on the Web site. Card sorting helps to build the structure for a web site, decide what to put on the home page, and label the home page categories. It also helps to ensure that information is organized on the site in a way that is logical to users.

3.2.2 Ethnography

Ethnographic analysis is derived from anthropology. Field observations are taken at a site of a possible user, which track the artifacts of work such as Post-It notes, items on desktop, shortcuts, and items in trash bins. These observations also gather the sequence of work and interruptions that determine the user's typical day.

3.2. 3 Heuristic Evaluation

Heuristic Evaluation is a usability engineering method for finding and assessing usability problems in a user interface design as part of an iterative design process. It involves having a small set of evaluators examining the interface and using recognized usability principles (the "heuristics"). It is the most popular of the usability inspection methods, as it is quick, cheap, and easy.
Heuristic evaluation was developed to aid in the design of computer user-interface design. It relies on expert reviewers to discover usability problems and then categorize and rate them by a set of principles (heuristics.) It is widely used based on its speed and cost-effectiveness. Jakob Nielsen's list of ten heuristics is the most commonly used in industry. By determining which guidelines are violated, the usability of a device can be determined.

3.2.4 Usability Inspection

Usability Inspection is a review of a system based on a set of guidelines. The review is conducted by a group of experts who are deeply familiar with the concepts of usability in design. The experts focus on a list of areas in design that have been shown to be troublesome for users.

3.2.5 Pluralistic Inspection

Pluralistic Inspections are meetings where users, developers, and human factors people meet together to discuss and evaluate step by step of a task scenario. As more people inspect the scenario for problems, the higher the probability to find problems. In addition, the more interaction in the team, the faster the usability issues are resolved.

3.2.6 Consistency Inspection

In consistency inspection, expert designers review products or projects to ensure consistency across multiple products to look if it does things in the same way as their own designs.

3.2.7 Activity Analysis

Activity analysis is a usability method used in preliminary stages of development to get a sense of a situation. It involves an investigator observing users as they work in the field. Also referred to as user observation, it is useful for specifying user requirements and studying currently used tasks and subtasks. The data collected is qualitative and useful for defining the problem. It should be used when you wish to frame what is needed, or "What do we want to know?"

3. 3 Inquiry methods

The following usability evaluation methods involve collecting qualitative data from users. Although the data collected is subjective, it provides valuable information on what the user wants.

3. 3.1 Task Analysis

Task analysis means learning about users' goals and users' ways of working. Task analysis can also mean figuring out what more specific tasks users must do to meet those goals and what steps they must take to accomplish those tasks. Along with user and task analysis, we often do a third analysis: understanding users' environments (physical, social, cultural, and technological environments).

3. 3.2 Focus Groups

A focus group is a focused discussion where a moderator leads a group of participants through a set of questions on a particular topic. Although typically used as a marketing tool, Focus Groups are sometimes used to evaluate usability. Used in the product definition stage, a group of 6 to 10 users are gathered to discuss what they desire in a product. An experienced focus group facilitator is hired to guide the discussion to areas of interest for the developers. Focus groups are typically videotaped to help get verbatim quotes, and clips are often used to summarize opinions. The data gathered not usually quantitative, but can help get an idea of a target group's opinion.

3. 3. 3 Questionnaires/Surveys

Surveys have the advantages of being inexpensive, require no testing equipment, and results reflect the users' opinions. When written carefully and given to actual users who have experience with the product and knowledge of design, surveys provide useful feedback on the strong and weak areas of the usability of a design. This is a very common method and often does not appear to be a survey, but just a warranty card.

.4 Prototyping methods

3.4.1 Rapid Prototyping

Rapid prototyping is a method used in the early stages of development to validate and refine the usability of a system. It can be used to quickly and cheaply evaluate user-interface designs without the need for an expensive working model. This can help remove hesitation to change the design, since it is implemented before any real programming begins. One such method of rapid prototyping is paper prototyping.

3.4.2 Subject Testing methods

These usability evaluation methods involve testing of subjects for the most quantitative data. Usually recorded on video, they provide task completion time and allow for observation of attitude.

3.4. 3 Remote usability testing

Remote usability testing is a technique that exploits users' environment (e.g. home or office), transforming it into a usability laboratory where user observation can be done with screen sharing applications.

3.4.4 Thinking Aloud Protocol

The Thinking Aloud Protocol is a method of gathering data that is used in both usability and psychology studies. It involves getting a user to verbalize their thought processes as they perform a task or set of tasks. Often an instructor is present to prompt the user into being more vocal as they work. Similar to the Subjects-in-Tandem method, it is useful in pinpointing problems and is relatively simple to set up. Additionally, it can provide insight into the user's attitude, which can not usually be discerned from a survey or questionnaire.

3.4.5 Subjects-in-Tandem

Subjects-in-tandem is pairing of subjects in a usability test to gather important information on the ease of use of a product. Subjects tend to think out loud and through their verbalized thoughts designers learn where the problem areas of a design are. Subjects very often provide solutions to the problem areas to make the product easier to use.

3.5 Other methods

3.5.1 Cognitive walkthrough

Cognitive walkthrough is a method of evaluating the user interaction of a working prototype or final product. It is used to evaluate the system's ease of learning. Cognitive walkthrough is useful to understand the user's thought processes and decision making when interacting with a system, specially for first-time or infrequent users.

3.5.2 Benchmarking

Benchmarking creates standardized test materials for a specific type of design. Four key characteristics are considered when establishing a benchmark: time to do the core task, time to fix errors, time to learn applications, and the functionality of the system. Once there is a benchmark, other designs can be compared to it to determine the usability of the system.

3.5. 3 Meta-Analysis

Meta-Analysis is a statistical procedure to combine results across studies to integrate the findings. This phrase was coined in 1976 as a quantitative literature review. This type of evaluation is very powerful for determining the usability of a device because it combines multiple studies to provide very accurate quantitative support.

3.5.4 Persona

Personas are fictitious characters that are created to represent the different user types within a targeted demography that might use a site or product. Alan Cooper introduced the concept of using personas as a part of interactive design in 1998 in his book *The Inmates Are Running the Asylum*, but had used this concept since as early as 1975.Personas is a usability evaluation method that can be used at various design stages. The most typical time to create personas is at the beginning of designing so that designers have a tangible idea of who the users of their product will be. Personas are the archetypes that represent actual groups of users and their needs, which can be a general description of person, context, or usage scenario. This technique turns marketing data on target user population into a few physical concepts of users to create empathy among the design team.

3.6 Evaluating with tests and metrics

Regardless of how carefully a system is designed, all theories must be tested using usability tests. Usability tests involve typical users using the system (or product) in a realistic environment. Observation of the user's behavior, emotions, and difficulties while performing different tasks, often identify areas of improvement for the system.

3.6.1 Prototypes

It is often very difficult for designers to conduct usability tests with the exact system being designed. Cost constraints, size, and design constraints usually lead the designer to creating a prototype of the system. Instead of creating the complete final system, the designer may test different sections of the system, thus making several small models of each component of the system. The types of usability prototypes may vary from using paper models, index cards, hand drawn models, or storyboards.
Prototypes are able to be modified quickly, often are faster and easier to create with less time invested by designers and are more apt to change design; although sometimes are not an adequate representation of the whole system, are often not durable and testing results may not be parallel to those of the actual system.

3.6.2 Metrics

Designers must use usability metrics to identify what it is they are going to measure, or the usability metrics. These metrics are often variable, and change in conjunction with the scope and goals of the project. The number of subjects being tested can also affect usability metrics, as it is often easier to focus on specific demographics. Qualitative design phases, such as general usability (can the task be accomplished?), and user satisfaction are also typically done with smaller groups of subjects. Using inexpensive prototype on small user groups, provide more detailed information, because of the more interactive atmosphere, and the designers ability to focus more on the individual user.

Testing the metrics

As the designs become more complex, the testing must become more formalized. Testing equipment will become more sophisticated and testing metrics become more quantitative. With a more refined prototype, designers often test effectiveness, efficiency, and subjective satisfaction, by asking the user to complete various tasks. These categories are measured by the percentage that complete the task, how long it takes to complete the tasks, ratios of success to failure to complete the task, time spent on errors, the number of errors, rating scale of satisfactions, number of times user seems frustrated, etc. Additional observations of the users give designers insight on navigation difficulties, controls, conceptual models, etc. The ultimate goal of analyzing these metrics is to discover a prototype design that users like to successfully perform given tasks.

Documenting metrics

After conducting usability tests, it is important for a designer to record what was observed, in addition to why such behavior occurred and modify the model according to the results. Often it is quite difficult to distinguish the source of the design errors, and what the user did wrong. However, effective usability tests will not generate a solution to the problems, but provide modified design guidelines for continued testing.

3.7 Benefits of usability

The key benefits of usability are:

Increased user efficiency

Reduced development costs

Reduced support costs

Corporate integration

By working to improve said factors, corporations can achieve their goals of increased output at lower costs, while potentially creating optimal levels of customer satisfaction. There are numerous reasons why each of these factors correlates to overall improvement. For example, making a piece of software's user interface easier to understand would reduce the need for extensive training. The improved interface would also tend to lower the time needed to perform necessary tasks, and so would both raise the productivity levels for employees and reduce development time and costs.

4.0 Conclusion

All the factors analysed above aid the design process and Increase usability in the workplace. They aid in fostering several responses from employees. Along with any positive feedback, workers who enjoy their work do it better, stay longer in the face of temptation, and contribute ideas and enthusiasm to the evolution of enhanced productivity.

5.0 Summary

Cognitive human thrust comprises cognitive psychology and modeling. Cognitive psychology studies human traits such as perception, memory, mental models and metaphors, among others- those that have direct implications to HCI design.

Cognitive modeling enables estimation of how long it takes people to perform a given task.

Inspection method is the process of observing users by an experimenter, or the testing and evaluation of a program by an expert reviewer.
Inquiry methods involve collecting qualitative data from users; such data provides valuable information on what the user wants.
Prototypes enable the designer test different sections of the system by making several small models of each component of the system.
Evaluation methods test and correct errors in implemented design.

6.0 Tutor Marked Assignment

1. What is the full meaning of the acronym GOMS as a cognitive model? Explain each of the terms in the GOMS
Distinguish between the following Cognitive models: (i) Parallel Design, (ii) Human Processor and (iii) Keystroke level
2. What is the primary benefit of the inspection methods used to evaluate usability in HCI? Briefly explain the following inspection methods:
(i) Card sorting (ii) Ethnography (iii) Heuristic evaluation (iv) Usability inspection (v) Activity analysis
3. What is the value of a Usability inspection and its focus?
4. Explain the following methods of inquiry in usability evaluation:
Task analysis, Group Focus, Questionnaire and Survey methods.
5 (a) What do you understand by the term Prototype as related to the design of Human Computer Interaction?
(b) In designing and evaluating usability, prototyping methods have been very valuable. Show your understanding of the following Prototyping methods:
(i) Rapid Prototyping, (ii) Testing methods, (iii) Remote Usability testing,
(iv)Thinking aloud Protocol, (v) Subjects in Tandem
6. The following are methods used to evaluate the design of the user interaction with the Computer before and /or after implementation: Cognitive Walkthrough, Benchmarking Meta analysis and Persona.
Which of the four above do you consider as the most suitable method and why?

7.0 Further Readings / References

- Ronald M. Baecker, Jonathan Grudin, William A. S. Buxton, Saul Greenberg (Eds.) (1995): *Readings in human—computer interaction. Toward the Year 2000*. 2. ed. Morgan Kaufmann, San Francisco 1995 ISBN 1-558-60246-1
- Jakob Nielsen: *Usability Engineering*. Academic Press, Boston 199 3 ISBN 0-12-518405-0
- Donald A. Norman: *The Psychology of Everyday Things*. Basic Books, New York 1988 ISBN 0-465-06709- 3
- Jef Raskin: *The humane interface. New directions for designing interactive systems*. Addison-Wesley, Boston 2000 ISBN 0-201- 379 37-6
- Ben Shneiderman and Catherine Plaisant: *Designing the User Interface: Strategies for Effective Human—Computer Interaction*. 4th ed. Addison Wesley, 2004 ISBN 0- 321-19786-0

UNIT 3: SYSTEM ORIENTED PERSPECTIVE OF HUMAN COMPUTER INTERACTION

IMPROVING UPON THE USER'S TECHNOLOGICAL PERSPECTIVE

Table of contents

1.0 Introduction

The input and output technological perspective of the system from the user's view is enhanced by the kind of support given the users of the system.

There are different types of support available to the users at different times particularly important during the implementation and presentation stages. Hence the computer interaction components require careful design.

Types of user support that can be designed and offered include quick rreferences, task specific help, full explanation, and tutorials.

The kind of help solutions are provided on specific problem oriented operations while documentation solutions are given on system oriented and general operations. The same design principles apply to both.

2.0 Objectives

By the end of this unit, the student should be able to :

- Understand help supports available to the users
- Describe user modelling and knowledge representation
- Know how to design user supports

.0 Main Content

.1 Technical Support offered the System Users

The following are the requirements of an effective design of technical support that can be offered to the system users

Continuous access concurrent to main application should be made available

Help Support

For accuracy and completeness, the help should match and cover actual system behaviour

There must be consistency between different parts of the help system and paper documentation

Robustness should exist for correct error handling and unpredictable behaviour

There should be flexibility such that the system allows the user to interact in a way appropriate to experience and task

Encountering problems while operating the system should not prevent the user continuing with the work

The approaches to user support include the following:

Command assistance should enable the user to request help on a particular command e.g., UNIX and DOS help can be good for quick reference.

If we assume that the user has knowledge of the command and what to look for, then the command prompts should provide information about correct usage when an error occurs and be good for simple syntactic errors

Context sensitive help:
Ensure that the help request is interpreted according to context in which it occurs. e.g. tool tips
The on-line tutorials allow the user to work through the basics of application in a test environment. Though often inflexible, it can be useful.
The on-line documentation though available in paper documentation is also made available on the computer. It should be continually available in common medium as well.
Since on-line documentation can be difficult to browse, then hypertext should be included to support browsing..

Adaptive Help Systems
These use knowledge of the context, the individual user, the task, the domain and the instruction to provide help adapted to the user's needs.
The problems with adaptive help systems are that they require considerable knowledge, the interaction is not controlled and it is difficult knowing what should be adapted and the scope of the adaptation.

Issues concerned in adaptive help:
Initiative ; the question is whether the user retain control or can the system direct the interaction of the user, and can the system interrupt the user to offer help?
Effect ; the question is what is going to be adapted and what information is needed to do this? Only what is needed is modelled.
Scope ; Is the scope of the modelling at application or system level? It is more complex at system level e.g. expertise varies between applications.

Wizards and assistants:
Wizards
Wizard is a task specific tool that leads the user through a task, step by step, using the user's answers to specific questions
Example is in the preparation of resume.
Wizard is useful for the safe completion of complex or infrequent tasks.
It has a limited flexibility in a constrained task execution so it must allow the user to go back to the beginning of the task.
Assistants
Assistants monitor the user behaviour and offer contextual advice though it can be irritating e.g. as in MS paperclip.
They must be under the user control e.g. XP smart tags

3.2 User modelling
In the user modelling of the knowledge representation, all help systems have a model of the user. It may be a single or generic user (non-intelligent user). The models could be an adaptable user-configured or adaptive system-configured.

Approaches to user modelling
Quantification: Here the user moves between levels of expertise based on quantitative measure of what he knows.
Stereotypes: The user is classified into a particular category.

Overlay: The idealized model of an expert use is constructed and actual use compared to ideal. The model may contain the commonality or difference
In a special case, the user behaviour is compared to a known error catalogue.

Knowledge representation
Knowledge representation occurs when knowledge is presented as rules, and facts are interpreted using inference mechanism.
The domain and task modelling of the knowledge representation covers common errors and tasks particularly the current task.
It usually involves the analysis of command sequences.
However the problems here are concerned on how to represent the tasks particularly when interleaved, and how to know the user's intention.

Knowledge representation: Advisory strategy
The advisory strategy for knowledge representation involves choosing the correct style of advice for a given situation in form of a reminder, tutorial, etc.
Few intelligent help systems model advisory strategy, but choice of strategy is still important.

Techniques for knowledge representation
The techniques for knowledge representation are rule based (e.g. logic, production rules) when knowledge are presented as rules and facts are interpreted using inference mechanism. They can also be used in relatively large domains..
It is frame based (such as a semantic network) when knowledge stored in structures with slots are to be filled but useful for a small domain..

Network based
The knowledge is network based when represented as relationships between facts and can be used to link frames.
It is example based when the knowledge is represented implicitly within decision structure and trained to classify rather than programmed with rules. This one requires little knowledge acquisition

Problems with knowledge representation and modelling
The problems here include knowledge acquisition, the resources and the interpretation of user behaviour

3. 3 Designing user support
User support is not an 'add on' but should be designed integrally with the system.
The designer should concentrate on content and context of help rather than on technological issues.

3. 3.1 Presentation issues in designing user support:
How is help requested? Is it at the command level, by button, by on/off function, or by separate application?
How is help displayed? Is it through a new window, or a whole screen, a split screen, pop-up boxes, or hint icons?
The designer should note that effective presentation requires clear, familiar, and consistent language. It should contain instructional rather than descriptive languages. Blocks of text should be avoided.

. 3.2 Implementation issues in designing user support

Implementation issues are whether the help is in form of an operating system command, a Meta command or an application.
There should be a clear indication of summary and example information.
What are the resources available in terms of screen space, the memory capacity and the speed of processing?
Is the structure of help data in form of a single file, a file hierarchy or a database?
Other issues concern the flexibility and extensibility of implementation and whether it is made in hard copy or by browsing.

4.0 Conclusion

Computer interaction components require careful design. The design should ensure that there are different types of support available to the users at different times particularly during the implementation and presentation stages.
Help solutions should be provided on specific problem oriented operations, while documentation solutions are given on system oriented and general operations.

5.0 Summary

Effective design of technical support reflects in continuous access concurrent to main application being made available
Accuracy and completeness are guaranteed when the help support matches and covers actual system behaviour
Wizard is a task specific tool that leads the user through a task, step by step, using the user's answers to specific questions while assistants monitor the user behaviour and offer contextual advice.
User modelling occurs when all help systems have a model of the user. Knowledge representation is the presentation of knowledge as rules and facts, and interpreted using inference mechanism.
User support should be designed integrally with the system. The designer should concentrate on content and context of help rather than on technological issues.

6.0 Tutor Marked Assignment

1. Mention three kinds of Help Support that can be designed for the Computer System Users. What are their demerits?
2. What is Knowledge representation? Describe the rule based, the frame based and the networked based techniques of Knowledge representation.
3. Designing the user support requires considering some presentation and implementation issues. Describe two of presentation issues and three of implementation issues that should guide the designer

7.0 Further Readings / References

- Henderson Jr, D.A. "The Trillium User Interface Design Environment," in *Proceedings SIGCH I'86: Human Factors in Computing Systems.* 1986. Boston, MA. pp. 221-227.
- Myers, B.A., *et al.*, "Garnet: Comprehensive Support for Graphical, Highly-Interactive User Interfaces." *IEEE Computer*, 1990. 2 3(11): pp. 71-85.
- Stallman, R.M., *Emacs: The Extensible, Customizable, Self-Documenting Display Editor* . MIT Artificial Intelligence Lab Report, Number, Aug, 1979, 1979.

UNIT 4: DEVICES TECHNOLOGICAL PERSPECTIVE

INTERACTION STYLES AND DEVICES TECHNOLOGICAL PERSPECTIVE

Table of contents

4.0 INTRODUCTION

An Interaction style can be described as an interaction technique that shares the same metaphor or design principle. Examples are command line and direct manipulation user interfaces.

Two major classes of interaction styles that will be considered are the dialogue style of Interaction between computer and user and the distinct styles of interaction

5.0 OBJECTIVES

By the end of this unit, you should be able to:

- Understand the various available interaction styles
- Distinguish among the interaction styles
- Understand elements of WIMP interface
- Understand and be able to choose Widgets

6.0 MAIN CONTENT

.1 Interaction styles

Common Styles of Interaction are:
Command line interface
Menus
Natural language
Question and answer, and query dialogue
Form-fills and Spreadsheets
WIMP (Widows, Icons, Menus, Pointers) Interface
Point and Click
Three—dimensional interfaces

Graphical user interface (GUI)
Copy and paste, Cut and paste
Single Document Interface, Multiple Document Interface, Tabbed Document Interface
Drag-and-drop

Cursor
Widgets (computing)
Direct manipulation interface
Zooming User Interface (ZUI)

Interaction paradigms include
Hypertext, hypermedia and hyperlinks
Speech recognition, Speech synthesis, Natural Language Processing, Non-speech audio input
Mouse gestures and handwriting recognition
Haptics and Telehaptics
Computer-mediated reality
Virtual Reality (VR)
Augmented Reality (AR)
CSCW: Computer Supported Collaborative (or Cooperative) Work, collaborative software
Ubiquitous Computing ("ubicomp")
Wearable computers
Brain-computer interface

Miscellaneous
Handheld devices
Human Computer Information Retrieval
Information retrieval
Internet and the World Wide Web
Multimedia
Software agents
Universal usability
User experience design
Visual programming languages

Brief explanation will be given on each of the common interfaces:

Command line interface
This is the way of expressing instructions to the computer directly and it comprises the function keys, single characters, short abbreviations, whole words, or a combination of them.
The characteristics of command line interface are as follow:
It is suitable for repetitive tasks
It is more valuable for expert users than novices
It offers direct access to system functionality
The command names and abbreviations should however be meaningful for an effective interface.
A typical example is found in the Unix Operating System

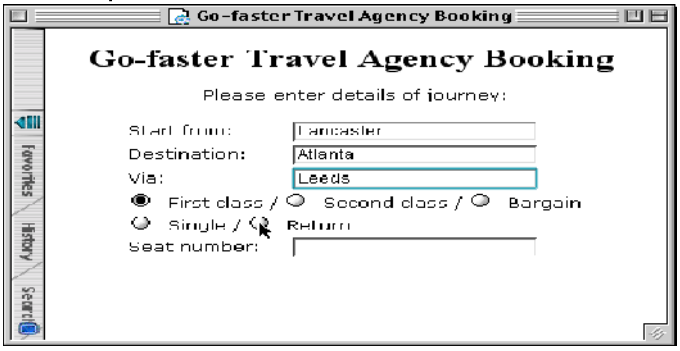
Menus
These are set of options displayed on the screen
Its characteristics include the following:
The Menu Options are visible, have less recall and it is easier to use
Because it relies on recognition the names should be meaningful
Menu Selection is done by either clicking numbers, letters, using the arrow keys or mouse or combination (e.g. mouse plus accelerators)
Menu options are often hierarchically grouped in a sensible manner.

It is a restricted form of full WIMP system

Natural language
This is a language that should be familiar to the user
It is in a form of speech recognition or typed natural language
Problems with the use of natural language are that it could be vague, ambiguous, and hard to use.
Part of the solutions to this is for the user to try and to understand a subset of the language thereby picking on key words.

Query interfaces
This basically comprises Question and answer interfaces where the user is led through interaction via series of questions to be answered.
It is suitable for novice users but has restricted functionality
It is often used in information systems
Some of these Query languages include the SQL used to retrieve information from database. This requires understanding of database structure and language syntax, hence requires some expertise

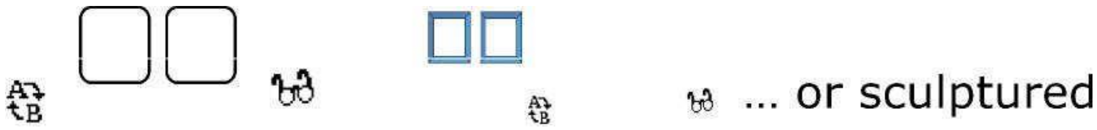
Form-fills
This is primarily used for data entry or data retrieval.
It is like a screen like paper form in which data is input in relevant place.
It requires good design and obvious correction facilities
An example of a Form-fill:



Spreadsheets
The first spreadsheet was VISICALC followed by Lotus 1-2- 3. However MS Excel is the most common spreadsheet today having a sophisticated variation of form-filling.
It has grid of cells containing a value or a formula. The formula can involve values of other cells e.g. sum of all cells in this column. The user can enter and alter data in the spreadsheet with considerable consistency.

Point and click interfaces
This is used in multimedia ,web browsers and hypertext. It is sometimes called 'just click something'! using icons, text links or location on map.
It requires minimal typing

Three dimensional interfaces
This is made up of virtual reality, 'ordinary' window systems and 3D workplace
The 'ordinary' window systems comprises highlighting, visual affordance and indiscriminate use
The 3D workspaces have uses for extra virtual space, it is light and occlusion to give depth and distance effects



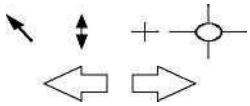
WIMP Interface
'WIMP' stands for Windows, Icon s, Menus and Pointers (or Windows, Icons, Mice, and Pull-down menus!)
It is a default style interface for majo rity of interactive computer systems, especi lly PCs and desktop machines
WIMP Interface
Windows
Icons
Menus
Pointers
... or windows, icons, mice, an d pull-down menus!

Elements of WIMP interface:
The elements of the WIMP interface are:
Windows, Icons, Menus and Pointers
Buttons, Toolbars, Palettes and Dialo g boxes

Details are given below:
Windows
These are areas of the screen that be have as if they were independent and can c ontain text or graphics which can be moved or resized. They can overlap and obscure each other, or can be laid out next to one another (tiled)
They are made up of
vi. scrollbars that allow the u ser to move the contents of the window up and down or from side to side
vii. title bars that describe th name of the window

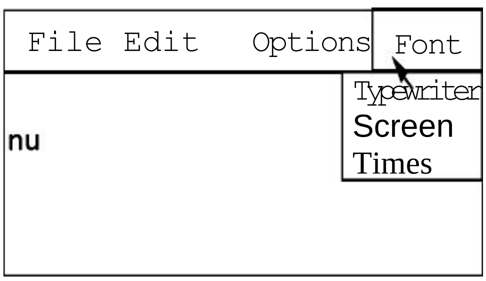
Icons
Icons comprise small picture or image that represents some object in the interface. It often represents a window or action. The windows can Abe closed down or 'iconised'
small representation may fit into m any accessible windows. The icons can be many and various.
They are highly stylized with realistic representations.

Pointers
These are important component WI It P style that relies on pointing and selectin things
uses mouse, trackpad, joystick, tra ckball, cursor keys or keyboard shortcuts to ccess
wide variety of graphical images



Menus
These are choice of operations or se vices offered on the screen
The required option is selected with pointer. However, this takes a lot of screen space
This problem is partly solved when a pop-up menu appears when needed

Kinds of Menus
Menu Bar at top of screen (normally) , menu drags down
i Pull-down menu - mouse hold and drag down menu



ii Drop-down menu - mouse click reveals menu
viii. Fall-down menus - mouse just moves over bar!

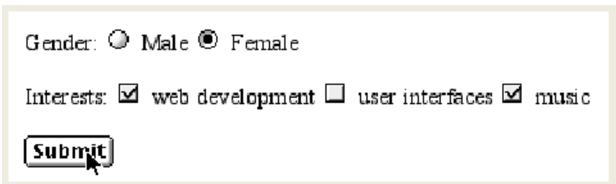
Contextual menu appears where you are
Pop-up menus take actions for selected object
Pie menus are arranged in a circle
This is easier to select item over larger target area
It is also quicker because it can move same distance to any option but this is not widely used!

Menus extras
Cascading menus
This has a hierarchical menu structure in which a menu selection opens new menu and so in ad infinitum

Keyboard accelerators
This comprises key combinations with same effect as menu item.
They are of two kinds:
- active when menu open — usually first letter and
- active when menu closed — usually Ctrl + letter

3.2 Menus design issues
In order to design an effective menu, the following issues should be considered:
- Which kind to use
- What to include in menus at all
- Words to use in action or description
- How to group items
- Choice of keyboard accelerators

Buttons
This is an individual and isolated region within a display that can be selected to invoke an action
The Special kinds that exist are
The radio buttons with a set of mutually exclusive choices and the check boxes with a set of non-exclusive choices.



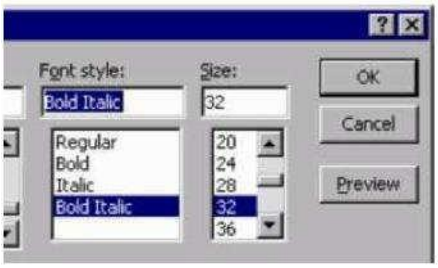
Toolbars
These are long lines of icons with fast access to common actions and are often customizable:
You can choose *which* toolbars to see and choose *what* options are on it

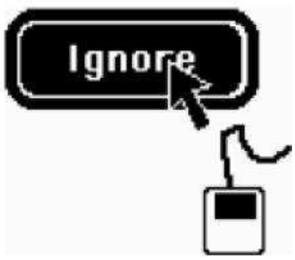
Palettes and tear-off menus
Palettes are little windows of actions shown or hidden via menu option in available shapes in drawing package
In tear-off and pin-up menus, menu 'tears off' to become palette

Dialogue boxes
These are information windows that pop up to inform of an important event or request information, for example when saving a file, a dialogue box is displayed to allow the user to specify the filename and location. Once the file is saved, the box disappears.

. 3 UNDERSTANDING AND CHOOSING WIDGETS

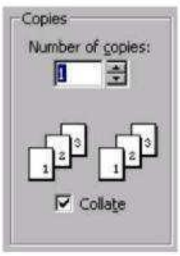
Widgets are bits that make the Graphical User Interface. They are the individual items on a Graphical User Interface (GUI). They can also be called the elements of the WIMP interface.
Examples of widgets include the check boxes, the tool bars, the buttons, etc. See the pictorial illustration below.

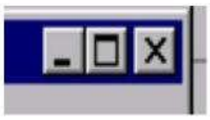
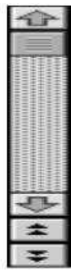




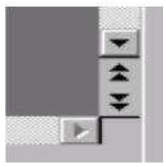
Three aspects of widgets can be identified in the following ways:
By appearance in the way they look like
By the nature of their interaction as to how they behave and
By their semantics as regards their meaning

By appearance:
Appearance includes words that are verbs that represent some action such as quit, exit, embolden, and italicize.

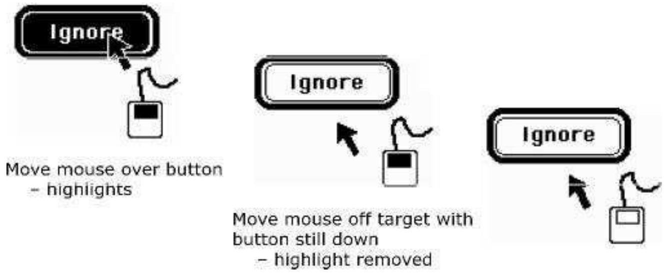



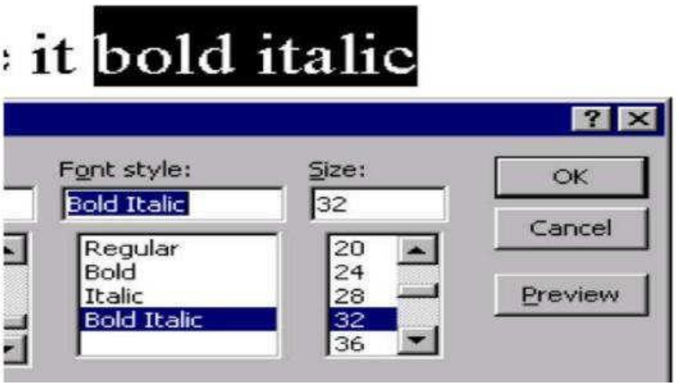
They could also be adjectives that describe the state of those words such as bold, italic etc.
They could be nouns that represent the name of the appearance such as Times New Roman, etc.
They could be combination of verbs and adjectives e.g. embolden + italic



By behavior
This describes the action the toolkit carries out on your behalf and this can be controlled. Examples are drawing and such interactions between the widgets themselves. But timing issues of this behavior should be watched such as the large selections under Windows applications.
See the pictorial example below.

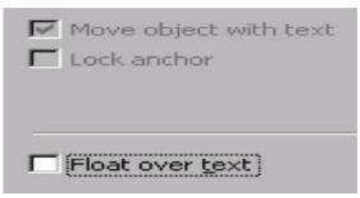


By semantics
Semantics are menus, buttons, etc th at do things as desired by the user..



An example is '... lets make it *bold italic'*

The semantics assignment is determi ned by the designer or user; YOU say what i means. The semantics is usually up to you.
Although widgets may link direct to atabase, even then, you say what links to th e database.
So to choose the widget for the job, ssign meaning first on what you want it to d ito, followed by appearance and then how you want done.
You may for example want actions ca rried out through menu, buttons, or toolbar ioor you want to set the status of options using checkbox, rad button, or combi-box.



4.0 Conclusion
Designing interaction styles should b e based on the following criteria:
Domain — this considers the ar ea of work under study e.g. graphic design
Goal — what the designer wants toachieve e.g. create a solid red triangle
Task — how you go wish to presentthe style ultimately in terms of operations o r actions, e.g. select fill tool, click over triangle

The style should be easy to focus on look and feel because if you want someone to do something, make it easy for them and understand their values.

5.0 Summary

Interaction styles are the nature of dialogs between the user and the system.

A graphical user interface (GUI) is a type of user interface which allows people to interact with electronic devices like computers, hand-held devices through graphical icons, and visual indicators by direct manipulation of the graphical elements.

In WIMP Interface, 'WIMP' stands for Windows, Icons, Menus and Pointers (or Windows, Icons, Mice, and Pull-down menus!)

It is a default style interface for majority of interactive computer systems, especially PCs and desktop machines

Widgets are bits that make the Graphical User Interface. They are the individual items on a Graphical User Interface (GUI). They can also be called the elements of the WIMP interface.

Menus are choice of operations or services offered on the screen

A button is an individual and isolated region within a display that can be selected to invoke an action

Toolbars are icons with fast access to common actions and are often customizable:

Palettes are little windows of actions shown or hidden via menu option in available shapes in drawing package

Dialogue boxes are information windows that pop up to inform of an important event or request information such as saving a file.

6.0 Tutor Marked Assignment

6. Mention 8 of the interaction styles available in HCI
7. Distinguish between the following pairs of interfaces
   - (ix) Widgets and Graphical User Interface
   - (x) Menus and Dialogue boxes
   - (xi) Drag — and- Drop and Copy or Cut and paste
   - (xii) Speech Recognition and Natural Language
   - (xiii) Command Line Interface and Direct manipulation Interface
   - (xiv) Query Dialogue and Form- fills
   - (xv) Mouse gestures and handwriting recognition
   - (xvi) Buttons and Palettes
8. Differentiate between the following types of widgets
   - (ii) Widgets by appearance, (ii) Widgets by behavior, (iii) Widgets by semantics
9. What are keyboard accelerators?
10. In designing menus, what are the issues that should be considered?

7.0 Further Readings / References

- Myers, B.A., "All the Widgets." *SIGGRAPH Video Review*, 1990. 57
- Meyrowitz, N. and Van Dam, A., "Interactive Editing Systems: Part 1 and 2." *ACM Computing Surveys*, 1982. 14(3): pp. 321- 352.
- Koved, L. and Shneiderman, B., "Embedded menus: Selecting items in context." *Communications of the ACM*, 1986. 4(29): pp. 312- 318.
- Myers, B.A., "A Taxonomy of User Interfaces for Window Managers." *IEEE Computer Graphics and Applications*, 1988. 8(5): pp. 65-84.
- Andrew Sears and Julie A. Jacko (Eds.). (2007). Handbook for Human Computer Interaction (2nd Edition). CRC Press. ISBN 0-8058-5870-9

- Brooks, F. "The Computer "Scientist" as Toolsmith--Studies in Interactive Computer Graphics," in *IFIP Conference Proceedings.* 1977. pp. 625-6 34.

MODULE 3: DESIGNS OF HUMAN-COMPUTER INTERACTION

UNIT 1: DESIGN GUIDELINES, RULES AND PRINCIPLES

Table of contents

1.0 INTRODUCTION

This unit offers guidelines for design of user interface software in six functional areas: data entry, data display, sequence control, user guidance, data transmission, and data protection.

The guidelines are proposed here as a potential tool for designers of user interface software. Guidelines can help establish rules for coordinating individual design contributions; can help to make design decisions just once rather than leaving them to be made over and over again by individual designers. It can also help to define detailed design requirements and to evaluate user interface software in comparison with those requirements.

The design of user interface software will often involve a considerable investment of time and effort. Design guidelines can help ensure the value of that investment.

4.0 OBJECTIVES

By the end of this unit, you should be able to:

- Select design guidelines
- Know how to monitor guidelines and carry out prototype tests
- Understand the concept of translating selected guidelines into design rules
- Know the importance of documenting design rules
- Explain various types of design rules
- Know how to use design rules
- Learn some design principles

5.0 MAIN CONTENT

3.1 DESIGN GUIDELINES

Guidelines are more suggestive and general. There are two types of guidelines, they are:

iii. Abstract guidelines or principles that are applicable during early design life cycle activities

iv. Detailed guidelines otherwise called style guides that are applicable during the later system life cycle activities

Understanding justification for guidelines aids in resolving conflicts

3.1.1 SELECTION OF DESIGN GUIDELINES

Not all of the guidelines can be applied in designing any particular system. For any particular system application, some of the guidelines will be relevant and some will not. Design guidelines must be generally worded so that they might apply to many different system applications. Thus generally-worded guidelines must be translated into specific design rules before they can actually be applied.

The process of selecting relevant guidelines for application and translating them into specific design rules is referred to here as "tailoring". Who will do this guidelines tailoring? It should be the joint responsibility of system analysts and human factors specialists assessing design requirements, of software designers assessing feasibility, and of their managers. It may also be helpful to include representatives of the intended system users in this process, to ensure that proposed design features will meet operational requirements.

Once all relevant guidelines have been identified, a designer must review them and decide which ones actually to apply.

There are two reasons why a designer might not wish to apply all relevant guidelines.

First, for any given application, some guidelines may conflict, and the designer must therefore choose which are more important.

Second, budgetary and time restrictions may force the designer to apply only the most important guidelines -- those that promise to have the greatest effect on system usability.

3.1.2 EXPERTISE EXPERIENCE VERSUS GUIDELINES

Guidelines cannot take the place of expertise experience.

An experienced designer, one skilled in the art, might do well without any guidelines. An inexperienced designer might do poorly even with guidelines. Few designers will find time to read an entire book of guidelines. If they do, they will find it difficult to digest and remember all of the material. If guidelines and/or the rules derived from guidelines are to be helpful, they must be kept continually available for ready reference.

Guidelines cannot take the place of expert design consultants, or at least not entirely. A design expert will know more about a specific topic than can be presented in the guidelines. An expert will know what questions to ask, as well as many of the answers. An expert will know how to adapt generally-stated guidelines to the specific needs of a particular system design application. An expert will know how to trade off the competing demands of different guidelines, in terms of operational requirements.

.1. 3 MONITORING GUIDELINES AND PROTOTYPE TESTING

For maximum effectiveness, guideline tailoring must take place early in the design process before any actual design of user interface software. In order to tailor guidelines, designers must have a thorough understanding of task requirements and user characteristics. Thus task analysis is a necessary prerequisite of guidelines tailoring.

The result of guidelines application will be a design for user interface software that may incorporate many good recommendations. However, even the most careful design will require testing with actual users in order to confirm the value of good features and discover what bad features may have been overlooked. Thus prototype testing must follow initial design, followed in turn by possible redesign and operational testing.

Indeed, testing is so essential for ensuring good design that some experts advocate early creation of an operational prototype to evaluate interface design concepts interactively with users, with iterative design changes to discover what works best. But prototyping is no substitute for careful design. Prototyping will allow rapid change in a proposed interface; however, unless the initial design is reasonably good, prototyping may not produce a usable final design.

Considering the system development process overall, guidelines application will not necessarily save work in user interface design, and in fact may entail extra work, at least in the initial stage of establishing design rules. But guidelines application should help produce a better user interface. Because guidelines are based on what is known about good design, the resulting user interface is more likely to be usable. Certainly the common application of design rules by all designers working on a system should result in more consistent user interface design. And the single objective on which experts agree is design consistency.

3.1.4 TRANSLATION OF SELECTED GUIDELINES INTO DESIGN RULES

Because guidelines are intended for use on a variety of systems, they are worded in general terms. Before a guideline can actually be applied it must be translated into specific design rules. For instance, a guideline which states that displays should be consistently formatted might be translated into design rules that specify where various display features should appear, such as the display title, prompts and other user guidance, error messages, command entries, etc.

Any guideline can have different possible translations. A guideline which states that each display should be uniquely identified could be translated into a design rule that display titles will be bolded and centered in the top line of the display. Or it could be translated into a design rule that display titles will be capitalized in the upper left corner of the display.

What would happen if guidelines were not translated into design rules, but instead were given directly to interface designers? If designers do not decide as a group what design rules will be used, then each designer will decide separately in the course of applying guidelines. The result will surely be an inconsistent design.

.2 DESIGN RULES

3.2.1 Types of design rules

Rules based on principles:

These comprise abstract design rules, rules based on low authority, and those based on high generality

Rules based on standards

These are specific design rules from high authority but with limited application

Rules derived from guidelines

These are rules of lower authority but of more general application

3.2.2 DOCUMENTATION, IMPLEMENTATION AND EVALUATION OF DESIGN RULES

After design rules have been specified for each selected guideline, those rules should be documented for reference by software designers and others involved in system development. Documentation of agreed rules, subject to periodic review and revision as necessary, will help coordinate the design process. Documented rules can then be applied consistently for a given application. With appropriate modifications, rules adopted for one application might later be used for other applications.

In the course of design, it may be determined that a particular design rule cannot be used. Therefore, some means must be provided to deal with exceptions. If a design rule is not appropriate for one particular display, then an exception can be made by whoever has been appointed to make such decisions. But if a design rule cannot be implemented at all, perhaps due to other design constraints, then all designers for that particular system must be notified, and perhaps another design rule must be substituted.

Finally, after the design is complete, it must be evaluated against the original design requirements to ensure that all design rules have indeed been followed. To help in the exception process and in the evaluation process, it may be useful to assign different weights to the various rules, indicating which are more important than others. Such weighting will help resolve the trade-offs that are an inevitable part of the design process.

3.2. 3 USING DESIGN RULES

Since design rules suggest how to increase usability, they may differ in generality and authority, therefore there exist various standards that guarantee uniformity of application. Some of those existing standards are:

- The standards set by national or international bodies to ensure compliance by a large community of designers. These standards require sound underlying theory particularly on this slowly changing technology.
- Hardware standards: These are more common than software standards. They are of high authority and low level of detail
- ISO 9241 standards that define usability as the effectiveness, the efficiency and the satisfaction with which users accomplish tasks
- "Broad brush" design rules
- Useful check list for good design
- Better design using these than using nothing!
- Different collections e.g.
  - Nielsen's 10 Heuristics (see Chapter 9)
  - Shneiderman's 8 Golden Rules
  - Norman's 7 Principles

There are Golden rules and heuristics governing designs

These are regarded as "Broad brush" design rules that provide a useful check list for good design. A better design is achieved using these than using nothing!

The different collections include: the Nielsen's 10 Heuristics rules, the Shneiderman's 8 Golden Rules and the Norman's 7 Principles.

3.2.4 The Shneiderman's 8 Golden Design Rules

1. *Strive for consistency*
2. *Enable frequent users to use shortcuts*
3. *Offer informative feedback*
4. *Design dialogs to yield closure*
5. *Offer error prevention and simple error handling*
6. *Permit easy reversal of actions*
7. *Support internal locus of control*
8. *Reduce short-term memory load*

. 3 DESIGN PRINCIPLES

3. 3.1 The Norman's 7 Design Principles

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

3. 3.2 DESIGN PRINCIPLES FORMULATED TO SUPPORT USABILITY :

Principle of Learnability : This is the ease with which new users can begin effective interaction and achieve maximal performance

Principle of Flexibility: These are the multiplicity of ways the user and system exchange information

Principle of Robustness: This is the level of support provided the user in determining successful achievement and assessment of goal-directed behaviour

The Principles of learnability are broken down into :

Predictability : This is determining effect of future actions based on past interaction history and its operation visibility

Synthesizability: This is assessing the effect of past actions, its immediate and its eventual honesty

Familiarity: This is how prior knowledge applies to new system and how easy one can guess its affordance Generalizability: This is extending specific interaction knowledge to new situations

Consistency: This concerns the likeness in input and output behaviour arising from similar situations or task objectives

Principles of flexibility comprise:

Dialogue initiative : This is the freedom from system imposed constraints on input dialogue and it compares the system against the user pre-emptiveness.

Multithreading: This is expressing the ability of the system to support user interaction for more than one task at a time. It also looks at the concurrent and interleaving multimodality.

Task migratability: This is passing responsibility for task execution between user and system

Substitutivity: This allows equivalent values of input and output to be substituted for each other. It compares representation multiplicity and equal opportunity

Customizability: This is the modifiability and adaptability of the user interface by user or the modifiability and adaptivity of the user interface by the system.

Principles of robustness are made up of:

Observability: This is the ability of the user to evaluate the internal state of the system from its perceivable representation. It considers the browsability, the defaults, the reachability, the persistence , and the operation visibility.

Recoverability: This concerns the ability of the user to take corrective action once an error has been recognized. It looks at the reachability, the forward and backward recovery and the commensurate effort.

Responsiveness:This is how the user perceives the rate of communication with the system and how stable is the response.

Task conformance: This explains the degree to which system services support all of the user's tasks, the task completeness and its adequacy.

4.0 Conclusion

The goal of interaction design is to design for maximum usability
Design rules comprise the principles of usability which look at the general understanding of the design, the standards and guidelines which set the direction for design, and the design patterns that capture and reuse design knowledge.
In designing computer-based information systems, special attention must be given to software supporting the user interface.

5.0 Summary

Guidelines are more suggestive and general. There are two types of guidelines; Abstract guidelines and detailed guidelines.
Understanding justification for guidelines aids in resolving conflicts.
Before a guideline can be applied, it must be translated into specific design rules. Those rules should be documented for reference by software designers and others involved in system development.
There are Golden rules and heuristics governing designs
For maximum effectiveness, guideline monitoring must take place early in the design process before any actual design of user interface software.
Guidelines cannot take the place of expertise experience.
There should be early creation of an operational prototype to evaluate interface design concepts interactively with users.
There are certain design principles of learnability, flexibility, and robustness that are formulated to support usability.

6.0 Tutor Marked Assignment

8. Mention those responsible for selecting relevant guidelines for application and translation into design rules. Indicate specific area of responsibility for each professional.
9. Why is it necessary to translate selected guidelines into design rules?
10. What are the advantages derivable from documenting design rules, why is the evaluation of the design necessary?
11. When is it most appropriate to monitor design guidelines and carry out prototype testing and why?
12. Distinguish between rules based on principles, those based on standards and those derived from guidelines.
1 3. What are those design principles formulated to support usability?
14. Mention any 6 of the Shneiderman's 8 Golden rules that govern interactive designs.

7.0 Further Readings / References

- C. Brown: Human-Computer Interface design guidelines. Ablex, 1989.
- W. Galitz: Handbook of screen format design. QED, 1989.
- C. Gram, G. Cockton (eds.): Design principles for interactive software. Capman & Hall, 1996.
- D. Hix, R. Hartson: Developing user interfaces. Wiley, 1993.
- ISO 9241 (Part 10: Dialogue principles, Part 12: Presentation of information, Part 14: Menu dialogues, Part 15: Command dialogues, Part 16: Direct manipulation dialogues, Part 17: Form fill-in dialogues)
- D. Mayhew: Principles and guidelines in software user interface design. Prentice, 1992.

UNIT 2: EVALUATION METHODS

Table of contents

3.0 MAIN CONTENT
- 3.1 Evaluation Techniques
  - 3.1.1 Cognitive Walkthrough
  - 3.1.2 Heuristic Evaluation
  - 3.1. 3 Review-based evaluation
  - 3.1.4 Evaluating through user Participation
- 3.2 Evaluating Implementations
  - 3.2.1 Experimental evaluation
  - 3.2.2 Analysis of data
  - 3.2. 3 Experimental studies on groups
  - 3.2.4 The Data gathering and Analysis processes
- 3. 3 Field studies
  - 3. 3.1 Observational Methods
  - 3. 3.2 Query Techniques
  - 3. 3. 3 Physiological methods

4.0 Conclusion
5.0 Summary
6.0 Tutor Marked Assignment
7.0 Further Reading/References

1.0 INTRODUCTION

Evaluation tests usability and functionality of system and can be carried out in the laboratory, in the field and/or in collaboration with users.
The evaluation technique which covers both design and implementation should be considered at all stages in the design life cycle.
The goals of evaluation are to assess extent of system functionality, to assess effect of interface on user, and to identify specific problems from design and implementation.

3.0 OBJECTIVES

By the end of this unit, you should be able to:

- Express your understanding of the three main evaluation techniques viz a viz the cognitive walkthrough, the heuristic, and the review-based evaluations.
- Master available methods of evaluating interaction design and its implementation
- Select the best evaluation methods among the options available

.0 MAIN CONTENT

3.1 Evaluation Techniques

The Evaluation Design Techniques include:
The Cognitive Walkthrough
The Heuristic Evaluation
The Review-based evaluation
User Participation

3.1.1 Cognitive Walkthrough

This technique was proposed by Polson et al It evaluates design on how well it supports user in learning task and is usually performed by expert in cognitive psychology.

The design expert 'walks through' the design to identify potential problems using psychological principles with forms used to guide the analysis
For each task, the walkthrough considers what impact the interaction will have on the user, the cognitive processes required and the learning problems that may occur.
The analysis focuses on goals and knowledge to establish whether the design leads the user to generate the correct goals.

3.1.2 Heuristic Evaluation
This was proposed by Nielsen and Molich. Here, usability criteria (heuristics) are identified; the designs are examined by experts to see if these are violated.
Examples of heuristics include:
Testing whether the system behaviour is predictable
Testing whether the system behaviour is consistent
Testing whether a feedback is provided
Heuristic evaluation 'debugs' design.

3.1. 3 Review-based evaluation
This evaluation reviews results from the literature that are used to support or refute parts of the design. However, care is needed to ensure the results are transferable to new design.
It is a model-based evaluation which in addition encompasses cognitive models that can be used to filter design options
An example is the GOMS prediction of user performance.
The design rationale can also provide useful evaluation information

3.1.4 Evaluating through user Participation
Evaluation could be carried out in two ways; 1. through laboratory studies and 2. through the field studies

Laboratory studies
Laboratory studies are appropriate if system location is dangerous or impractical for constrained single user systems to allow controlled manipulation of use
The advantages of carrying out laboratory studies are appropriate specialist equipment available and are utilised in an uninterrupted environment.
Disadvantages could be lack of context and difficulty in observing several users cooperating.

Field Studies approach
This approach is appropriate where context is crucial for longitudinal studies
The advantages are that studies are carried out in a natural environment where context of evaluation is retained. Though observation may alter such context. Advantageously, longitudinal studies are also possible.
Obvious disadvantages are that there could be distractions and noise particularly in which the location of the field is within a public place.

3.2 Evaluating Implementations
To evaluate implementations, the evaluator uses artefacts such as simulation, prototypes and the full implementation.

3.2.1 Experimental evaluation
This is a controlled evaluation of specific aspects of interactive behaviour. Here the evaluator chooses the hypothesis to be tested with a number of experimental conditions considered different only in the value of some controlled variable.

The changes in behavioural measure are attributed to the different conditions.

Experimental evaluation factors

The following eexperimental factors are given consideration:
Subjects: This identifies who the representative is, and the measure of the sufficient sample for the experiment.
Variables: These are the things to modify and measure
Hypothesis: This considers what you would like to show
Experimental design: This looks at how you are going to do it

Variables experimental factors

There are two variables; Independent variable (IV) and Dependent variable (DV)
The independent variable characteristics are changed to produce different conditions. Examples of the characteristics include the interface style and number of menu items.
The dependent variable (DV) characteristics are those measured in the experiment. Examples of such characteristics include the time taken and number of errors.

Hypothesis experimental factor

This is a prediction of outcome framed in terms of IV and DV. For example, "error rate will increase as font size decreases".
For example ,if the null hypothesis states that there is no difference between conditions, our aim is to disprove this
e.g. A null hypothesis may be stated that there is "no change with font size". So we disprove.

Experimental design factors

Within groups design: Here each subject performs experiment under each condition.
The advantage here is that transfer of learning is made possible. It is also less costly and less likely to suffer from user variation.
Between groups design: Each subject here performs under only one condition hence there is no transfer of learning. Also more users are required and variation can bias results.

3.2.2 Analysis of data

It is necessary that before you start to do any statistics, you have to look at the data and you have to save the original data.
The choice of statistical technique depends on type of data and the information required.
Type of data : This could either be discrete, that is, comprising finite number of values or continuous, comprising any value.

Analysis - types of test

Parametric test:
This assumes a normal distribution and it is robust and powerful.
Non-parametric test:
This does not assume a normal distribution. It is less powerful but more reliable
Contingency table test:
This classifies data by discrete attributes. It counts number of data items in each group.
The information required is to establish whether there is a difference and how big the difference is. It also seeks to establish how accurate the estimate is.
However, parametric and non-parametric tests are used to mainly establish whether there is a difference.

.2. 3 Experimental studies on groups

These are more difficult than single-user experiments.
These studies identify problems associated with subject groups, choice of task, the data gathering and the analysis of the data gathered.

Some of the problems identified with subject groups are:
The larger the number of subjects the more expensive the experimental design becomes. It also takes a longer time to 'settle down'. It creates an even more variation that makes it difficult to adhere to timetable.
Therefore, only three or four groups are recommended.
The tasks involved in experimental studies on groups are the needs to encourage cooperation among the groups through the use of multiple channels.
The options that may be adopted are:
Creative task such as *writing a short report on a particular experiment*
Decision games such as desert survival task games modelling a decision phenomenon
Control task such as demonstrated in a particular firm.

3.2.4 The Data gathering and Analysis processes
This can be done using several video cameras with direct logging of application data.
The problems with data gathering are synchronisation of data and the sheer volume of data required.
The one solution to this is to record from each perspective.

Analysis of data
Because of the vast variation between groups,
Carry out experiments within groups
Conduct a micro-analysis such as gaps in speech.
Conduct anecdotal and qualitative analysis and look at interactions between group and media.
Realise that controlled experiments may 'waste' resources!

3. 3 Field studies
In field studies, experiments are dominated by group formation but are more realistic because:
There is a distributed cognition with the work studied in context
The real action is a situated action having both the physical and social environment being crucial.
Contrast:
Psychology — controlled experiment
Sociology and anthropology comprises open study and rich data

3. 3.1 Observational Methods
These involve the following:
Think Aloud
Cooperative evaluation
Protocol analysis
Automated analysis
Post-task walkthroughs

Think Aloud method
The user observed performing task as he is asked to describe what he is doing and why, what he thinks is happening etc.
The advantages of this method are :-
It is simple and requires little expertise
It can provide useful insight

It can show how system is actually in use
The disadvantages are:-
It is subjective and selective
The act of describing may alter task performance

Cooperative evaluation method
This is a variation on think aloud. The user collaborates in evaluation such that both the user and the evaluator can ask each
other questions throughout.
Additional advantages here are that:-
It is less constrained and easier to use
The user is encouraged to criticize the system
Clarification is possible between the user and collaborator

Protocol analysis method
This requires paper and pencil; it is therefore cheap and limited to writing speed.
It uses audio that is good for think aloud but difficult to match with other protocols.
It uses video that enables an accurate and realistic analysis but needs special equipment. It is obtrusive.
Other analysis tools involve computer logging that is automatic and unobtrusive in which large amounts of data may be difficult to analyze
It requires a user notebook that is coarse and subjective, providing useful insights and good for longitudinal studies.
However, mixed use of these tools is carried out in practice.
The audio or video transcription is difficult and requires skill. Some automatic support tools are similarly available.

Automated analysis
This is a workplace project involving a post task walkthrough where the user reacts on action after the event. It is used to fill in intention.
Advantages
The analyst has time to focus on relevant incidents
It helps avoid excessive interruption of task
Disadvantages
There is a lack of freshness
There may be post-hoc interpretation of events

Post-task walkthroughs
Here transcript is played back to participant for comment
The advantages are that the response to transcript playback is immediate and is fresh in mind.
The evaluator has time to identify questions and hence useful to identify reasons for actions and alternatives considered
It is mostly necessary in cases where think aloud is not possible.

3. 3.2 Query Techniques
Query technique comprises Interviews and Questionnaires

Interviews
The analyst questions the user on one-to -one basis and is usually based on prepared questions.
The interviews are informal, subjective and relatively cheap to conduct.
The advantages are:

It can be varied to suit context
Issues can be explored more fully
It can elicit user views and identify unanticipated problems
The disadvantages are that it is very subjective and time consuming.

Questionnaires
In this method, set of fixed questions are given to users.
The advantages are that it is quick and reaches large user group.
It can be analyzed more rigorously.
The disadvantages are
It is less flexible and less probing
There is a need for a careful design on what information is required and how answers are to be analyzed.
Styles of question are: general, open-ended, scalar, multi-choice, and ranked.

3. 3. 3 Physiological methods
These comprise Eye tracking and Physiological measurement

Eye tracking
With this method, the head or desk mounted equipment tracks the position of the eye. The eye movement reflects the amount of cognitive processing a display requires.
Measurements include
Fixations: Here, the eye maintains a stable position. The number and duration of measurements indicate level of difficulty with display
Saccades: In this case, there is a rapid eye movement from one point of interest to another.
Scan paths: This involves moving straight to a target with a short fixation at the target being optimal.

Physiological measurements
In physiological measurement, the emotional response is linked to physical changes which may help determine a user's reaction to an interface.
The measurements include:
Heart activity, including blood pressure, volume and pulse.
Activity of sweat glands such as in Galvanic Skin Response (GSR)
Electrical activity in muscle called electromyogram (EMG)
Electrical activity in brain called electroencephalogram (EEG)
However, some difficulties are always experienced in interpreting these physiological responses; therefore, more research is required in this area.

4.0 Conclusion

Guides towards choosing an evaluation method comprise:
Commencement of evaluation process: Design versus Implementation
Style of evaluation: Laboratory versus Field
Nature of evaluation: Subjective versus Objective
Type of measures: Qualitative versus Quantitative
Level of information: High level versus Low level
Level of interference: Obtrusive versus Unobtrusive
Resources available: Time, Subjects, Equipment and Expertise

5.0 Summary

Cognitive Walkthrough evaluates design on how well it supports user in learning task and is usually performed by expert in cognitive psychology.
In heuristic evaluation, usability criteria (heuristics) are identified and the designs are examined by experts to see if these are violated.
Review-based evaluation reviews results from the literature which are used to support or refute parts of design.
User participation evaluation is carried out through laboratory studies and field studies
Experimental evaluation is a controlled evaluation of specific aspects of interactive behaviour by choosing the hypothesis to be
tested with a number of experimental conditions.
Analysis of data is done through parametric test, non-parametric test, and contingency table test of data.
Experimental studies on groups identifies problems associated with subject groups, choice of task, the data gathering and the analysis of the data gathered.
Observational methods involve think aloud, cooperative evaluation, protocol analysis, automated analysis, and post-task walkthroughs
Query technique comprises Interviews and Questionnaires
Physiological methods of evaluation comprise Eye tracking and Physiological measurements.

7.0 Tutor Marked Assignment

12. What is the purpose of carrying out evaluation tests? Enumerate the available techniques used in carrying out the evaluation.
1 3. What is the objective of the "Cognitive walkthrough" and how is it carried out?
14. Provide 3 examples of Heuristics
15. Explain the two ways by which evaluation is carried out through user participation
16. Carrying out evaluation through the laboratory and field studies has some obvious advantages and disadvantages. What are they?
17. Describe the 4 experimental factors to consider while carrying out an experimental evaluation.
18. Describe 3 types of tests that can be carried out for analyzing data
19. "Experimental studies on subject groups are more difficult than single-user experiments". What are the specific problems associated with subjects groups to justify this statement?
20. Describe the Query techniques of evaluating design. What are their advantages and disadvantages?
21. Distinguish between the "Think Aloud" and Cooperative observational methods of evaluating designs?
22. Describe the physiological methods employed to evaluate interactive design.

7.0 Further Readings / References

- J. Dumas, J. Redish: A practical guide to usability testing. Ablex, 199 3.
- D. Freedman, G. Weinberg: Walkthroughs, Inspections, and technical reviews. Dorset, 1990.
- ISO 9241 (Part 11: Guidance on usability, Part 1 3: User guidance)
- Monk, P. Wright, J. Haber, L. Davenport: Improving your Human-Computer Interface: a practical technique. Prentice Hall, 199 3.
- J. Nielsen, R. Mack (ed.): Usability inspection methods. Wiley, 1994.
- D. Norman: The psychology of everyday things. Basic Books, 1988.

UNIT 3: PARTICIPATORY DESIGN

Table of contents

1.0 INTRODUCTION

Participatory design is an approach to design that attempts to actively involve the end users in the design process to help ensure that the product designed meets their needs and is usable.

In participatory design, end-users are invited to cooperate with researchers and developers during a system interaction design process. They participate during several stages of the design process such as in the initial exploration and problem definition both to help define the problem and to focus ideas for solution. During development, they help evaluate proposed solutions.

Participatory design can be seen as a move of end-users into the world of researchers and developers, while a move of researchers and developers into the world of end-users is known as empathic design. This unit looks at both as necessarily participatory design.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Know the meanings of user centered design, distributed participatory design, and ethnography
- Describe the characteristics of Participatory Design(PD)
- Understand Hybridism and the Third Space concept
- Explain the diversity of Participatory Design techniques operating in third spaces
- Appreciate the unresolved issues in Participatory Design

.0 MAIN CONTENT

3.1 Major Concepts of Participatory Design:

User-design versus User-centered design:

There is a very significant differentiation between user-design and User-centered design.

There is an emancipatory theoretical foundation and systems theory bedrock on which user-design is founded.

In user-centered design, users are taken as centers in the design process, consulting with users heavily, but not allowing users to make the decisions, nor empowering users with the tools that the experts use.

For example, most of the internet documentation and information content are user-designed. Users are given the necessary tools to make their own entries. While users are allowed to propose changes or have input on the design, a smaller and more specialized group decide about features and system design.

Ethnography and Participatory design

In participatory design, workers enter into design context while in ethnography (as used for design), the designer enters into work context. Both make workers feel valued in design and encourages workers to 'own' the products. The user is an active member of the design team.

Participatory design in software development

This is the user involvement in design, with more emphasis on the involvement of a broad population of users rather than a small number of user representatives.

Many groups and projects apply participatory design research methods on a regular basis, and, hence, are part of the development and appropriation of the methods, as well as of disseminating the methods to industrial practice.

Distributed participatory design

Distributed Participatory design (DPD) is a design approach and philosophy that supports the direct participation of users and other stakeholders in system interaction analysis and design work. Nowadays design teams most often are distributed, which stress a need for support and knowledge gathered from design of distributed systems. Distributed Participatory design aims to facilitate understanding between different stakeholders in distributed design teams by giving each the opportunity to engage in hands-on activities.

Ethics

The ethics involve a participatory socio-technical approach devised by Mumford.

It states that the system development is about managing change and that non-participants are more likely to be dissatisfied.

There are three levels of participation: consultative, representative, and consensus.

Design groups including stakeholder representatives make design decisions and job satisfaction is the key to solution

3.2 CHARACTERISTICS OF PARTICIPATORY DESIGN

Participatory design is context and work oriented rather than system oriented. It is collaborative and iterative.

Hence the unit focuses on participatory practices that share these attributes, including (a) site-selection of PD work; (b) workshops; (c) story collecting and story telling through text, photography, and drama; (d) games for analysis and design; (e) the correlation of descriptive and functional prototypes and (f) brainstorming, pencil and paper exercises

Participatory design (PD) is a set of theories, practices, and studies related to end users as full participants in activities leading to software and hardware computer products and computer based activities

The field is extraordinarily diverse, drawing on fields such as (a) user-centered design, (b) graphic design, (c) software engineering,
(d) architecture, (e) public policy,(f) psychology, (g) anthropology, (h) sociology, (i) labor studies,(j) communication studies, and (k) political science.

Researchers and practitioners are brought together—but are not necessarily brought into unity—by a pervasive concern for the knowledge, voices, and rights of end users, often within the context of software design and development, or of other institutional settings (e.g., workers in companies, corporations, universities, hospitals, and governments).

This unit primarily addresses methods, techniques, and practices in participatory design, with modest anchoring of those practices in theory with the involvement of software professionals and the end users.

3. 3 HYBRIDITY AND THE THIRD SPACE CONCEPTS OF PARTICIPATORY DESIGN

This is concerned with participatory methods that occur in the hybrid space between software professionals and end users.

Why is this hybrid space important? An influential argument was made that the border or boundary region between two domains, or two spaces, is often a region of overlap or hybridism— that is a "third space" that contains an unpredictable and changing combination of attributes of each of the two bordering spaces. In such a hybrid space, enhanced knowledge exchange is possible, precisely because of those questions, challenges, reinterpretations, and renegotiations.

These dialogues *across differences* and *within differences*—are stronger when engaged in by groups, emphasizing not only a shift from assumptions to reflections, but also from individuals to collectives.

Guides and Expectations of Hybridism or Third Space Concept

There is an overlap between two or more different regions or fields (inbetweenness)

It is not "owned" by any reference field but partaking of selected attributes is done in reference fields.

Potential site of conflicts exist between or among reference fields, hence questioning and challenging of assumptions are unavoidable

Mutual learning and synthesis of new ideas are core benefits when agreed working language among the participants is ensured.

When working assumptions and dynamics are ensured, understandings, relationships and collective actions emerge while dialogues across and within different disciplines exist.

Considerations in the design process:

What is considered to be data are posers

The rules of evidence may become challenges to overcome

How are the conclusions drawn become issues to be resolved

Reduced emphasis on authority and increased emphasis on interpretation are norms

Reduced emphasis on individualism and increased emphasis on Collectivism result in Heterogeneity.

Organizations comprise multiple constituencies each with their own professional identities and views of others. By contrast, the methods

allow for the creation of new perspectives and new locations, and they acknowledge the possibility that each participant can make different choices at different moments about where to locate his or her perspective, standpoint, and thus, accountability.

There is a need for "a new set of skills and competencies that go beyond technical design skills to create conditions that encourage a collaborative design process and active reflection for working with groups. These push on the traditional boundaries between the users and designers"

A large part of the design process, especially in large-scale projects and organizations involving several actors, is not dedicated to analytical work to achieve a solution but mostly to efforts at reconciling conflicting [conceptual] frames or at translating one frame into another. Much work of the designer is concerned with defining collectively what the relevant problem is and how to evaluate such problem.

3.4 PARTICIPATORY DESIGN IN HCI SOFTWARE DEVELOPMENT

Participatory design desires to bridge the gap between two spaces—the world of the software professionals, and the world of the end users. Each world has its own knowledge, practices and well-defined boundaries. Movement from one world to the other is known to be difficult. This difficulty is manifested in our elaborate methods for requirements analysis, design, and evaluation—and in the frequent failures to achieve products and services that meet users' needs and/or are successful in the marketplace.

Traditional scientific practice in HCI has focused on instruments and interventions that can aid in transferring information between the users' world and the software world. Most of the traditional methods are relatively one-directional; for example, we analyze the requirements *from* the users, we deliver a system *to* the users, and we collect usability data *from* the users.
While there are many specific practices for performing these operations, relatively few of them involve two-way discussions, and fewer still afford opportunities for the software professionals to be surprised—*to learn something that we didn't know we needed to know*.
The PD tradition has, from the outset, emphasized mutuality and reciprocity—often in a hybrid space that enabled new relationships and understandings.
"The mutual validation of diverse perspectives": Floyd (1987) analyzed software practices into two paradigms, which she termed product-oriented (focused on the computer artifact as an end in itself) and process-oriented (focused on the human work process, with the computer artifact as means to a human goal). In her advocacy of balancing these two paradigms, Floyd noted that the process-oriented paradigm required mutual learning among users and developers

Most of PD theories and practices require the combination of multiple perspectives—in part, because complex human problems require multiple disciplines (e.g., software expertise and work-domain expertise) for good solutions and in part because the workplace democratic tradition reminds us that all of the interested parties should have a voice in constructing solutions methods suitable for a software professional's organization with *concrete* methods suitable for work with end users.
Muller and colleagues elaborated on this taxonomic dimension by asking *whose work domain serves as the basis for the method*
At the *abstract* end of the continuum, the users have to enter the world of the software professionals in order to participate—e.g., rapid prototyping. At the *concrete* end of the continuum, the software professionals have to enter the world of the users in order to participate; for example, ethnography and end-user "design" by purchasing software for small companies
"What about the practices that did not occur at the *abstract* or *concrete* end-points of the continuum? *What about the practices in between?*" These practices turn out to occur in an uncertain, ambiguous, overlapping disciplinary domain that does not "belong" to either the software professionals or the end users (e.g., these practices occur in neither the users' turf nor the software professionals' turf).
The practices in between the extremes are hybrid practices, and constitute the third space of participatory design.

3.5 NEGOTIATION, SHARED CONSTRUCTION, AND COLLECTIVE DISCOVERY IN PD AND HCI

This describes a diversity of participatory design techniques, methods, and practices that provide hybrid experiences or that operate in intermediate third spaces in HCI. Because the theme is hybridism, these descriptions are organized in terms, strategies, and moves that introduce novelty, ambiguity, and renewed awareness of possibilities, occurring at the margins of existing fields or disciplines.
A storytelling method provides a space in which people negotiates the naming and defining of workplace activities

3.5.1 Site Selection

One of the simplest parameters that can be manipulated to influence hybridism is the site of the work. There are two approaches to participatory design: (1) Bring the designers to the workplace or (2) Bring the workers to the design room at a site different from the work place.
The selection of the site can be important in a discussion of participatory architectural practice,

Work place site selection

Being in a foreign environment and with other users, users will tend to take a more general view of things; however, when collaborating with users *in their work context*, users tend to feel more at ease as they are on their home ground—the designers are the visitors. Tools and environment are physically present and easy to refer to. This makes for a conversation grounded in concrete and specific work experiences. The idea was born to create a type of design event with activities in both environments and with two sets of resources to support design collaboration.

New site selection

In terms of hybridism, the selection of site can be a deliberate strategy to introduce new experiences and perspectives to one or more parties in the design process—a de-centering move that can bring people into positions of ambiguity, renegotiation of assumptions, and increased exposure to heterogeneity.
Site selection initially appears to be a matter of *moving across the boundary* between different work cultures, rather than *living within the boundary*. The use of *common design practices across sites*, however, makes those practices (and the membership of the design group) into a kind of movable third space. The practices and the group membership become stable features that persist across multiple sites. At the same time, the practices, and even the membership grow and evolve with exposure to new sites and new understandings. In these ways, the practices become an evolutionary embodiment of the knowledge of the learning of the group

Benefits of using new site:

- Improved learning and understanding.

It is a move from"symmetry of ignorance" toward "symmetry of knowledge" as diverse parties educate one another through a "symmetry of learning"—and even a kind of "transformation" through exposure to new ideas.
The selection of site can also lead to the strengthening of the voices that were comfortable at each site.

- Greater ownership. The procedures could strengthen the user involvement in their projects. There would also be increases in commitment and ownership of the evolving knowledge and design of the group.

3.5.2 Workshops

Workshops may serve as another alternative to other site selection. Workshops are usually held to help diverse parties ("interested parties" or "stakeholders") communicate and commit to shared goals, strategies, and outcomes (e.g., analyses, designs, and evaluations, as well as workplace-change objectives).
Workshops are often held at sites that are in a sense neutral—they are not part of the software professionals' workplace, and they are not part of the workers' workplace.
More importantly, workshops usually introduce novel procedures that are not part of conventional working practices. These novel procedures take people outside of their familiar knowledge and activities, and must be negotiated and collectively defined by the participants. Workshops are thus a kind of hybrid or third space, in which diverse parties communicate in a mutuality of unfamiliarity, and must create shared knowledge and even the procedures for developing those shared knowledge.
The best-known workshop format in PD is the Future Workshop

A Future Workshop proceeds through three stages: (a) *Critiquing* the present, (b) *Envisioning* the future, and (c) *Implementing*, or moving from the present to the future. These three activities involve participants in new perspectives on their work, and help to develop new concepts and new initiatives.
A number of workshops have focused on simple materials and informal diagrams, rather than on formal notations. The tools are simple diagrams or drawings with no special formalisms because staff members

participating in the workshops, as well as those to whom the results are later presented, typically have no experience.
Using technical descriptions, a workshop is described as a family of "generative tools" of activities that are selectively combined into strategic design, under an overall conceptual strategy that combines market research ("what people say"), ethnography ("what people do"), and participatory design ("what people make").
The activities include the construction of collages focused on thinking (e.g., "how do you expect your work to change in the future?"), mapping (e.g., laying out an envisioned work area on paper), feeling ("use pictures and words to show a health-related experience in your past"), and story telling.

A type of storyboarding workshop format is described as that in which people create narratives using photographs, putting them in sequences and in many cases altering (typically through the addition of speech bubbles to show what people were thinking or doing).
The various workshop approaches have several commonalities. Each workshop brings together diverse participants to do common work, to produce common outcomes and to develop a plan of joint action They are thus opportunities that require mutual education, negotiation, creation of understanding, and development of shared commitments. Each workshop takes place in an atmosphere and often in a site that is not "native" to any of the participants. Thus, all of the participants are at a disadvantage of being outside of their own familiar settings, and they must work together to define their new circumstances and relationships. The combination of diverse voices leads to syntheses of perspectives and knowledge.
Benefits. Advantages claimed for this type of hybridism include:

- Development of new concepts that have direct, practical value for product design
- Engagement of the interested parties ("stakeholders") in the process and outcome of the workshop.
- Combinations of different people's ideas into unified concepts.
- Production of artifacts that are the expected and useful "inputs" to the next stage of the development process

.5. 3 NARRATIVE STRUCTURES

Stories and Story telling

Stories and story telling have played a major role in ethnographic work since before there was a field called "HCI". . Stories have also had an important history in HCI.
Stories in participatory work may function in at least three ways. First, they may be used as triggers for conversation, analysis, or feedback. Second, they may be told by end users as part of their contribution to the knowledge required for understanding product or service, opportunities, and for specifying what products or services should do. Third, they may be used by design teams to present their concept of what a designed service or product will do, how it will be used, and what changes will occur as a result Hypermedia technologies can be utilized to enable communities tell their own stories with the intention that "plurality, dissent, and moral space can be preserved. It enables multiple authors reuse community materials selectively, telling different stories within a common context.
The different accounts were organized according to themes, and laid out spatially on the image of a fictitious island for navigation by end users.
The work enters several areas or aspects of hybridism. First, the authors of the stories (e.g., community members) were using hypermedia technology for the first time, and were thus in the role of learners, even while they were the owners of the stories, and were thus in the role of experts. Second, the authors wrote from their own perspectives, which were sometimes in strong conflict with one another. Third, the authors could make use of one another's materials, effectively moving away from single-author narratives and into a kind of collaborative collage of materials, which conveyed interlinked stories.

Fourth, just as the community members were negotiating and defining their roles as learner-experts, the software professionals/researchers were negotiating and defining their roles as expert's facilitators-students.

Using Paper and Pencil to tell stories

A second line of practice and research has emphasized end users telling their stories using a system of paper-and-pencil, card-like templates. The earliest version was the Collaborative Analysis of Requirements and Design (CARD) technique later developed into a more general tool

The card-based practices used pieces of cardboard about the size of playing cards. Each card represents a component of the user's work or life activities, including user interface events (e.g., screen shots), social events (conversations, meetings) and cognitive, motivational, and affective events (e.g., the application of skill, the formation of goals or strategies, surprises and breakdowns, evaluations of work practices). The cards were used by diverse teams in analysis, design, and evaluation of work and technology. Because the cards were novel object to all the participants, they occasioned third-space questionings and negotiations, resulting in new shared understandings and co-constructions. Often, teams used the cards to prepare a kind of story board, narrating the flow of work and technology used and annotating or innovating cards to describe that work. The resulting posters formed narratives of the work that were demonstrated to be understandable to end users, corporate officers, and software professionals, and which led to insights and decisions of large commercial value

Using Photographs for story telling

Stories can be told in many ways. One approach that has informed recent PD work is end-user photography through (a) taking pictures and (b) organizing pictures into albums. These activities allow end users to enter into a kind of native ethnography, documenting their own lives. In keeping with the issues raised in the preceding "Stories" section, it is important that the informants themselves (the end users) control both the camera and the selection of images . They thus become both authors and subjects of photographic accounts of their activities. This dual role leads to one kind of hybridity, in which the photographic activities partake of both the world of common social life, and the world of documenting and reporting on working conditions.

Photo documentaries were used as a means of providing familiar, concrete artifacts to enable design collaborations. Photo documentaries are used as one component of a set of user-composed diary techniques, with a subsequent user created collages to serve as a rich source of discussions.

End-user photography is an interesting case of hybridity and the production of third spaces. Photography is a good example of an "in-between" medium—one that is part of many people's informal lives but that is also an intensively studied medium of communication and argumentation. Photography occurs at the margin of most people's work, and yet can easily be incorporated into their work.

Discussions around the photographs, and combination of the photographs into photo narratives or collages can lead to mutual learning and new ideas, particularly through the inclusion of the voices of the photographers, the viewers, and especially the people depicted in the photographs

Benefits. The use of end-user photographs appears to be new and experimental, and there are few strongly supported claims of benefits. Informal claims of success and contribution include the following:

- Richer, contextualized communication medium between end users and designers. (In some cases, the designers were not, themselves, software professionals.)
- Stronger engagement of designers with end-users' worlds.
- Enhanced sharing of views and needs among end users, leading to stronger articulation by them as a collective voice.

The informants should make their own decisions about what was important, and therefore what they should photograph.

Dramas and Videos

Drama provides another way to tell stories—in the form of theatre or of video. One of the important tensions with regard to drama in PD is the question of whether the drama is considered a finished piece, or a changeable work-in-progress.

Many PD drama-practitioners make reference to Boal's Theatre of the Oppressed. Boal described theatrical techniques whose purpose was explicitly to help a group or a community find its voice(s) and articulate its position(s).

The most influential of Boal's ideas was his Forum Theatre, in which a group of nonprofessional actors performs a skit in front of an audience of interested parties. The outcome of the skit is consistent with current events and trends—often to the dissatisfaction of the audience. The audience is then invited to become authors and directors of the drama, changing it until they approve of the outcome.

Changes in work patterns and work-group relations were acted out by software professionals in the end-users' workplace, using cardboard and plywood prototypes, in anticipation of new technologies, the workers served as the audience, and critiqued the envisioned work activities and working arrangements. The drama was carried out iteratively, with changes, until it was more supportive of the skilled work of the people in the affected job titles. The researchers made repeated visits with more detailed prototypes, again using the vehicle of a changeable drama, to continue the design dialogue with the workers. This work was widely credited with protecting skilled work from inappropriate automation, and leading to a product that increased productivity while taking full advantage of workers' skills.

Muller et al. (1994) presented a related tutorial demonstration piece called Interface Theatre, with the stated goal of engaging a very large number of interested parties in a review of requirements and designs (e.g., in an auditorium). In Interface Theatre, software professionals acted out a user interface "look and feel" using a theatrical stage as the screen, with each actor playing the role of a concrete interface component.

Dramatic approach brings a strong overlap of the world of end users and the world of software developers, showing concrete projections of ideas from one world into the other world—and, in most uses, allowing modification of those ideas. Drama is marginal to the work domains of most software professionals and most end users, and thus moves all parties into an ambiguous area where they must negotiate meaning and collaboratively construct their understandings. Agreements, conflicts, and new ideas can emerge as their multiple voices and perspectives are articulated through this rich communication medium.

Benefits

· Building bridges between the worlds of software professionals and users.

· Enhancing communication through the use of embodied (e.g., acted-out) experience and through contextualized narratives.

· Engaging small and large audiences through direct or actor-mediated participation in shaping the drama (influencing the usage and design of the technology).

· Increasing designers' empathy for users and their work.

· Simulating use of not-yet-developed tools and technologies to explore new possibilities.

· Fuller understanding by focus group members, leading to a more informed discussion.

3.6 GAMES

From theory to practice, the concept of games has had an important influence in participatory methods and techniques.

Ehn's theoretical work emphasized the negotiation of language games in the course of bringing diverse perspectives together in participatory design. In this view, part of the work of a heterogeneous group is to understand how to communicate with one another.

The work of heterogeneous teams is, in part, the "mutual validation of diverse perspectives"

Games have been an important concept in designing practices, with the convergent strategies of enhanced teamwork and democratic work practices within the team.

When properly chosen, games can serve as levelers, in at least two ways. First, games are generally outside of most workers' jobs and tasks. They are therefore less likely to appear to be "owned" by one worker, at the expense of the alienation of the non-owners. Second games are likely to be novel to most or all of the participants. Design group members are more likely to learn games at the same rate, without large differences in learning due to rank, authority, or background. This in turn can lead to greater sharing of ideas. In addition, games can help groups of people to cohere together and communicate better.

One of the purposes of games is enjoyment of self and others—and this can both liven a project and build commitment among project personnel.

"Design-by-playing" approach, introducing several games into PD practice: Examples include:

· Specification Game, a scenario-based game based on a set of "situation cards," each of which described a workplace situation.

Players (members of the heterogeneous analysis/design team) took turns drawing a card and leading the discussion of the work situation described on the card.

· Organization Kit and Desktop Publishing Game, in which cards illustrating components of work or outcomes of work were placed on posters, with annotations.

· CARD, a card game for laying out and/or critiquing an existing or proposed work/activity flow

· PICTIVE, a paper-and-pencil game for detailed screen design

· Icon Design Game, a guessing game for innovating new ideas for icons (this game assumes subsequent refinement by a graphic designer).

· Interface Theatre, for design reviews with very large groups of interested parties

The games emphasize hands-on, highly conversational approaches to discussing both the user interface concept itself and the work processes that it was intended to support..

The Technology Game adds simple shapes that stand for technologies, again playing those shapes onto the work environment in the Landscape Game.

Finally, the Scenario Game moves back to the real world, enacting possibilities based on new ideas from the preceding three games. The enactments may be video recording, both for documentary purposes and to generate further video material for another cycle of the four games.

Each of these games would take all of its players outside of their familiar disciplines and familiar working practices, but strategically reduced the anxiety and uncertainty of the situation by using the social scaffolding of games. Each game requires its players to work together through mutual learning to understand and define the contents of the game, and to interpret those contents to one another in terms of multiple perspectives and disciplines. The conventional authority of the software professionals was thus replaced with a shared interpretation based on contributions from multiple disciplines and perspectives.

Benefits. Participatory design work with games has been claimed to lead to the following benefits:

· Enhanced communication through the combination of diverse perspectives.

· Enhanced teamwork through shared enjoyment of working in a game-like setting.

· Greater freedom to experiment and explore new ideas through flexible rules and redefinition of rules during the game.

· Improved articulation of the perspectives, knowledge, and requirements of workers.

· New insights leading to important new analyses and designs with documented commercial value.

3.7 CONSTRUCTIONS

Preceding sections have considered hybridism in participatory activities, such as site selections, workshops, stories, photography, dramas, and games. This section continues the survey of participatory practices that bring users and software professionals into unfamiliar and ambiguous "third space" settings.

Collaborative construction of various concrete artifacts comprising:

· Low-tech prototypes for analysis and design.

・ Cooperative Prototyping

Low-Tech Prototypes that includes participatory prototyping:
Low-tech prototypes may lead to "third space" experiences because they bring people into new relationships with technologies—relationships that are "new" in at least two important ways. First, the end users are often being asked to think about technologies or applications that they have not previously experienced. Second, in *participatory* work with low-tech prototypes, end users are being asked to use the low-tech materials to reshape the technologies—a "design-by-doing" approach
In this way, participatory work with lowtech prototypes involves much more user contribution and user initiative than the more conventional use of "paper prototypes" as surrogates for working systems in usability testing
The UTOPIA project provided impressive demonstrations of the power of low-tech cardboard and plywood prototypes to help a diverse group to think about new technologies, office layouts, and new working relations that might result from them.

Benefits. The low-tech participatory prototyping approaches benefits include:
・ Enhanced communication and understanding through grounding discussions in concrete artifacts.
・ Enhanced incorporation of new and emergent ideas through the ability of participants to express their ideas directly
via the low-tech materials, and through the construction of artifacts that can be used in other techniques, especially drama and video documentaries.
・ Enhanced working relations through a sense of shared ownership of the resulting design.
・ Practical application with measured successes in using low-tech design approaches to real product challenges, achieving consequential business goals.

Cooperative Prototyping
This last section on participatory methods is concerned with software prototyping.
The potential of cooperative prototyping in several projects, using different technology infrastructures led to enhanced communication
with end users, improved incorporation of end-user insights into the prototypes, and stronger collective ownership and collective action planning by the team. Also observed is the time consuming breakdowns in the design process itself, when new ideas required significant programming effort.
In a different prototyping approach, a system is delivered to its end users as series of iterative prototypes, each of which gradually
adds functionality
What appears to be critical is that the prototype functions as a *crucial artifact* in the end-users' work, such as,

(a) a resource of documents for librarians
(b) an online event checklist that served as the crucial coordination point for the work of diverse contributions or (c) a database supporting funding work in a nonprofit organization . Trigg (2000) provided a series of observations and tactical recommendations about how to engage the users in the evaluations that both they and the software professionals had agreed were needed.

This very brief survey of cooperative prototyping and "iterative delivery" approaches shows several aspects of hybridity. In the case of cooperative prototyping, the cooperative work may be done in a physical third space that is neither the end-users' office nor the software developers' office
In the case of the delivery of iterated prototypes, each prototype is presented in the end users' setting, but is unusual and only partially functional, and thus occasions reflection about its nature, its role in the end users' work, and, ultimately, the work itself. In both cases, the invitation (or perhaps the necessity) of the end-users' actions to help shape the technology becomes an important means of refocusing

their attention, as well as the attention of the software developers. The ensuing conversations are concerned with the interlinked feasibility of changes to technology and to work practices, with attributes of hybridity including polyvocal dialogues, challenging one another's assumptions, and developing plans for collective actions.

Benefits. Some of the virtues of the low-tech prototyping approaches have also been claimed for the cooperative prototyping and "iterative delivery" approaches as follow:

- Enhanced communication and understanding through grounding discussions in concrete artifacts.
- Enhanced working relations through a sense of shared ownership of the resulting design.

Additional claims for software-based prototypes include:

- Earlier understanding of constraints posed by the practical limitations of software.
- Improved contextual grounding of the design in the end-users' work practices.

3.8 Brainstorming

The most well-known idea generation technique is brainstorming, introduced by Osborn (1957). His goal was to create synergy within the members of a group: ideas suggested by one participant would spark ideas in other participants. Subsequent studies challenged the effectiveness of group brainstorming, finding that aggregates of individuals could produce the same number of ideas as groups. They found certain effects, such as production blocking, free riding, and evaluation apprehension, were sufficient to outweigh the benefits of synergy in brainstorming groups.

Brainstorming, is an important group-building exercise for participatory design; designers may brainstorm ideas by themselves.

Brainstorming in a group is more enjoyable and, if it is a recurring part of the design process, plays an important role in helping group members share and develop ideas together.

The simplest form of brainstorming involves a small group of people. The goal is to generate as many ideas as possible on a pre-specified topic: quantity, not quality, is important.

Brainstorming sessions have two phases: the first for generating ideas and the second for reflecting upon them. The initial phase should last no more than an hour. One person should moderate the session, keeping time, ensuring that everyone participates, and preventing people from critiquing each other's ideas. Discussion should be limited to clarifying the meaning of a particular idea.

A second person records every idea, usually on a flipchart or transparency on an overhead projector.

After a short break, participants are asked to reread all the ideas and each person marks their three favorite ideas.

One variation is designed to ensure that everyone contributes, not just those who are verbally dominant. Participants write their ideas on individual cards or Post-it notes for a prespecified period. The moderator then reads each idea aloud.

Authors are encouraged to elaborate (but not justify) their ideas, which are then posted on a whiteboard or flipchart.

Group members may continue to generate new ideas, inspired by the others they hear.

Another variant of brainstorming, called "video brainstorming" is a very fast technique for prototyping interaction: instead of simply writing or drawing their ideas, participants act them out in front of a video camera. The goal is the same as other brainstorming exercises, i.e., to create as many new ideas as possible, without critiquing them. However, the use of video, combined with paper or cardboard mock ups, encourages participants to experience the details of the interaction and to understand each idea from the perspective of the user, while preserving a tangible record of the idea.

Each video brainstorming idea should take two to five minutes to generate and capture, allowing participants to simulate a wide variety of ideas very quickly. The resulting video clips provide illustrations of each idea that are easier to understand and remember than hand-written notes.

Video brainstorming requires thinking more deeply about each idea than in traditional oral brainstorming. It is possible to stay vague and general when describing an interaction in words or even with a sketch, but acting out the interaction in front of the camera forces the author of the idea and the other participants to consider seriously the details of how a real user would actually interact with the idea. Video brainstorming also encourages designers and users to think about new ideas in the context in which they will be used. Video clips from a video brainstorming session, even though rough, are much easier for the design team to interpret than written ideas from a standard brainstorming session.
Unlike standard brainstorming, video brainstorming encourages even the quietest team members to participate.

.9 UNRESOLVED ISSUES IN PARTICIPATORY DESIGN:

· Participation by non-organized workforce. The field of PD has long been concerned about how to engage in meaningful participative activities with workers or others who are not organized into a group with collective bargaining power or other collective representation.

· Evaluation and metrics. One of the weaknesses of the literature on participatory practices is the dearth of formal evaluations.
There is a small set of papers that have examined software engineering projects across companies, and have found positive outcomes related to end-user participation
There are no formal experiments comparing participatory methods with non-participatory methods in a credible workplace context. Such studies would be difficult to perform, because they would require that a product be implemented and marketed twice (once with participation, and once without).
The problem is made more difficult because measurements and metrics of organizational outcomes, user participation, and user satisfaction are currently vexing research issues

4.0 CONCLUSION
Participatory design (PD) is a set of theories, practices, and studies related to end users as full participants in activities leading to software and hardware computer products and computer based activities
Hybridism is at the heart of PD, fostering the critical discussions and reflections necessary to challenge assumptions and to create new knowledge, working practices, and technologies. When we consider HCI as a set of disciplines that lie between the space of work and the space of software development, we see that the hybrid third spaces developed within PD have much to offer HCI in general.

5.0 SUMMARY
In user-centered design, users are taken as centers in the design process, consulting with users heavily. In participatory design, workers enter into design context while in ethnography; the designer enters into work context. Both make workers feel valued in design and encourages workers to 'own' the products.
Participatory design in software development is the user involvement in design,
Distributed Participatory design (DPD) is a design approach and philosophy that supports the direct participation of users and other stakeholders in system interaction analysis and design work.
The ethics involved in the participatory socio-technical approach devised by Mumford, states that the system development is about managing change and that non-participants are more likely to be dissatisfied. There are three levels of participation: consultative, representative, and consensus.
Design groups including stakeholder representatives make design decisions and job satisfaction is the key to solution

6.0 Tutor Marked Assignment
1. Differentiate between the following pairs of terms

User design and User Centered design
Ethnography and Participatory design
2(a) What are the benefits derivable from a distributed participatory design exercise?
(b) Briefly describe the characteristics of the Participatory design
3. What do you understand as the "Third Space Concept" and "Hybridism" in participatory design?
4. In participatory design exercise, the designers are either brought to the workplace or the workers are brought to the design room at a different site from the work place. Briefly itemize the various benefits accruing from selecting any of the options.
5. Describe the workshops of participatory design as an alternative to other site selection. What are the obvious benefits of this alternative arrangement on participatory design?
6. What are the advantages of stories and story telling in participatory design?

7.0 Further Readings / References

- Grudin, J. (1993). Obstacles to Participatory Design in Large Product Development Organizations: Schuler, D. & Namioka, A. (1993). Participatory design: Principles and practices. Hillsdale, NJ: Erlbaum.
- Kensing, F. 2003. Methods and Practices in Participatory Design. ITU Press, Copenhagen, Denmark.
- Kensing, F. & Blomberg, J. 1998. Participatory Design: Issues and Concerns In Computer Supported Cooperative Work, Vol. 7, pp. 167-185.

UNIT 4: SYSTEM INTERACTIVE DESIGN PATTERNS

Table of contents

4.0 Conclusion
5.0 Summary
6.0 Tutor Marked Assignment
7.0 Further Reading/References

1.0 INTRODUCTION

A pattern is an invariant solution to a recurrent problem within a specific context.
An HCI design pattern is an approach to reusing knowledge about successful design solutions
Patterns do not exist in isolation but are linked to other patterns in *languages* which enable complete designs to be generated

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Know the guides for developing effective design patterns
- Understand the processes of design
- Describe various screen designs and layout
- Know how to design acceptable presentation
- Understand Prototyping

.0 MAIN CONTENT

.1 CHARACTERISTICS OF PATTERNS

The characteristics of patterns include the following:

- Capturing the design practice and not the theory
- Capturing the essential common properties of good examples of design
- Representing design knowledge at varying levels of social, organisational, and conceptual framework
- Embodying values and expressing what is humane in interface design
- Patterns are intuitive and readable and can therefore be used for communication between all stakeholders
- A pattern language should be generative and assist in the development of complete designs.

.2 GUIDES AT DEVELOPING EFFECTIVE DESIGN PATTERNS

3.2.1 Commencement of design process:
The human interaction designer would commence his design process by asking the following questions:
The design:

- What is the design all about?
- What are the interventions?
- What are the goals?
- What are the constraints?

The design process

- What happens when?

The Users

- Who are the users?
- What are their likes and dislikes on interactivity?

Navigation of Interaction

- How does the user find his way around a system?

.2.2 Design Considerations:

Scenarios of interaction
Part of the scenarios is a probe of rich stories relating to design issues that include users' experiences and expectations.

Iteration and prototypes
Remember that the designer never get it right the first time!

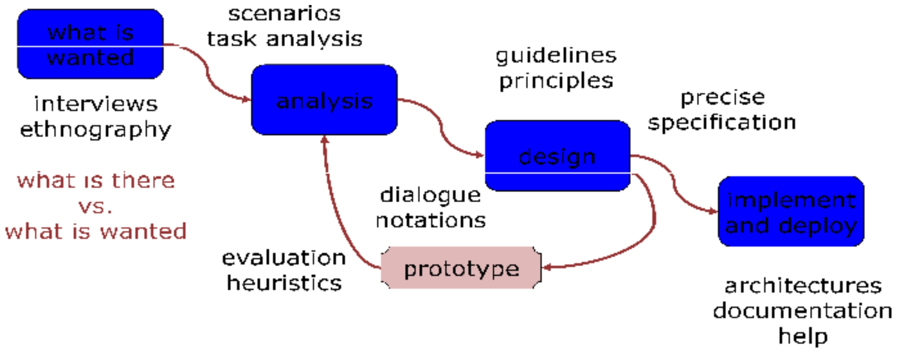
Interactions and Interventions
The designer should design interactions not just interfaces and not just the immediate interaction because technology changes
The designer should design interventions not just artefacts and not just the system, but also related documentation such as manuals and tutorials.

What is design?
Design is achieving goals within constraints, so the design should consider those to benefit from the goals and for what purpose.
The design should consider the constraints in terms of materials and platforms and the corresponding trade-offs.
The Golden rule of design is for the designer to understand his materials for Human—Computer Interaction
Understanding materials means understanding computer's limitations, its capacities, its tools and platforms.
It also means understanding people, their psychological and social aspects.
The design should consider the possibility of human error and their interaction, since to err is human

'To err is human' example:
Example of accident reports on air crash, industrial accident, hospital mistake, reveal the enquiry blaming 'human error' on all. But concrete lintel breaks because of too much weight applied on it so the blame goes to 'lintel error' not design error since we know how concrete behaves under stress.
So human 'error' is normal and we know how users behave under stress hence design for it!
Treat the user at least as well as physical materials! The Central focus is the user.

. 3 THE DESIGN PROCESSES

. 3.1 The design life cycle



Explanation of processes in the diagr am:
Requirements: This is identifying wh t is there and what is wanted
Analysis: This is on ordering and und erstanding
Design: This concerns what to do an how to decide on what to do
Interaction and prototyping: Means etting it right and finding what is really nee ed!
Implementation and deployment: M aking it happen and delivering

To carry out the above, consider The limited time available as a desig The trade-off usability: Find out problems that may work against the ultimate usage of the designed interaction and ensure such problem(s) are tackled r ight from the onset.
Remember that a perfect system is o ne that was badly designed; so do not expect to design a perfect system. Systems are dynamic.

3. 3.2 User focuses
It is essential that you know your use rs, their personality and cultural probes.
Talk to those class of users, watch th em and use your imagination of their percep tion of the system they want.
Innovate a 'user' model not necessar ily a real person but carry out a prototype o a system.

Design's cultural probes
Cultural probes can be carried out as follows:

- By direct observation though sometimes hard; for example on psychiatric patients
- By giving out probe packs that consist of items to prompt responses. These are given to people to open in their own environment and to record what is meaningful *to them*. These probe packs are used to inform interviews, prompt ideas and en-culture designers
- Gathered stories for the desi n are used and reused
- By communicating with others to validate models and understand dynam ics
- Finding out what will users want to do by step-by-step walkthrough on wat they can see using sketches, screen shots etc or what they are used to doing e.g manipulating keyboard and mouse, etc.
- Find out their thinking on the proposed interaction design
- Explore the depths by explori ng interaction to determine what happens w hen
- Explore cognition to determi e the users thinking
- Explore architecture of the sy stem to determine what is happening inside
- Use particular scenarios to co mmunicate with other designers, clients and users
- Validate other models by com paring them with your models
- Express dynamics through scr eenshots appearances and scenario behavio urs.

- Use several scenarios and use several methods since scenarios provide one linear path through system design,

An example of a personality probe for a design.
Sola is 37 years old, She has been the Warehouse Manager for five years and worked for an Engineering company for twelve years. She didn't go to the university, but has studied in her evenings for a business diploma. She has two children aged 15 and 7 and does not like to work late. She did part of an introductory in-house computer course some years ago, but it was interrupted when she was promoted and could no longer afford to take the time. Her vision is perfect, but her right-hand movement is slightly restricted following an industrial accident 3 years ago. She is enthusiastic about her work and is happy to delegate responsibility and take suggestions from her staff. However, she does feel threatened by the introduction of yet another new computer system (the third in her time at the Engineering company).

3. 3. 3 Navigation design
Within the local structure, utilise a single screen
Within the global structure, utilise a whole site

Levels of design to guide the designer include:-

- Widget choice level containing menus, buttons etc.
- Screen design level
- Application navigation design
- Environment design level that comprises other applications and operating systems.

Example of a web design:
A web interaction design comprises:

- The widget choice level containing elements and tags e.g <a href = li....." >
- The screen design such as page design
- The application navigation design such as site structure
- The environment design such as the web, the browser, and external links

The physical devices interaction design comprises:

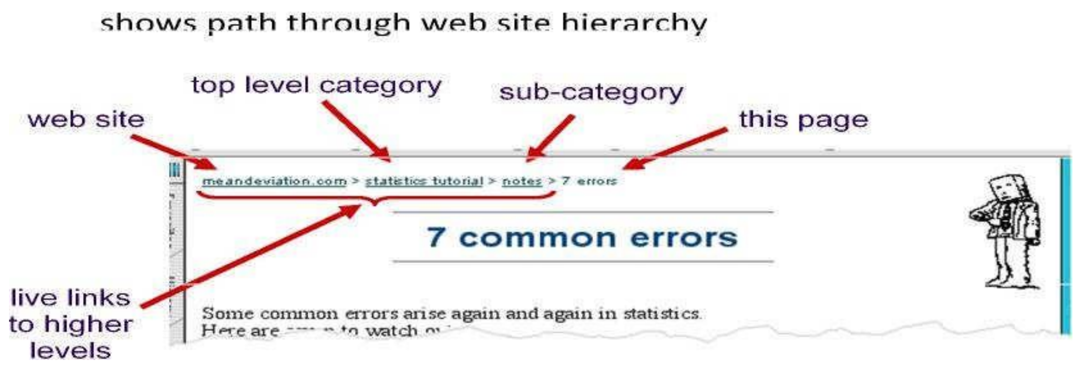
- The widget choice level comprising the controls such as buttons, knobs and dials
- The screen design such as the physical layout
- The application navigation design such as the modes of device
- The environment design such as the real world

Structure of design should be viewed from the following platforms:

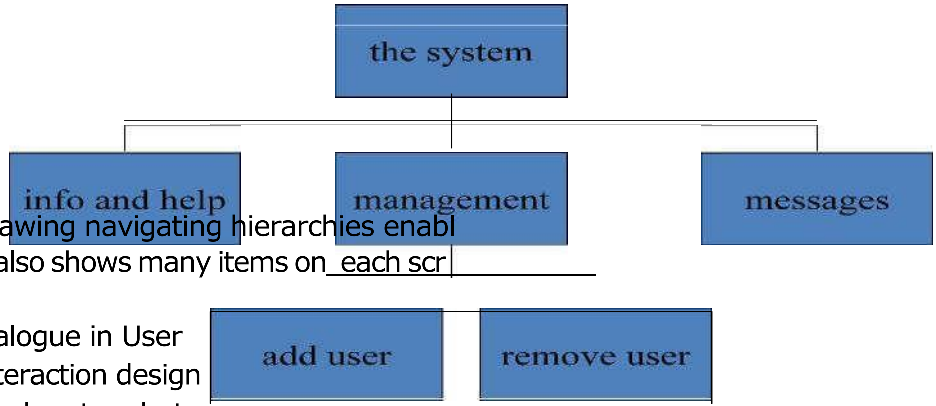
- Within a screen
- Locally looking from one screen looking out
- Globally from the structure of the site and movement between screens within application
- And wider still, consider relationship with other applications

The four golden rules of the design are;
Knowing where you are
Knowing what you can do
Knowing where you are going or what will happen
Knowing where you have been or what you have done

Example of where you are of a Web Site address:



The following Hierarchical diagram s hows parts of application with screens or gro ups of screens typically showing a
Functional separation.



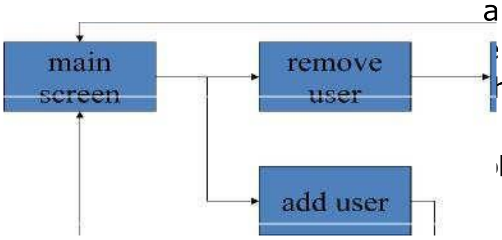
Drawing navigating hierarchies enabl
It also shows many items on each scr

Dialogue in User
Interaction design
i and system but
details differ each ti Network
diagrams show different pa than the es short term memory but not the menu siz e.
hierarchy. een with the items structured within the scr een
It shows the relationship between ap

a computer dialogue having patterns of int raction between users

hs through system including branches that re more task oriented

lications and beyond. See illustration belo



Network diagram shows what leads t
more task oriented than a hierarchic
When considering style issues, it ide

o what, what happens when, it shows functi onal branches and is
al diagram.
tify platform standards and consistency

Network diagrams also identify (i) fu nctional issues such as cut and paste and (ii) navigation issues such as embedded applications and links to ther applications such as the web browsing.

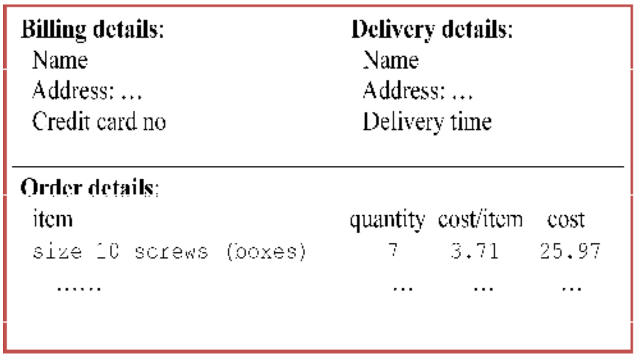
.4 SCREEN DESIGNS AND LAYOU T:

3.4.1 Principles of design

The basic principles in the screen des ign and layout are the grouping of data thatthe screen displays, the a structure, order and their alignment. Use of white space to separate group of dat is also recommended as shown in the diagrams below.

The designer should ask what the us r is doing, he should think of what informat ion should be displayed, the order of display and possible com parisons. The design should ensure that for m follows function on display.

.4.2 Grouping and Structure desig n



From the diagram above, one can distinguish items that are logically together as against those physically together.

The order of groups and items is important and should be considered in the desi n.

Think about what is a natural order;the order should match the screen order!

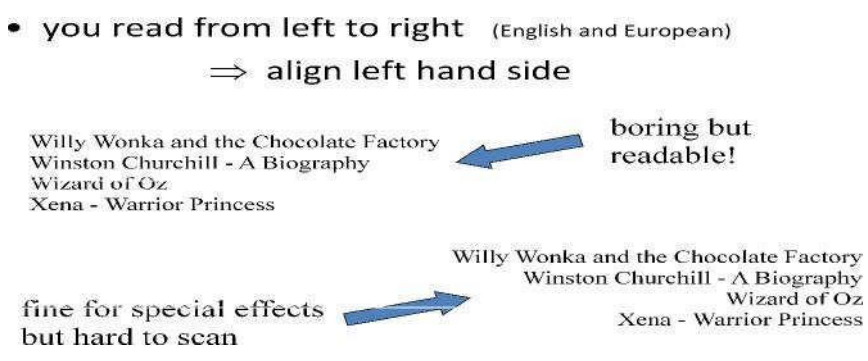
Use boxes, space etc and set up the ight tabbing.

Decoration: Using boxes to group logical items

Use fonts for emphasis and headingsbut these should not be too many!!

Consider the best method to align items; for example, you can separate items usi ng white spaces as in the example above.

.4. 3 Alignment of text



⇒ align left h nd side
Alignment of names

From the three boxes illustrated oes not enable easy
above, the alignment of names in the mes from the first name. The second and thi rd boxes enable easy
first box d recognition or identification box has its names separated by white spac es and the third by
of surna recognition of surnames. The
secon commas.

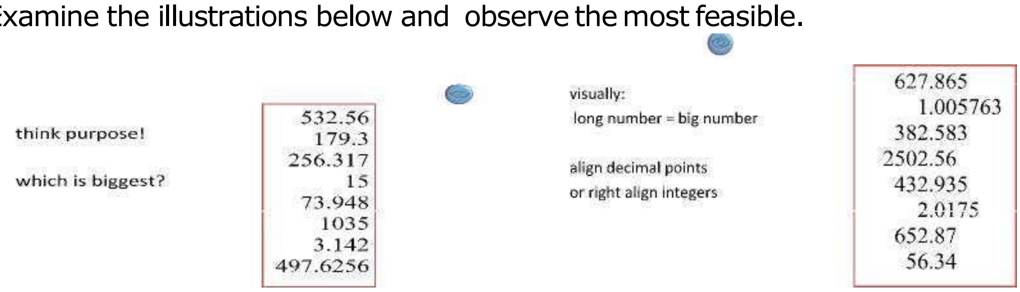
Alignment of numbers

While aligning numbers, think of the purpose such alignment would serve. Consi er the biggest and the smallest numbers.

Usually the longest and/or the bigge t numbers appear immediately feasible to t he eye gaze.

Align decimal numbers properly eith er left or right. Right align integers (number s without decimals)

Examine the illustrations below and observe the most feasible.



In multiple columns tables as in the a bove illustrations, scanning across gaps bet een lines of data is hard. It is particularly harder with table of data containing large database fields.

sherbert	75
toffee	120
chocolate	35
fruit gums	27
coconut dreams	85

sherbert	75
toffee	120
chocolate	35
fruit gums	27
coconut dreams	85

sherbert	75
toffee	120
chocolate	35
fruit gums	27
coconut dreams	85

sherbert	75
toffee	120
chocolate	35
fruit gums	27
coconut dreams	85

Graying to highlight
Scanning across gaps hard here Use leaders alignment ca n help visuals

In multiple columns, use leaders (lin s that link data with their field names or de cription). See above
To highlight data within a table, you can light grey out the data. Do not colour, ot herwise the data would be covered from being feasible. See he last table above.

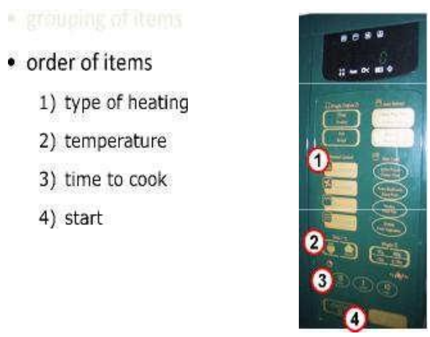
Use space to highlight

Use space to structure

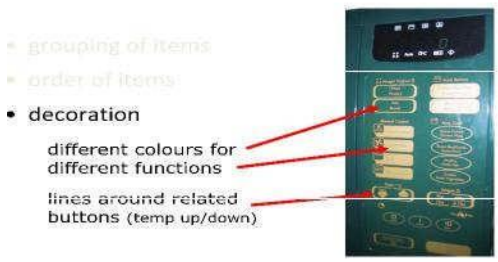
Observing the above tables of data, ne can see that spaces can be used to separ ate data, it can be used to structure and highlight as well.

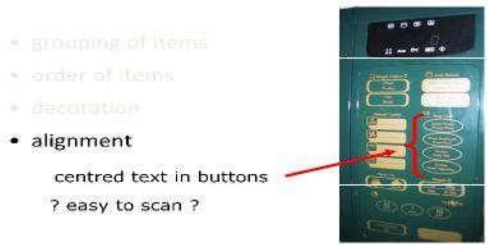
.5 PRESENTATION AND PHYSICA L CONTROLS OF DATA

5.1 Grouping and Ordering of Ite ms

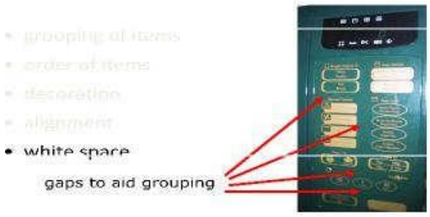


Here items of same functions are gro uped together Here i tems are displayed in order of priority of functions





Here different colours are used to di ferentiate functions Tex are centered in buttons here
with lines around related buttons



Gaps aid grouping here

Use white space within gaps to aid g ouping and proper alignment as in the abov e illustration.

5.2 Forms and dialogue boxes

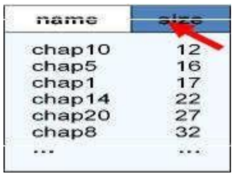
In designing forms and dialogue box s, the designer should pay attention to presentation of the form and dialogue box. He should also conside r how data would be entered into the form.He should consider the importance of similar layout issues c oncerning the form and dialogue boxes suchas alignment and the label lengths.
In presenting an effective logical layo ut, the designer should use task analysis, ap propriate groupings , a natural order for entering informatio n such as from top to bottom, left to right (depending on the culture adopted) and setting tab order for ke yboard entry.

Look at the illustration below:

The box or form in the middle (The s econd) presents an acceptable arrangement
The designer should indicate which a nd what area is active and passive such as w here the user should click and where to type.
The styles used should be consistent such as in web underlined links.
The labels and icons should have sta ndards for common actions. The language us ed in the labels should be bold and should represent the curre t state or action.

.5. 3 Creating 'affordances' in designs

The word 'affordances' is a psycholo ical term used for physical objects. The sha pes and sizes of the objects suggest actions to be taken o n the object. Actions such as pick up, twist a nd throw. So in a user interface terminology, one can say th at buttons 'afford' pushing depending on th eir state.
For screen objects, the button-like o ject 'affords' mouse click while the physical -like objects suggest use. There is a culture of computer langu ge use such as icons 'afford' clicking or eve double clicking.



The designer should design appropri te appearance for interface objects to pres nt accurate information. Aesthetics, utility, colour and 3D feat ures could be added for appropriate usability.
In presenting information, purpose atters. The purpose would enable the desi ner determine the sort order, in which column would the da ta be input and whether it is numeric or alp abetic. See the table of figures above.
The designer should consider using t ext or diagram to make his presentation effe ctive. He can also consider presenting the information using graphs such as scatter graph or histog am.
In every decision on presentation, th e designer should use the paper presentatio n principles but add interactivity. An example is a 'dancin g histogram'.

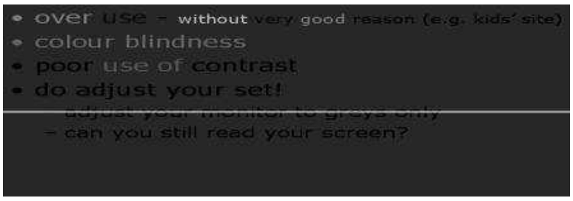
.5.4 Aesthetics and Utility

Aesthetically pleasing designs increa e user satisfaction and improve productivity . Beauty and utility may however sometimes conflict.
Mixed up visual styles make present tion easy to distinguish.
Clean design and little differentiation leads to confusion on the part of the user. or example, backgrounds behind text may be good to look at, ut hard to read. Look at the illustration abo ve. Both can however work together if carefully done, as d monstrated in the design of the counter in onsumer products presentation.

3.5.5 Using Colour and 3D in pr sentation.
Both Colour and 3D effects are often used very badly!
In using colour, the designer should remember that older monitors have limited alette of colours while modern monitors have millions of colours to manipulate hence colour is over use d because of its abundance.
The designer should also beware of c olour blindness as a result of using too man colours. Colours should be used sparingly in order to reinforc e other information.

D effects are good for physical infor mation and some graphs but if over used as in text in perspective and D pie charts, it can blur Information .

Bad Use of Colour



Example of Bad Use of Colour

A bad use of colour is an over use of colour without very good reason (e.g. kids' s ite) which may eventually cause colour blindness .

Poor use of contrast as occurs when you do adjust your set! For example, when y ou adjust your monitor to greys only, you may not be able to re ad your screen.

Across countries and cultures, there is localisation and internationalisation that r equire changing interfaces for particular cultures and languages .

In globalisation, when you try to cho ose symbols etc. that work everywhere, you simply change language and use 'resource' database instead of literal text. But changes are required on si zes and left-right order, etc.

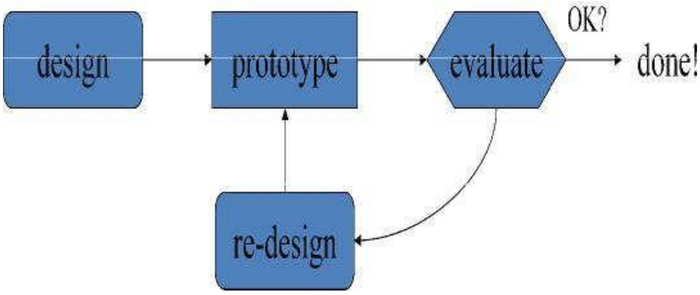
3.6 PROTOTYPING

Prototyping is an essential compone t of interactive system design.

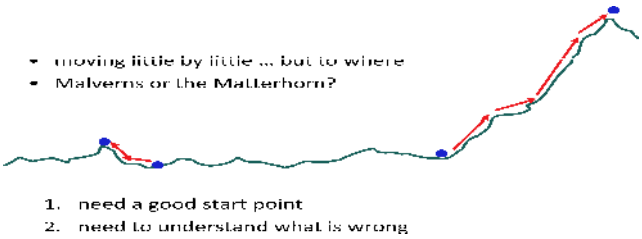
Prototypes may take many forms, fr m rough sketches to detailed working proto types. They provide concrete representations of design i eas and give designers, users, developers a d managers an early glimpse into how the new system wil l look and feel. Prototypes increase creativity, allow early evaluation of design ideas, help designers think through, solve design problems, and suppo rt communication within multidisciplinary design teams.

Prototypes are concrete and are not abstract; hence provide a rich medium for e xploring a design space. They suggest alternate design paths and reveal important details about particula r design decisions. They force designers to be creative and to articulate their design decisions. Prototypes embody design ideas and encourage designers to confront the r differences of opinion.

The precise aspects of a prototype o fer specific design solutions and designers c an decide to generate and compare alternatives. The imprecise or incomplete aspects of a prototype highlig ht the areas that must be refined or require additional ideas.



In prototyping you never get it right irst time; if at first you don't succeed then c ontinue ...



Pitfalls of prototyping are

Moving little by little ... but to where . These pitfalls are avoided by Malverns or t e Matterhorn rules that say that

1. the designer needs a good start point and 2. needs to understand what is rong

4.0 CONCLUSION

Design is an active process of workin g with a design space, expanding it by gener ating new ideas and contracting as design choices are ma de. Designing effective interaction is difficult and many interactive systems (including many websites) h ave a good look but a poor feel.

The quality of interaction is tightly lin ked to the end users and a deep understand ing of their work practices. Designers must take the c ntext of use into account when designing th e details of the interaction.

Prototypes are flexible tools that he p designers envision a design space, reflectupon it, and test their design decisions. Prototypes are dive rse and can fit within any part of the design .process, from the earliest ideas to the final details of the desig Prototypes provide one of the most effect ive means for designers to communicate with each other, as we ll as with users, developers, and managers, t hroughout the design process.

5.0 SUMMARY

The processes of design comprise re uirements, analysis, the design itself, Intera ction and prototyping, Implementation and deployment.

The basic principles in screen design and layout are the grouping of data that the screen displays, the structure, order and their alignment.

In designing forms and dialogue box s, the designer should pay attention to pres entation and purpose of presentation.

'Affordances' is a term used to refle t on the shapes and sizes of physical object that suggest actions to be taken on the object.

Aesthetically pleasing designs increa e user satisfaction and improve productivity while mixed up visual styles make presentation easy to distinguish.

The designer should beware of colour blindness resulting from using too many co lours. Colours should be used sparingly in order to reinforce other information.

A bad use of colour is an over use of colour without very good reason and it may eventually cause colour blindness .

3D effects are good for physical infor mation and some graphs but if over used ca n blur Information.

Prototypes provide concrete represe ntations of design ideas and give designers, users, developers and managers an early glimpse into how the new system will look and feel.

6.0 Tutor Marked Assignment

12. Define the term Pattern,
13. Mention any four of the s articularly as it relates to Human Computer Interaction (HCI) Id
14. For effective commencemix characteristics of design patterns ent of ask himself certain questions as guides, whatthe design process, the designer shou are those likely areas of question?

15. What are those issues that are likely to be considered by the designer during his design process?
16. Describe the design lifecycle of a typical design pattern
17. What are the major user focuses and cultural probes that can guide the pattern designer in his design process?
18. (i) Produce the hierarchical diagram that relates functional parts of applications with their groups of screens. What are the advantages of this diagram to designer?
    (ii) Draw a network diagram that shows the relationship between applications. What are the significances of the network diagram when considering design style issues?
19. Describe the basic principles governing screen design and layout
20. What are the concepts the designer should consider for effective presentation and physical controls of data? How do you relate these concepts specifically to the design of forms and dialogue boxes?
21. What do you understand by the term "Affordances in designing"? Do you agree that the designer should use "Affordances" concepts in his design process, and why?

(c) Why do designers have to include aesthetics, utilities, and 3D effects in their design patterns?

(d) What are the negative implications of over applying these effects in the designs?

22. Itemize the benefits of prototyping interactive System designs

7.0 Further Readings / References

- Gamma, Erich, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software.* Reading, MA: Addison-Wesley, 1995.
- Alexander, Christopher, Sara Ishikawa, Murray Silverstein, Max Jacobsen, Ingrid Fiksdahl-King, and Shlomo Angel. *A Pattern Language.* New York: Oxford University Press, 1977.
- Norman, Donald. *The Design of Everyday Things.* New York: Basic Books, 1988.
- Cypher, Allen. "Eager: Programming Repetitive Tasks By Example." In *Proceedings of CHI '91*, ACM, 1991.

UNIT 1: DESIGN OF USER INTERFACE CONCEPTS

Table of contents

1.0 INTRODUCTION

This unit offers guidelines for design of user interface software in six functional areas: data entry, data display, sequence control, user guidance, data transmission, and data protection.

The guidelines are proposed here as a potential tool for designers of user interface software.

Guidelines can help establish rules for coordinating individual design contributions and can also help to make design decisions just once rather than leaving them to be made over and over again by individual designers. They can help define detailed design requirements and to evaluate user interface software in comparison with those requirements.

The design of user interface software will often involve a considerable investment of time and effort. Design guidelines can help ensure the value of that investment.

In designing computer-based information systems, special attention must be given to software supporting the user interface.

4.0 OBJECTIVES

By the end of this unit, you should be able to:

- Explain the concept of User Interface Design
- Understand User System Interface
- Know the significance of User Interface
- Explain the function of User Interface Software
- Describe the interaction design phases
- Understand the concept of formative evaluation

.0 MAIN CONTENT

.1 USER INTERFACE DESIGN OF INFORMATION SYSTEMS

Computers today are used for a broad range of applications. User interface design guidelines cannot be applied usefully in every case. Some computers may be embedded as components in larger systems, so that they communicate only with other computers and not directly with human users. When there is no user interface, then no user interface design guidelines are needed.

To the extent that information systems support human users performing defined tasks, careful design of the user-system interface will be needed to ensure effective system operation. The guidelines are intended to improve user interface design for such information systems.

Users of information systems interact with a computer in order to accomplish information handling tasks necessary to get their jobs done. They differ in ability, training and job experience. They may be keenly concerned with task performance, but may have little knowledge of (or interest in) the computers themselves. Design of the user-system interface must take account of those human factors.

3.1.1 USER-SYSTEM INTERFACE

User-system interface is broadly defined to include all aspects of system design that affect system use. Hence we are concerned with the user interface to computer-based information systems, i.e., with those aspects of system design that influence a user's participation in information handling tasks.

It focuses on those design features of the user interface that are implemented via software (i.e., the design of computer program logic) rather than hardware (the design of equipment). The guidelines are worded in terms of the functions that a user must perform, and the functional capabilities that a designer should provide, rather than the particular physical devices that might be used to implement those functions. Thus a particular guideline might deal with "pointing" as a function, with no necessary recommendation whether pointing should be accomplished via touch display or light pen or any other physical device. Software is not the only significant factor influencing user performance. Other aspects of user interface design are important, including workstation design, physical display characteristics, keyboard layout, environmental factors such as illumination and noise, and the design of paper forms and written documentation, user training courses, etc. To achieve a good user interface design, all of those factors must be designed with care.

3.1.2 USER INTERFACE SOFTWARE

What sets data processing systems apart as a special breed is the function of each switch button, the functional arrangement among the buttons. The size and distribution of elements within a display are established not in the design of the equipment but in how the computer is programmed. The 'design' in the programs equally establishes the contents of processed data available to the operator and the visual relationships among the data. In combination with or in place of hardware, it can also establish the sequence of actions which the operator must use and the feedback to the operator concerning those actions.

User interface design cannot be the concern only of the psychologist or the human factors specialist. It is a significant part of information system design that must engage the attention of system developers, designers, and ultimately system users as well.

In designing computer-based information systems, special attention must be given to software supporting the user interface.

A comprehensive set of guidelines for design of user interface software in computer-based information systems exist in another unit of this study pack. Also, the general problems of user interface design and the particular need for guidelines to design user interface software are identified,.

3.1. 3 INFORMATION SYSTEMS AND INTERFACE USERS

Computers today are used for a broad range of applications. User interface design guidelines cannot be applied usefully in every case. Some computers may be embedded as components in larger systems, so that they communicate only with other computers and not directly with human users. When there is no user interface, then no user interface design guidelines are needed.

The particular tasks for which a general-purpose computer might be used are not defined in advance by the designer. Instead, a user must provide exact instructions to program the computer to perform any task at hand. The designer may try to ensure that the computer can process appropriate programming languages, but otherwise is not concerned with explicit design of a user interface.
To the extent that information systems support human users performing defined tasks, careful design of the user-system interface will be needed to ensure effective system operation.
Users of information systems interact with a computer in order to accomplish information handling tasks necessary to get their jobs done. They differ in ability, training and job experience. They may be keenly concerned with task performance, but may have little knowledge of (or interest in) the computers themselves. Design of the user-system interface must take account of those human factors.

3.2 SIGNIFICANCE OF THE USER INTERFACE

The design of user interface software is not only expensive and time-consuming, but it is also critical for effective system performance.
In a constrained environment, such as that of many military and commercial information systems, users may have little choice but to make do with whatever interface design is provided. There the symptoms of poor user interface design may appear in degraded performance. Frequent and/or serious errors in data handling may result from confusing user interface design. Tedious user procedures may slow data processing, resulting in longer queues at the checkout counter, the bank cashier window, the visa office, a company's security check out point, or any other workplace where the potential benefits of computer support are outweighed by an unintended increase in human effort.
In situations where degradation in system performance is not so easily measured, symptoms of poor user interface design may appear as user complaints. The system may be described as hard to learn, or clumsy, tiring and slow to use. The users' view of a system is conditioned chiefly by experience with its interface. If the user interface is unsatisfactory, the users' view of the system will be negative regardless of any niceties of internal computer processing.
A data entry application in which relatively simple improvements to user interface software -- including selection and formatting of displayed data, consistency in wording and procedures, on-line user guidance, explicit error messages, re-entry rather than overtyping for data change, elimination of abbreviations, etc. - - resulted in significantly improved system performance. Data entry was accomplished 25 percent faster, and with 25 percent fewer errors. How can that kind of design improvement be achieved in general practice?

3. 3 INTERFACE DESIGN PRACTICE

User interface software design can be regarded as art rather than science.
As an art, user interface design is best practiced by experts, by specialists experienced in the human engineering of computer systems. Most established information systems, call for a system development sequence starting with requirements analysis, functional specification and verification before any software design begins. The actual course of user interface software development will sometimes depart from that desired sequence. There may be no explicit attempt to determine user interface requirements.
Specifications may include only rudimentary references to user interface design, with general statements that the system must be "easy to use". In the absence of effective guidance, both the design and implementation of user interface software may become the responsibility of programmers unfamiliar with operational requirements. Detection and correction of design flaws may occur only after system prototyping, when software changes are difficult to make.

3.4 THE INTERACTION DESIGN PHASES

In this phase, the user interface is specified, sketched, developed, and tested. The goal is to support the identified issues during context, task and user analyses and to meet the HCI evaluation metrics requirements. Design is also based on accepted conventions and experience.

The main activities are interface specification and formative evaluations. Interface specification includes semantic understanding of the information needs to support systems requirements and HCI analysis results.
The syntactical and lexical decisions include metaphors, media, dialogue, and presentation designs. Details of these are given below.

3.4.1 METAPHOR AND VISUALIZATION DESIGN

Metaphor and visualization design helps the user develop a mental model of the system. It is concerned with finding or inventing metaphors or analogies that are appropriate for users to understand the entire system or part of it. Well accepted metaphors include a shopping cart for holding items before checking out in E-Commerce context, and light bulbs for online helps or daily tips in productivity software packages.

3.4.2 MEDIA DESIGN

Media design is concerned with selecting appropriate media types for meeting the specific information presentation needs and human experience needs. Popular media types include text, static images (e.g., painting, drawing or photos), dynamic images (e.g., video clips and animations), and sound. The bandwidth needed for transmitting information depends on the media type. In addition, some media types contain affective qualities that can make presentations more interesting and stimulating, or annoying and distasteful.

3.4. 3 DIALOGUE DESIGN

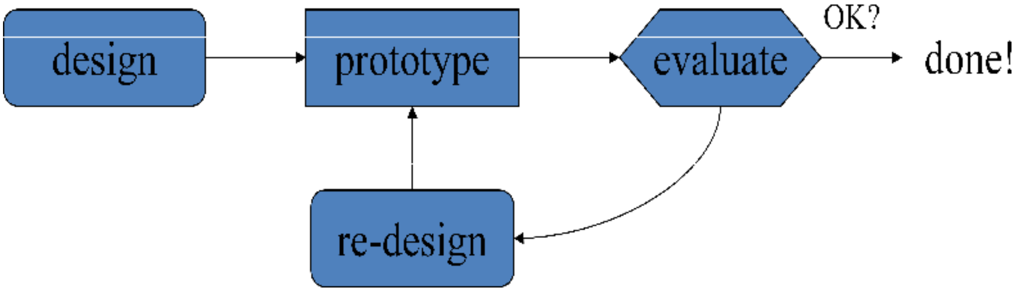
Dialogue design focuses on how information is provided to and captured from users during a specific task. Dialogues are analogous to a conversation between two people. Many existing interaction styles can be used such as menus, forms, natural languages, dialog boxes, and direct manipulation.

3.4.4 PRESENTATION DESIGN

Presentation design concerns the decisions on information architecture and display layout incorporating metaphors, media, and dialogue designs with the rest of the displays.
Commonly established user interface design principles and guidelines may be applied during the design stage.
For example, the following presentation design principles were suggested by Sutcliffe 1997:
1.Maximize visibility —this means all necessary information should be immediately available.
2.Minimize search time with minimum keystrokes
3. Provide structure and sequence of display
4. Focus on user attention on key data — here, important information should be salient and easilycomprehended
S. Provide only relevant information and
6. No overloading of user's working memory.

3.4 FORMATIVE EVALUATION

Formative evaluations identify defects in designs thus inform design iterations and refinements. A variety of different formative evaluations can occur several times during the design stage to form final design decisions. In fact, we propose that formative evaluations occur during the entire HCI development life cycle, as depicted below.



4.0 CONCLUSION

Current user interfaces and their und erlying systems are just too frustrating for for software developers should be to increase the reliability and usability of ever ostusersAmajoreffortsystem.Thefrequencyof system reboots, application crashes, and incompatible file formats that stop user s in the middle of their tasks must be reduced. Then the inc mprehensible instructions, ambiguous men us, and troubling dialog boxes need to be revised to enable users to complete their work promptly and c nfidently. Long download times of web pages, dropped sessions for networked applications, and the disrup tion caused by unsolicited email (spam), and destructive viruses. As the number of users has grown, all thes e problems have become more serious. Novice users want the benefits of email or web services, but they a re the poorly equipped and motivated to overcome the problems that high-tech early adopters proudly ns onquered. Improved training can help, but improved desi are an important component of making t ers he next generation of interfaces more appealing and the u more satisfied.

5.0 SUMMARY

User-system interface includes all as pects of system design that affect system us particularly on those er design features of the user interface that are implemented via software.

Special attention must be given to so ftware supporting the user interface and ot physical and human factors influencing user performance .

The users' view of a system is conditioned chiefly by experience with its interface d.

User interface design is best practiceby experts, by specialists experienced in th e human engineering of computer systems.

The interaction design phase is made up of Metaphor and Visualization Design, Media Design, Dialogue Design, Presentation Design, and the Formative Evaluation of the designs.

6.0 Tutor Marked Assignment

8. What do you understand as t he user- system interface?
9. 'User interface design cannot be the concern only of the psychologist or t he human factors ance of this statement?

he design formative evaluation?
ation systems and user interface

12. What is the significance of th
13. What are the implications of
14. Mention and explain briefly, t specialist'. What is the signifi
10. What do you understand as t
11. Differentiate between inform

e user interface to the user and other stakeh olders in an organization?
he absence of an effective guidance to use r interface design?
he four syntactical/lexical phases of interact ion design.

7.0 Further Readings / Reference

- Cooper, Alan. *About Face: The Essentials of User Interface Design.* Foster City, CA: IDG Books Worldwide, 1995.
- Kasik, D.J. "A User Interface 11 anagement System," in *Proceedings SIGGRAPH'82: Computer Graphics.* 1982. Boston, MA. 16. pp. 99-106.

- D. Hix, R. Hartson: Developing user interfaces. Wiley, 199 3.
- L. Barfield: The user interface - concepts & design. Addison Wesley, 199 3.
- Monk, P. Wright, J. Haber, L. Davenport: Improving your Human-Computer Interface: a practical technique. Prentice Hall, 199 3.
- Smith, D.C., *et al.* "The Star User Interface: an Overview," in *Proceedings of the 1982 National Computer Conference.* 1982. AFIPS. pp. 515-528.

- Jef Raskin: *The humane interface. New directions for designing interactive systems*. Addison-Wesley, Boston 2000 ISBN 0-201- 379 37-6
- Ben Shneiderman and Catherine Plaisant: *Designing the User Interface: Strategies for Effective Human—Computer Interaction*. 4th ed. Addison Wesley, 2004 ISBN 0- 321-19786-0
- Card, S.K., "Pioneers and Settlers: Methods Used in Successful User Interface Design," in *Human-Computer Interface Design: Success Stories, Emerging Methods, and Real-World Context*, M. Rudisill, *et al.*, Editors. 1996, Morgan Kaufmann Publishers: San Francisco. pp. 122-169.
- Myers, B.A., "A Taxonomy of User Interfaces for Window Managers." *IEEE Computer Graphics and Applications*, 1988. 8(5): pp. 65-84.

UNIT 2: USER INTERFACE DESIGN PRINCIPLES AND CRITERIA/RATIONALE

Table of contents

1.0 INTRODUCTION

The human computer interface can be described as the point of communication between the human user and the computer. The flow of information between the human and the computer is defined as the loop of interaction.

Design criteria and principles are important to designing a new user interface and to evaluate a current user interface.

There are seven principles that may be considered at any time during the design of a user interface and these are: Tolerance, Simplicity, Visibility, Affordance, Consistency, Structure and Feedback. These are briefly discussed in this unit.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Carry out and evaluate simple design using some primary design principles

- Experiment design using some experimental design principles
- Explain the 1 3 principles of display design
- Learn the Norman's 7 design principles
- Know the concepts and types of design rationale

5.0 MAIN CONTENT

.1 PRIMARY DESIGN PRINCIPLES

The following seven principles mentioned in the introduction above can be used to guide or evaluate design at any time in the process. The principles can be re-framed or re-structured to suit a particular company or project, or by professional designers.

Principle	Description	Example
Visibility	Clarity	Is the goal obvious? Are icons used?
Feedback	Information sent back to user after their action	Is the feedback in sound? Is there a label showing success or failure?
Affordance	How *clear* is the use of an element to the user?	Label "Push" on one side of a door; a button saying "Click Me"
Simplicity	Utilise the principles of usability	Place an Open File option on a menu, under File tag
Structure	Are the elements set out in a meaningful way from the perspective of the user?	Grouping information within a dialogue box.
Consistency	How easy is it to learn and remember the appearance, positioning and behaviour of the elements?	X to close a window is always on top right hand side of the window; the most important buttons are the same size with only labels indicating different goals.
Tolerance	Prevents user making errors or provides easy recovery or graceful failure	Ignoring of wrong or invalid keyboard input; Hiding options inappropriate in a context.

The loop of interaction has several aspects to it including the:
Task Environment: These are the conditions and goals set upon the user.
Machine Environment: This is the environment that the computer is connected, e.g. a laptop in a college student's dormitory room.
Areas of the Interface: Non-overlapping areas involve processes of the human and computer not pertaining to their interaction. While the overlapping areas only concern themselves with the processes pertaining to their interaction.
Input Flow: This begins in the task environment as the user has some task that requires using their computer.
Output: This is the flow of information that originates in the machine environment.
Feedback: These are loops through the interface that evaluate, moderate, and confirm processes as they pass from the human through the interface to the computer and back.

3.2 EXPERIMENTAL DESIGN PRINCIPLES

Experimental design principles are also important to evaluate a current user interface or to design a new user interface and they are described below:

Early focus on user(s) and task(s):

Establish how many users are needed to perform the task(s) and determine who the appropriate users should be; someone that has never used the interface, and will not use the interface in the future, is most likely not a valid user.

In addition, define the task(s) the users will be performing and how often the task(s) need to be performed.

Empirical measurement:

Test the interface early on with real users who come in contact with the interface on an everyday basis, respectively. Keep in mind that results may be altered if the performance level of the user is not an accurate depiction of the real human-computer interaction.

Establish quantitative usability specifics such as: the number of users performing the task(s), the time to complete the task(s), and the number of errors made during the task(s).

Iterative design:

After determining the users, tasks, and empirical measurements to include, perform the following iterative design steps:

Design the user interface

Test the interface design

Analyze results of using the interface

Repeat the iterative design process until a sensible, user-friendly interface is created.

Design Methodologies

A number of diverse methodologies outlining techniques for human—computer interaction design have emerged since the rise of the field in the 1980s.Most design methodologies stem from a model for how users, designers, and technical systems interact.

Early methodologies, for example, treated users' cognitive processes as predictable and quantifiable and encouraged design practitioners to look to cognitive science results in areas such as memory and attention when designing user interfaces.

Modern models tend to focus on a constant feedback and conversation between users, designers, and engineers and push for technical systems to be wrapped around the types of experiences users want to have, rather than wrapping user experience around a completed system.

User-centered design:

User-centered design (UCD) is a modern, widely practiced design philosophy rooted in the idea that users must take center-stage in the design of any computer system. Users, designers and technical practitioners work together to articulate the wants, needs and limitations of the user and create a system that addresses these elements. Often, user-centered design projects are informed by ethnographic studies of the environments in which users will be interacting with the system.

Display Design

Displays are human-made artifacts designed to support the perception of relevant system variables and to facilitate further processing of that information. Before a display is designed, the task that the display is intended to support must be defined (e.g. navigating, controlling, decision making, learning, entertaining, etc.). A user or operator must be able to process whatever information that a system generates and displays; therefore, the information must be displayed according to principles in a manner that will support perception, situation awareness, and understanding.

3. 3 THIRTEEN PRINCIPLES OF DISPLAY DESIGN

These are principles of human perception and information processing that can be utilized to create an effective display design.
A reduction in errors, a reduction in required training time, an increase in efficiency, and an increase in user satisfaction are a few of the many potential benefits that can be achieved through utilization of these principles.
Certain principles may not be applicable to different displays or situations. Some principles may seem to be conflicting, and there is no simple solution to say that one principle is more important than another. The principles may be tailored to a specific design or situation. Striking a functional balance among the principles is critical for an effective design.

The thirteen principles are:
Perceptual Principles
1. Make displays legible (or audible)
A display's legibility is critical and necessary for designing a usable display. If the characters or objects being displayed cannot be discernible, then the operator cannot effectively make use of them.
2. Avoid absolute judgment limits
Do not ask the user to determine the level of a variable on the basis of a single sensory variable (e.g. color, size, loudness). These sensory variables can contain many possible levels.
3. Top-down processing
Signals are likely perceived and interpreted in accordance with what is expected based on a user's past experience. If a signal is presented contrary to the user's expectation, more physical evidence of that signal may need to be presented to assure that it is understood correctly.
4. Redundancy gain
If a signal is presented more than once, it is more likely that it will be understood correctly. This can be done by presenting the signal in alternative physical forms (e.g. color and shape, voice and print, etc.), as redundancy does not imply repetition. A traffic light is a good example of redundancy, as color and position are redundant.
5. Similarity causes confusion: Use discriminable elements
Signals that appear to be similar will likely be confused. The ratio of similar features to different features causes signals to be similar. For example, A42 3B9 is more similar to A42 3B8 than 92 is to 9 3. Unnecessary similar features should be removed and dissimilar features should be highlighted.

Mental Model Principles
6. Principle of pictorial realism
A display should look like the variable that it represents (e.g. high temperature on a thermometer shown as a higher vertical level). If there are multiple elements, they can be configured in a manner that looks like it would in the represented environment.
7. Principle of the moving part
Moving elements should move in a pattern and direction compatible with the user's mental model of how it actually moves in the system. For example, the moving element on an altimeter should move upward with increasing altitude.

Principles Based on Attention
8. Minimizing information access cost

When the user's attention is averted from one location to another to access necessary information, there is an associated cost in time or effort. A display design should minimize this cost by allowing for frequently accessed sources to be located at the nearest possible position. However, adequate legibility should not be sacrificed to reduce this cost.

9. Proximity compatibility principle

Divided attention between two information sources may be necessary for the completion of one task. These sources must be mentally integrated and are defined to have close mental proximity. Information access costs should be low, which can be achieved in many ways (e.g. close proximity, linkage by common colors, patterns, shapes, etc.). However, close display proximity can be harmful by causing too much clutter.

10. Principle of multiple resources

A user can more easily process information across different resources. For example, visual and auditory information can be presented simultaneously rather than presenting all visual or all auditory information.

Memory Principles

11. Replace memory with visual information: knowledge in the world

A user should not need to retain important information solely in working memory or to retrieve it from long-term memory. A menu, checklist, or another display can aid the user by easing the use of their memory. However, the use of memory may sometimes benefit the user rather than the need for reference to some type of knowledge in the world (e.g. a expert computer operator would rather use direct commands from their memory rather than referring to a manual). The use of knowledge in a user's head and knowledge in the world must be balanced for an effective design.

12. Principle of predictive aiding

Proactive actions are usually more effective than reactive actions. A display should attempt to eliminate resource-demanding cognitive tasks and replace them with simpler perceptual tasks to reduce the use of the user's mental resources. This will allow the user to not only focus on current conditions, but also think about possible future conditions.

An example of a predictive aid is a road sign displaying the distance from a certain destination.

13. Principle of consistency

Old habits from other displays will easily transfer to support processing of new displays if they are designed in a consistent manner. A user's long-term memory will trigger actions that are expected to be appropriate. A design must accept this fact and utilize consistency among different displays.

3.4 The Norman's 7 Design Principles

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

3.4.1 DESIGN PRINCIPLES FORMULATED TO SUPPORT USABILITY :

Principle of Learnability : This is the ease with which new users can begin effective interaction and achieve maximal performance

Principle of Flexibility: These are the multiplicity of ways the user and system exchange information

Principle of Robustness: This is the level of support provided the user in determining successful achievement and assessment of goal-directed behaviour

The Principles of learnability are broken down into :

Predictability : This is determining effect of future actions based on past interaction history and its operation visibility
Synthesizability: This is assessing the effect of past actions, its immediate and its eventual honesty
Familiarity: This is how prior knowledge applies to new system and how easy one can guess its affordance Generalizability: This is extending specific interaction knowledge to new situations
Consistency: This concerns the likeness in input and output behaviour arising from similar situations or task objectives

Principles of flexibility comprise:
Dialogue initiative : This is the freedom from system imposed constraints on input dialogue and it compares the system against the user pre-emptiveness.
Multithreading: This is expressing the ability of the system to support user interaction for more than one task at a time. It also looks at the concurrent and interleaving multimodality.
Task migratability: This is passing responsibility for task execution between user and system
Substitutivity: This allows equivalent values of input and output to be substituted for each other. It compares representation multiplicity and equal opportunity
Customizability: This is the modifiability and adaptability of the user interface by user or the modifiability and adaptivity of the user interface by the system.

Principles of robustness are made up of:
Observability: This is the ability of the user to evaluate the internal state of the system from its perceivable representation. It considers the browsability, the defaults, the reachability, the persistence , and the operation visibility.
Recoverability: This concerns the ability of the user to take corrective action once an error has been recognized. It looks at the reachability, the forward and backward recovery and the commensurate effort.
Responsiveness:This is how the user perceives the rate of communication with the system and how stable is the response.
Task conformance: This explains the degree to which system services support all of the user's tasks, the task completeness and its adequacy.

3.5 THE DESIGN RATIONALE

Design rationale is an information that explains why a computer system is the way it is.
Benefits of design rationale are :
Communication exists throughout the life cycle
Reuse of design knowledge is made across products
Design rationale enforces design discipline
It presents arguments for design trade-offs
It organizes potentially large design space
It is used to capture contextual information

3.5.1 Types of Design Rationale:

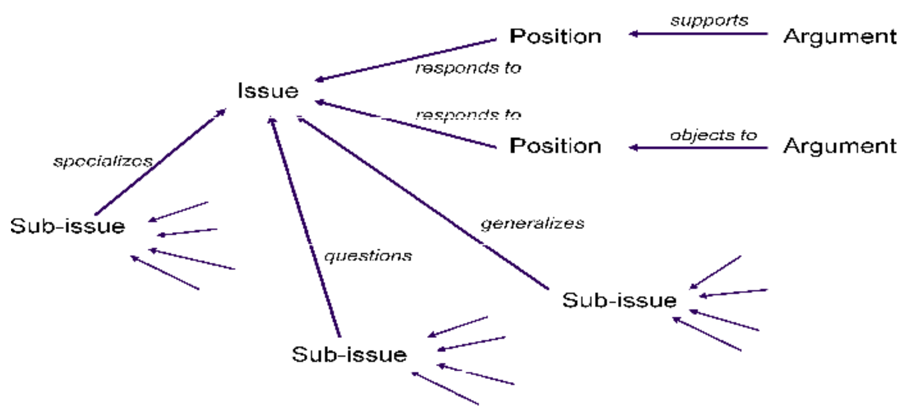
Process-oriented: this preserves order of deliberation and decision-making
Structure-oriented: this emphasizes post hoc structuring of considered design alternatives
Two examples design rationale are:
Issue-based information system (IBIS) and
Design space analysis

Issue-based information system (IBIS)
The Issue-based information system (IBIS) provides basis for much of design rationale research and it is process-oriented. The main elements are :
Issues: These describe the hierarchical structure with one 'root' issue

Positions: These contain the potential resolutions of an issue

Arguments: These modify the relationship between positions and issues

The graphical version or structure of IBIS (gIBIS) is produced below



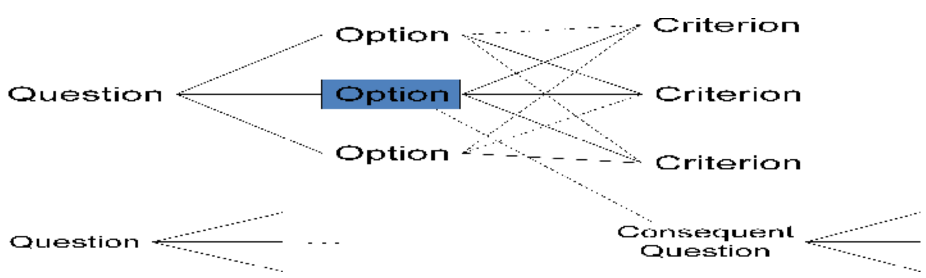
Design space analysis

This is structure-oriented

QOC (Questions, Options and Criteri) is a hierarchical structure made up of questions (and sub-questions) and represent major issues of a design

The options provide alternative solutions to the question while the criteria is the means to assess the options in order to make a choice

The QOC notation (Question, Option and Criterion)



5.2 Characteristics of psychological design rationale

- To support task-artefact cycle in which user tasks are affected by the systems they use
- It aims to make explicit consequences of design for users
- Designers identify tasks the stem will support
- Various scenarios are suggested to test task
- Users are observed while usi g the system
- The psychological claims of the system are made explicit
- The negative aspects of the design can be used to improve next iteration of design

4.0 CONCLUSION

The various design principles mentioned can be used to guide or evaluate design at any time in the process. The principles can be re-framed or re-structured to suit a particular company or project, or by professional designers.

5.0 SUMMARY

Experimental design principles are important to evaluate a current user interface or to design a new user interface. They comprise the empirical measurement and iterative design.

Most design methodologies stem from a model on how users, designers, and technical systems interact. User-centered design (UCD) is a modern, widely practiced design philosophy rooted in the idea that users must take center-stage in the design of any computer system

Displays are human-made artifacts designed to support the perception of relevant system variables and to facilitate further processing of that information.

Principles of display design comprise the Perceptual Principles, the Mental Model Principles, Principles Based on Attention, and the Memory Principles

Design rationale is an information that explains why a computer system is the way it is, it could be Process-oriented and/or Structure-oriented.

6.0 Tutor Marked Assignment

7. Briefly describe the seven principles to be considered while designing a user interface
8. Explain the 5 aspects that govern the loop of human Computer interaction.
9. Why is it necessary for an interactive designer to consider experimental design principles during his design? Mention some of these principles
10. The thirteen principles of display design are Principles of human perception and information processing that can be utilised by the designer to create an effective display.
    (c) What are the potential benefits achievable through utilization of these principles?
    (d) Mention any 2 principles under each of the following categories (i) Perceptual (ii) Mental model (iii) Those based on attention and (iv) memory
11. Mention the five benefits of a design rationale
12. (a) What is the objective of a design rationale?
    (b)Produce the graphical versions of an example of a design rationale. You can select either the given examples of issue based information System (IBIS) or the Design space analysis
7. What are the characteristics of the psychological design rationale?

7.0 Further Readings / References

- C. Brown: Human-Computer Interface design guidelines. Ablex, 1989.
- Cooper, Alan. *About Face: The Essentials of User Interface Design.* Foster City, CA: IDG Books Worldwide, 1995.
- Kasik, D.J. "A User Interface Management System," in *Proceedings S IGGRAPH'82: Computer Graphics.* 1982. Boston, MA. 16. pp. 99-106.
- D. Hix, R. Hartson: Developing user interfaces. Wiley, 199 3.
- L. Barfield: The user interface - concepts & design. Addison Wesley, 1993.
- Monk, P. Wright, J. Haber, L. Davenport: Improving your Human-Computer Interface: a practical technique. Prentice Hall, 199 3.
- W. Galitz: Handbook of screen format design. QED, 1989.
- C. Gram, G. Cockton (eds.): Design principles for interactive software. Capman & Hall, 1996.
- ISO 9241 (Part 10: Dialogue principles, Part 12: Presentation of information, Part 14: Menu dialogues, Part 15: Command dialogues, Part 16: Direct manipulation dialogues, Part 17: Form fill-in dialogues)
- D. Mayhew: Principles and guidelines in software user interface design. Prentice, 1992.

UNIT 3: USER INTERFACE DESIGN PROGRAMMING TOOLS

Table of contents

1.0 INTRODUCTION

The programming tools for the design of user interface give implementation support for the levels of services for programmers. These include the windowing systems that provide the core support for separate and simultaneous user-system activity. They enable easy programming of the application and the control of dialogue between the system and the user. The interaction toolkits for example, bring programming closer to the level of user perception while the user interface management systems control the relationship between the presentation and functionality.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Explain the various levels of programming support tools
- Utilize toolkits for programming interaction objects
- Understand the concepts of the User Interface Management Systems(UIMS)

.0 MAIN CONTENT

3.1 How Human Computer Interaction affects the programmer

Advances in coding have elevated programming through hardware that specifically improves upon the programmer's Interaction-technique.

The layers of development tools, as earlier mentioned, also contribute to how human computer interaction affects the programmer. These tools incorporate the windowing systems, the interaction toolkits and the user interface management systems as exemplified in the following:

Levels of programming support tools

- Windowing systems
  - device independence
  - multiple tasks
- Paradigms for programming the application
  - read-evaluation loop
  - notification-based

- Toolkits
  - programming interaction objects
- UIMS
  - conceptual architectures for separation
  - techniques for expressing dialogue

3.1.1 Elements of the Windowing Systems

Device independence

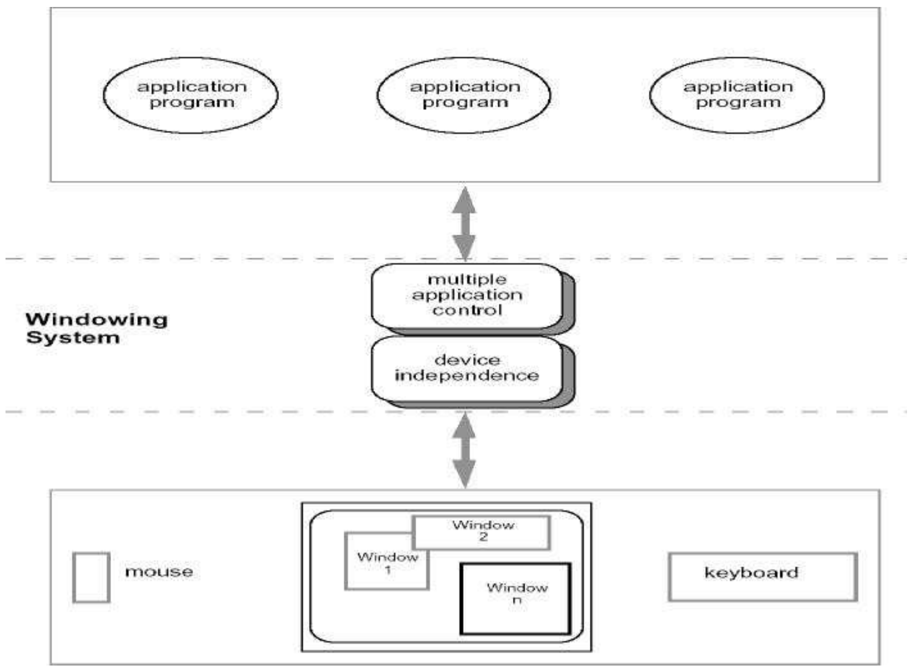
Programming the abstract terminal device drivers using the image models for output and input is device independent. Also device independence is the creation of the image models for output and the input, partially. These image models are the pixels, the PostScript (as in Macintosh Operating System X and NextStep), the Graphical Kernel System (GKS) and the Programmers' Hierarchical Interface to Graphics (PHIGS)

Resource sharing

Another element of the windowing system is resource sharing. This is the act of achieving simultaneity of user tasks. Resource sharing enables the use of the window system to support independent processes by the isolation of individual applications.

Elements of windowing systems

3.1.2 Role of a windowing system



As shown in the diagram above, the windowing system comprising the multiple application control and the device independent control enables the interface between the application programs and the user.

3.1. 3 The Architectures of windowing systems

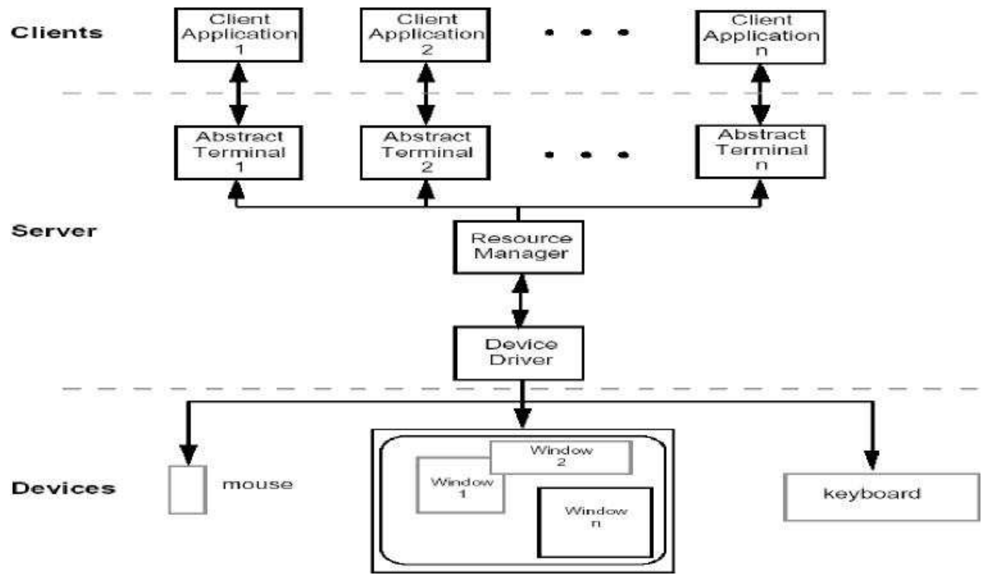
The Architectures of windowing systems are analysed through three possible software architectures if we all assume device driver is separate and know how they differ and how the multiple application management is implemented.

The three possible software architectures are in the following forms:

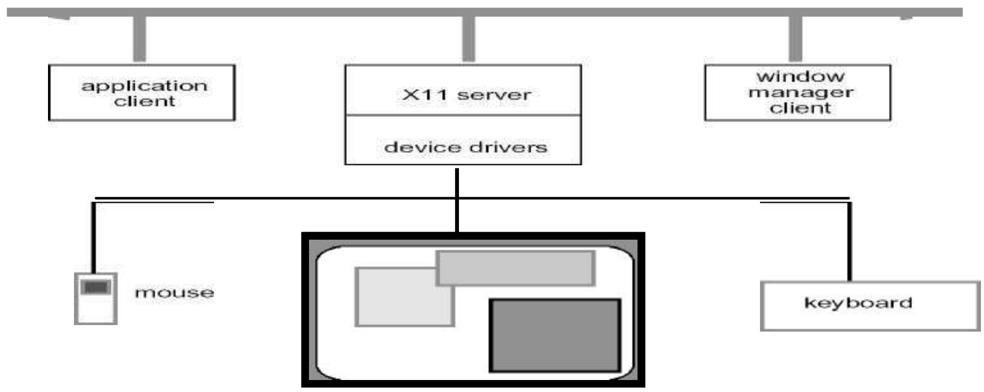
1. When each application manages all processes. Here, everyone worries about synchronization and reduces portability of applications

2. When management role within kernel of operating system ensures that applications are tied to operating system, and
3. When management role as separate application ensures maximum portability

The client-server architecture is illustrated below:

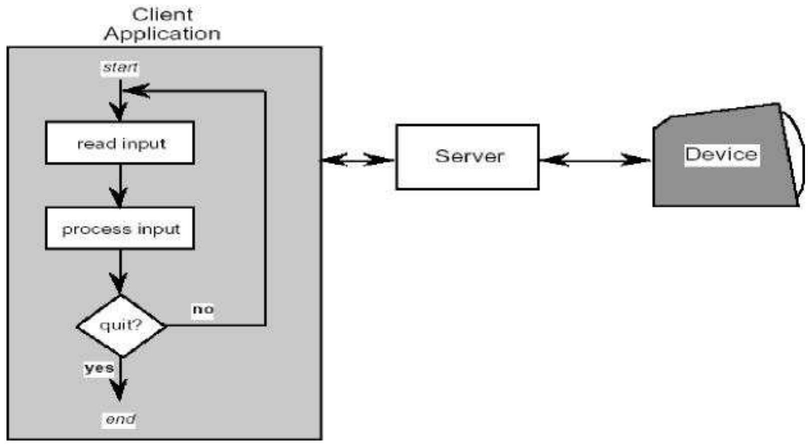


3.1.4 X Windows architecture
The X Windows architecture comprises the Pixel imaging model with some pointing mechanism and the X protocol that defines the server-client communication.
The architecture also contains a separate window manager client that enforces policies for input and output. Policies on how to change input focus, how the tiled windows compare with overlapping windows and policies on inter-client data transfer. See the pictorial illustration below.



.1.5 Typical program models of the application:

Programming the application - 1



A typical read-evaluation loop is provided below:

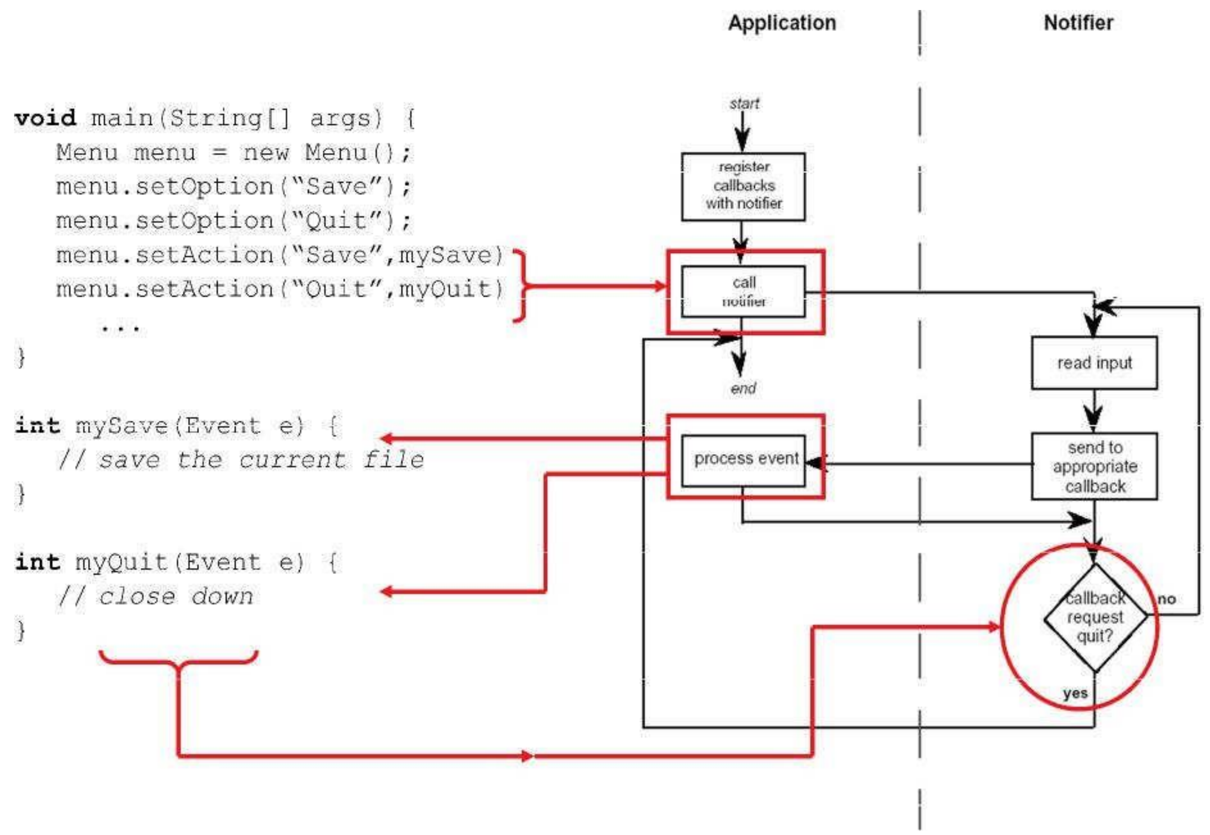
repeat

    read-event(myevent)
    case myevent.type
        type_1:
            do type$_G$1 processing
        type_2:
            do type$_G$2 processing
            type_n:
            do type$_G$n processing
    end case

end repeat

Programming the application - 2

Notification-based



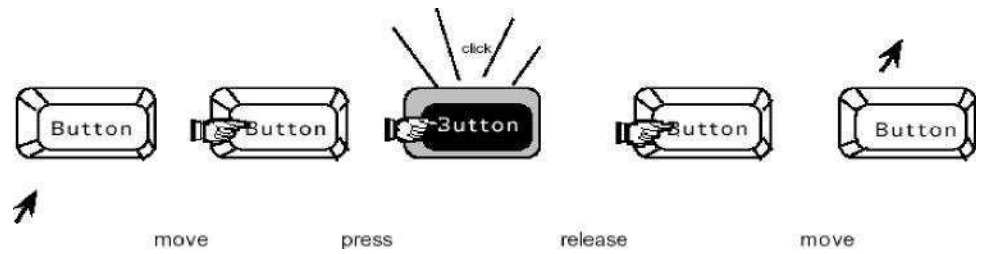
.2 USING TOOLKITS

3.2.1 User Interface Toolkits

User interface toolkits are probably t he most widely used tool nowadays to impl ment applications. All three major platforms (Unix/Linux, MacOS, and Windows) c ome with at least one standard UI toolkit.

Toolkits are interaction objects withinput and output intrinsically linked. They en able programming with interaction techniques using and gadgets. They promote consistency and generalizations through widgets similar look and feel.

A widget is a software object that ha s three facets that closely match the MVC model: a presentation, behavior, and an application interfac e. Sample illustration of widgets is provided a below.



The presentation defines the graphical aspect of the widget. The overall presentation of an interface is tree. created by assembling widgets intoWidgets, such as buttons are the leav s of the tree. Composite ree and a widgets constitute the nodes ofcontrol the layout of their children. The behavior of a widget pports: a the t defines the interactionbutton can be pressed, a scrollbar can be scrolled, and a text methods it su field can be edited.

The application interface defines ho a widget communicates the results of the ser interaction to the of the application. It is usually based rest on a notification mechanism.

One limitation of widgets is that their behaviors are limited to the widget itself. Interaction techniques that involve multiple widgets, such as drag-and-drop, cannot be supported by the widgets' behaviors alone and require separate support in the UI toolkit.

3.2.2 Prototypes and Widgets

In general, prototyping new interaction techniques requires either implementing them within new widget classes, which is not always possible, or not using a toolkit at all. Implementing a new widget class is typically more complicated than implementing the new technique outside the toolkit, (for example, with a graphical library), and is rarely justified for prototyping. Many toolkits provide a blank widget, such as the Canvas in Tk or JFrame in Java Swing, which can be used by the application to implement its own presentation and behavior. This is usually a good alternative to implementing a new widget class, even for production code.

A number of toolkits have also shifted away from the widget model to address other aspects of user interaction. For example, GroupKit was designed for groupware, Jazz for zoomable interfaces, the Visualization and InfoVis toolkits for visualization, Inventor for 3-D graphics, and Metisse for window management.

Creating an application or a prototype with a UI toolkit requires solid knowledge of the toolkit and experience with programming interactive applications. In order to control the complexity of the interrelations between independent pieces of code (creation of widgets, callbacks, global variables, etc.), it is important to use well-known design patterns, otherwise, the code quickly becomes unmanageable and, in the case of a prototype, unsuitable to design models.

Toolkits are amenable to object-oriented programming using Java interfaces that include the AWT (abstract windowing toolkit), a Java toolkit, and some Java classes for buttons and menus, etc.

Some Java interfaces are (i) Notification based such as AWT 1.0 with the need to subclass basic widgets and AWT 1.1 and beyond with call-back objects (ii) Swing toolkit built on top of AWT with higher level features that also uses the MVC architecture.

3.2. 3 The User Interface Management Systems (UIMS)

The UIMS add another level above toolkits because toolkits may be too difficult for non-programmers. Concerns of UIMS include the conceptual architecture, the implementation techniques and the support infrastructure

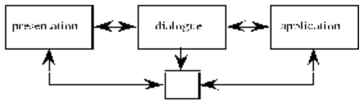
UIMS as conceptual architecture

The conceptual architecture is viewed as the s*eparation between application semantics and presentation. This improves:*

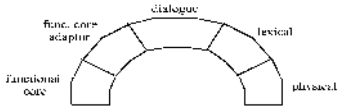
- *v. Portability which runs on different systems*
- *vi. Reusability having components reused thereby cutting costs*
- *vii. Multiple interfaces that access same functionality*
- *viii. Customizability; here, the system is customised to suit the designer and user.*

The User Interface Management System (UIMS) tradition of interface layers and logical components comprise

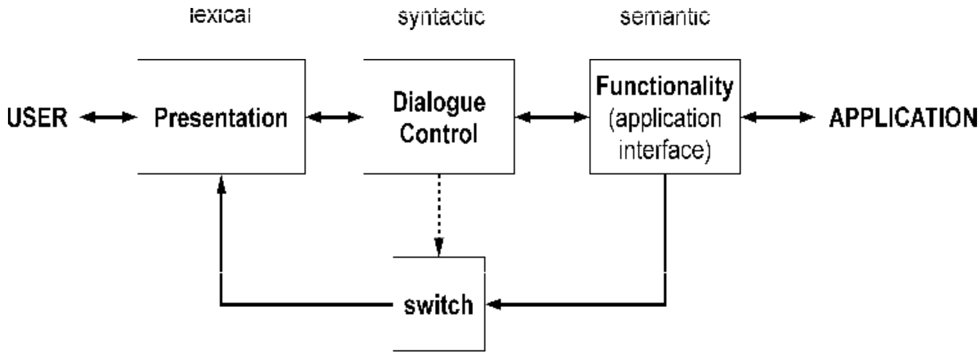
- linguistic: lexical/syntactic/semantic
- Seeheim:



- Arch/Slinky



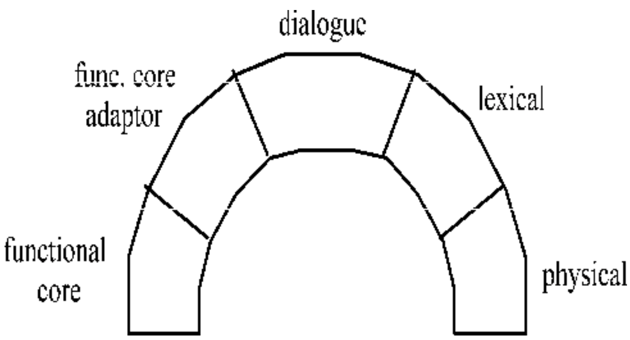
The Seeheim model: Its concept and implementation
Seeheim concept arose as a result of implementation experience with conceptua l approach as principal contribution. The concepts are part f the normal user interface language.



The above depicts different kinds offeedback. For example, the movement of th e mouse carried out at the presentation interface is known as th lexical feedback, the menu highlights as adialogue control is known as the syntactic feedback while a fun ction carried out at the application interface such as sum of number changing ,is regarded as the semanti c feedback. Because the semantic feedback is oft en slower, programmers prefer to use the r pid lexical and/or the syntactic feedbacks.
The lower box representing the bypa ss switch is needed for implementation. The switch enables a direct communication between application and presentation. Though regulated by a di logue control, it also provides a rapid semantic feedback.

The Arch/Slinky model characteristic s

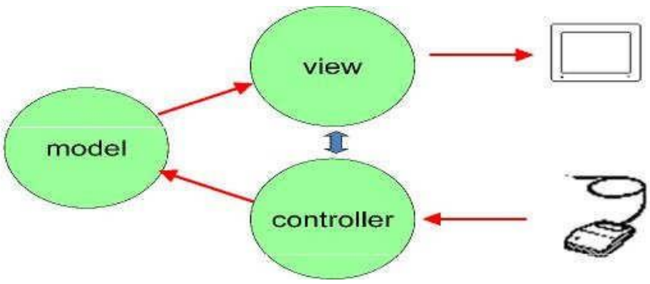
- This model contains more lay ers to distinguish the lexical and the physica l
- Like a 'slinky' spring, differen layers may be thicker (that is more import nt) in different systems or in different components



Monolithic vs. Components
Seeheim has big components and is ften easier to use smaller ones especially. if using object-oriented toolkits .

Smalltalk used the model—view—cont roller (MVC)
Model indicates the internal logical state of component
View shows how it is rendered on sc een
Controller processes user input

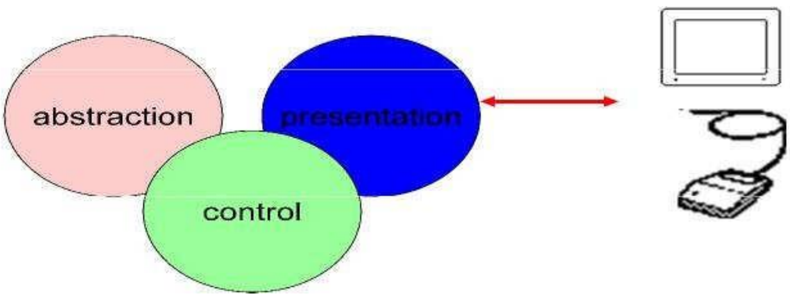
The Model - View - Controller (MVC)



Input ・ control ・ model ・ view・ outputMVC

issues
MVC is largely pipeline model:
Using the pictorial illustration above, the Input is transmitted to the controller, th e controller processes user input and connects with the mo del. Since the model represents the internal the logical state of the component, a manipulation is carrie out and through the view model, the result of the manipulation is output on screen. In graphical interfa ce, input only has meaning in relation to ou put e.g. a mouse click There is the need to know *what* was clicked and the controller has to decide wha t to do with the click. Using the internal logical state of the component through the model, the view di plays how it is rendered on the screen.
However, in practice, the controller irectly 'talks' to view.

The Presentation, Abstraction and C ntrol (PAC) model
presentation - abstraction - control



The PAC model is closer to the Seehe im model in principle because the term 'Pre sentation' describes how input and output are managed, the ' Abstraction' describes the logical state of th component while 'Control' represents the state of me diation between the 'Presentation' and 'Abs raction' .
The PAC model manages the hierarc y and multiple views through the control pa rt of PAC communicate the 'Abstraction'
Though the PAC model is direct, the MVC model is used more in practice as can b e found in the use of Java Swing.

The Implementation of UIMS takes t he following forms:
Implementing the techniques for dia logue controller through the use of the

- Menu networks
- State transition diagrams
- Grammar notations
- Event languages
- Declarative languages
- Constraints
- Graphical specification

4.0 CONCLUSION

Programming tools for the design of user interface enable easy programming of the application and the control of dialogue between the system and the user.

5.0 SUMMARY

Levels of programming support tools comprise Windowing systems that are device independence with multiple tasks, the Paradigms for programming the application with read-evaluation loop that is notification-based , Toolkits containing programming interaction objects, and the User Interface Management System made up of conceptual architectures for separation together with the techniques for expressing the dialogue .

The layers of programming development tools contribute to how human computer interaction affects the programmer.

6.0 Tutor Marked Assignment

1 (a) Explain the two elements of the windowing systems

(b)Using the pictorial representation of a windowing system, describe its role in interactive programming

9. Describe the architecture of a Windowing System using the client server architecture diagram.
10. What are the functions of the User interface toolkits?
11. In what ways are widgets used to support prototyping?
12. What is the alternative to the use of a took-kit? What could require the use of such alternative?
13. Using a suitable pictorial representation, describe the concept and implementation of the Seeheim model
14. What is the concept supporting the model 'View Controller'(MVC)? How useful is this concept to the programmer?
15. Explain how closely is the relationship between the MVC model and the Presentation, Abstraction and Control (PAC) model?
16. In what form can the implementation of the User Interface Management System be carried out?

7.0 Further Readings / References

- Buxton, W., *et al.* "Towards a Comprehensive User Interface Management System," in *Proceedings SIGGRAPH'83: Computer Graphics.* 198 3. Detroit, Mich. 17. pp. 35-42.
- Kasik, D.J. "A User Interface Management System," in *Proceedings S IGGRAPH'82: Computer Graphics.* 1982. Boston, MA. 16. pp. 99-106.
- Scheifler, R.W. and Gettys, J., "The X Window System." *ACM Transactions on Graphics*, 1986. 5(2): pp. 79-109.
- Myers, B.A., "User Interface Software Tools." *ACM Transactions on Computer Human Interaction*, 1995. 2(1): pp. 64-10 3.

- Palay, A.J., *et al.* "The Andrew Toolkit - An Overview," in *Proceedings Winter Usenix Technical Conference.* 1988. Dallas, Tex. pp. 9-21.

UNIT 4: THE SOFTWARE DESIGN PROCESS OF HUMAN COMPUTER INTERACTION

Table of contents

1.0 INTRODUCTION

Software engineering is the discipline for understanding the software design process, or life cycle. Therefore, this unit looks at the software design process of human computer interaction by analysing the pros (linearity) and cons (non linearity) of the water fall model that comprises the design life cycle. The usability engineering process that measures the user's experiences is weighed against the ISO usability standards 9241. For a successful and effective design, management issues concerned with interactive design and prototyping are considered along the relevant design rationale.

2.0 OBJECTIVES

By the end of this unit, you should be able to:

- Understand the distinct activities that constitute the software engineering life cycle
- Understand the concepts of the water fall model in relationship to the software life cycle
- Differentiate between software engineering life cycle usability engineering
- Explain the concepts of interactive design and prototyping

.0 MAIN CONTENT

3.1 The software process of Human Computer Interaction

The software process comprises the following:

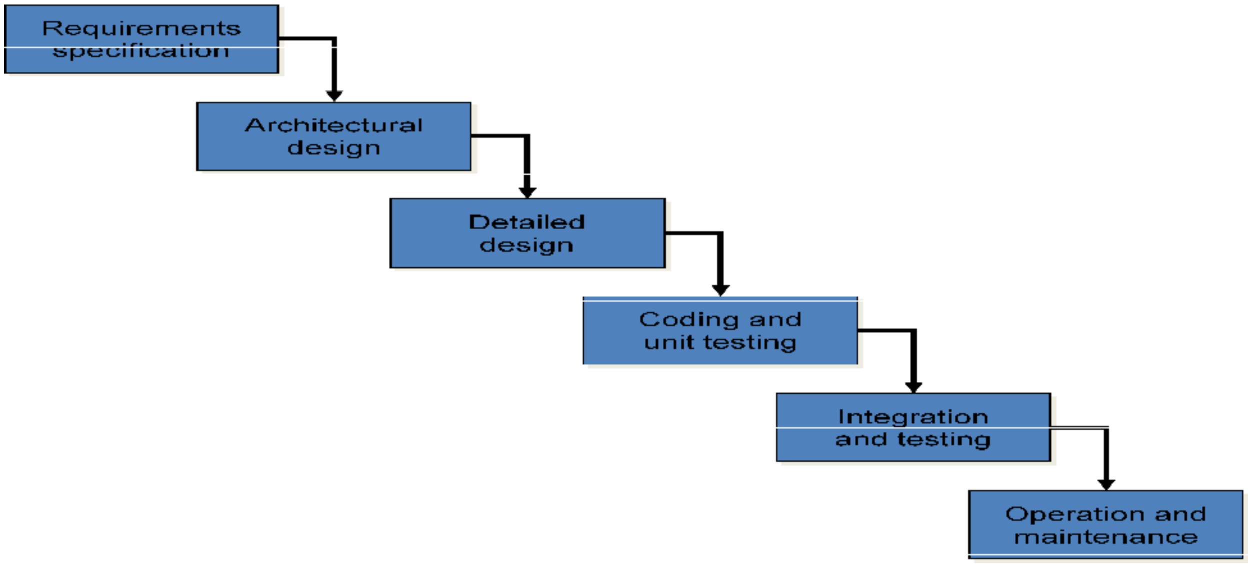
- Software engineering and the design process for interactive systems
- Usability engineering
- Iterative design and prototyping
- Recording the design knowledge using the design rationale

Software engineering is the discipline for understanding the software design process, or life cycle. The design for usability occurs at all stages of the life cycle, not as a single isolated activity.
Usability engineering is the ultimate test of usability based on measurement of user experience.
Iterative design and prototyping overcomes inherent problems of incomplete requirements

Design rationale is information that xplains why a computer system is the way it is.

.2 The waterfall model

The waterfall model depicts the software life cycle.
The pictorial illustration that follows reflects a mountain top from where water f lls towards the bottom of the mountain, hence called a waterf ll. It shows the commencement of the life cycle (the requirements specification) through the design, co ding and testing processes to its ultimate ter mination (the operation and maintenance).



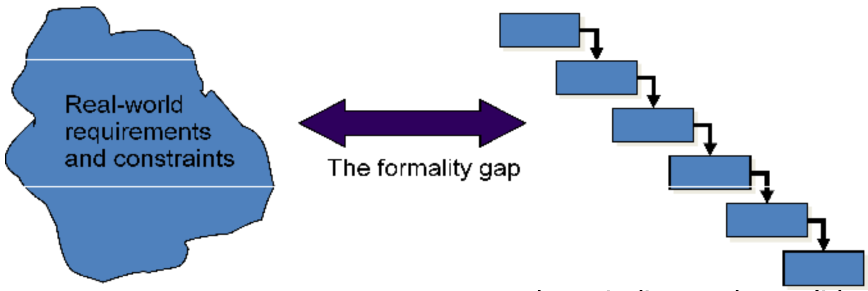
Activities in the software lifecycle ar :

Requirements specification:
Here, the designer and client try to c apture what the system is expected to provi de and can be expressed in natural language or more precise l anguages, such as a task analysis would prov ide.

Architectural design:
This is a high-level description of ho the system will provide the services requir d. It describes how to d. factor system into major component s of the system and how they are interrelate It shows the needs to satisfy both functional and non-funct ional requirements

Detailed design
This concerns a refinement of archit ctural components and their interrelations to identify modules to implemented separately. The refine ent is governed by the non-functional requ be rements.

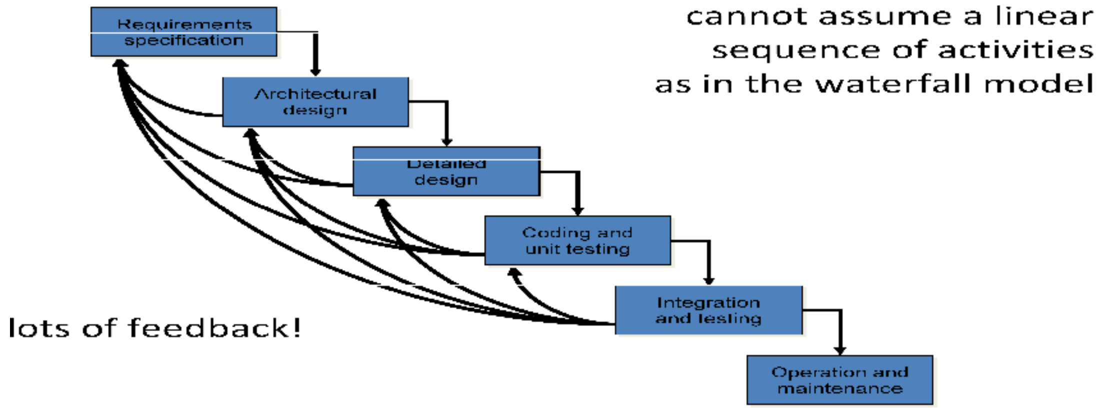
Verification and Validation
Verification: This is ensuring that the product is designed right
Validation: This is ensuring that the right product is designed.



The formality gap shown in the diagr am above indicates that validation will alwa s rely to some extent on subjective means of proof.

Management and contractual issues describes design in commercial and legal co ntexts

. 3 The life cycle for interactive s ystems



The life cycle for interactive systems cannot assume a linear sequence of activitie s as in the waterfall model because there are lots of feed backs occurring within the initial requireme ts specification, the designs, the coding and testing proce sses. See the illustration above.

3.4 Usability engineering:

This is the ultimate test of usability based on measurement of user experience. Usability engineering demands that specific usability measures be made explicit as requirements.

Usability specification comprises the usability attribute and/or principle, the mea suring concept and the measuring method. It also depicts either the present level the worst case level, t e planned level or the best case level.

The problems associated with usabili ty specifications are:

i. Usability specification requires lev el of detail that may not be possible in arly life of the design process.

ii. It does not necessarily satisfy usab the e ility.

Example of a usability specification

Attribute: Backward recoverability

Measuring concept: Undo an errone us programming sequence

Measuring method: Number of expli cit user actions to undo current program

Present level: An undo is not allowe presently

Worst case: This considers as many a ctions as it takes to program in mistakes

Planned level: A maximum of two ex plicit user actions are allowed

Best case: One explicit cancel action is allowed

3.5 ISO usability standard 9241:

ISO usability standard 9241 adopts the following traditional usability categories:Effectiveness: This is achieving what you want to.

Efficiency: This is doing it without wasting effort.

Satisfaction: This is showing whether or not you enjoy the process.

Some metrics from ISO 9241 are :

Usability objective	Effectiveness measures	Efficiency measures	Satisfaction measures
Suitability for the task	Percentage of goals achieved	Time to complete a task	Rating scale for satisfaction
Appropriate for trained users	Number of power features used	Relative efficiency compared with an expert user	Rating scale for satisfaction with power features
Learnability	Percentage of functions learned	Time to learn criterion	Rating scale for ease of learning
Error tolerance	Percentage of errors corrected successfully	Time spent on correcting errors	Rating scale for error handling

3.6 Iterative design and Prototyping

Iterative design overcomes inherent problems of incomplete requirements while prototypes simulate or animate some features of intended system.

Different types of prototypes can be identified as: (i) throw-away, (ii), incremental and (iii) evolutionary

Management issues concerned with Interactive design and Prototyping:

The management issues are:

- Time allocated to the design,
- Planning the design,
- Non-functional features of the design and
- Contracts of the design.

Prototyping Techniques:

These include the following:

- Storyboards which need not be computer-based but can be animated
- Limited functionality simulations in which some part of the system functionality is provided by designers. Such tools like HyperCard are common.
- Wizard of Oz technique
- Warning about the iterative design. These show concerns about the design inertia in which early bad decisions stay bad. It is better to diagnose the real usability problems in prototypes; and not just the symptoms.

4.0 CONCLUSION

The life cycle for interactive systems cannot assume a linear sequence of activities as in the waterfall model because there are lots of feedbacks!

5.0 SUMMARY

The software engineering life cycle consists of distinct activities and the consequences for interactive system design. The waterfall model depicts the software life cycle.
Usability engineering makes usability measurements explicit as requirements and is the ultimate test of usability based on measurement of user experience.
Iterative design overcomes inherent problems of incomplete requirements.
Prototyping techniques comprise storyboards, limited functionality simulations or animations of intended system, and Wizards.
Design rationale records design knowledge through the process and structure orientations.

6.0 Tutor Marked Assignment

1. Briefly explain the four tasks of the software process of Human Computer Interaction.
2. (a) What do you understand by the term "the Waterfall model" in the software design process of human computer interaction?
   (b) With the aid of a related diagram, briefly explain the activities depicted by the "waterfall model"
3. What do you observe as the major departure of the interactive system life cycle from the "waterfall model"? What do you think as the reason behind this major departure?
4. (a) Explain the value of usability engineering in the Software design process
   (b) What are the major problems associated with usability specifications? ©
   Give any three examples of a usability specification
5. What are the usability objectives of the International Standards Organisation (ISO) 9241? Mention its usability categories
6. (a) Differentiate between the objectives of interactive design and prototyping. (b) What are the management issues concerned with interactive design and prototyping?

14.0 Further Readings / References

- Swinehart, D., *et al.*, "A Structural View of the Cedar Programming Environment." *ACM Transactions on Programming Languages and Systems*, 1986. 8(4): pp. 419-490.
- Swinehart, D.C., *Copilot: A Multiple Process Approach to Interactive Programming Systems.* PhD Thesis, Computer Science Department Stanford University, 1974, SAIL Memo AIM-2 30 and CSD Report STAN-CS-74-412.
- Teitelman, W., "A Display Oriented Programmer's Assistant." *International Journal of Man-Machine Studies*, 1979. 11: pp. 157-187. Also Xerox PARC Technical Report CSL-77- 3, Palo Alto, CA, March 8, 1977.
- Newman, W.M. "A System for Interactive Graphical Programming," in *AFIPS Spring Joint Computer Conference.* 1968. 28. pp. 47-54.
- Shneiderman, B., "Direct Manipulation: A Step Beyond Programming Languages." *IEEE Computer*, 198 3. 16(8): pp. 57-69.
- Smith, D.C., *Pygmalion: A Computer Program to Model and Stimulate Creative Thought.* 1977, Basel, Stuttgart: Birkhauser Verlag. PhD Thesis, Stanford University Computer Science Department, 1975.

UNIT 5: INTERACTIONS IN HYPERTEXT, MULTIMEDIA AND THE WORLD WIDE WEB

Table of contents

HUMAN COMPUTER INTERACTION IN HYPERTEXT, MULTIMEDIA AND THE WORLD-WIDE WEB

1.0 INTRODUCTION

Understanding hypertext

It enables you to find information by navigating the hyperspace using the web technology

Hypermedia is not just text but hypertext systems containing additional media such as illustrations, photographs, video and sound.

The web contains protocols, browsers, web servers, clients and a lot of networking. The challenges remain a loss in hyperspace and information overload.

The advantage of this option is an interactive Data Base access. The availability of bandwidth and the security of data are problem issues to be resolved.

2.0 OBJECTIVES

Understanding hypertext

Multimedia or Hypermedia

Animation

Video and Audio effects.

Web technology issues:

Network issues

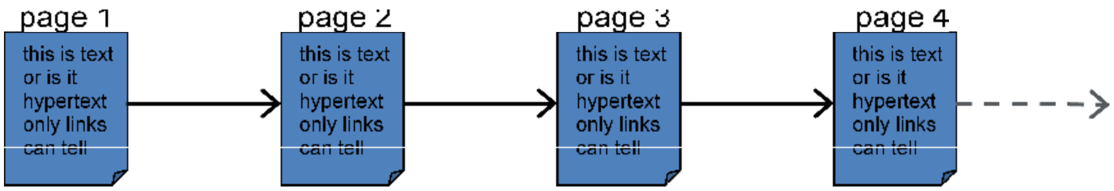
The web content could be made stat ic by unchanging pictures and text or made ynamic with interaction and applications on the web.

.0 MAIN CONTENT

3.1 Understanding hypertext

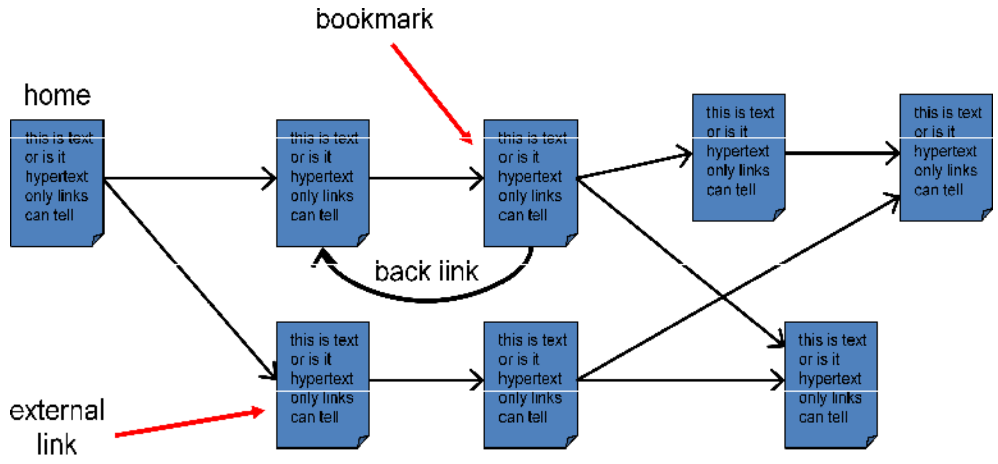
Hypertext enables you to find inform ation by navigating the hyperspace using th web technology Hypertext is made up of rich content of graphics, audio, video, computation and nteraction.

Text



In some cases hypertext imposes stri ct linear progression on the reader but in m st cases it is not linear as shown in the diagram below:

Hypertext - not just linear



From the above, we can see that the non-linear structure contains blocks of text (pages) with links between pages that create a mesh o r network. The users follow their own path f r the information desired.

3.2 Multimedia or Hypermedia

Multimedia is sometimes also called Hypermedia. The term can also be used forimple audio and video effects.

Hypermedia is not just text but hype rtext systems containing additional media su ch as illustrations, photographs, video and sound.

Links and hotspots may be in media with their pictures, the times and locations i video.

Animation

Animation is adding motion to image s particularly images on things that change i n time. Examples are digital faces that take seconds to tick past or warp into the next and analogue fac e with hands swept around the clock face.

Animation comprises live displays fo r showing status and progress, flashing caratat text entry location, busy cursors (in form of hour-glass, c lock, and spinning disc) and progress bars

Animations are used for education a
interesting and entertaining images.
Used for data visualisation by creatin
visualisation is done using animated
In science, complex molecules and th
and viewed on the screen.
Animated characters are useful in wi

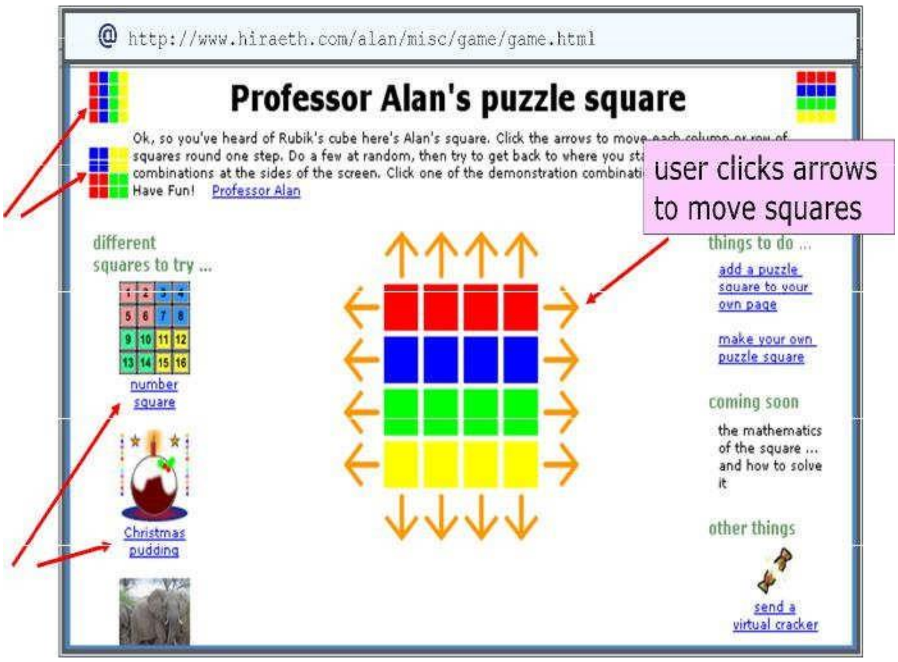
Video and Audio effects.
The current technology on improved
Tools are now available to edit soun
They are easy to embed in web page
The memory occupied could be man
It can however affect the download t
and sometimes frustrating.
It may be hard to add 'links' in hyper

Using animation and video
Animations and videos are potentiall
arcade games.
But how do we harness the full possi
In order to gain more experience fro
theorists, cartoonists, artists, and wr

3. 3 Interacting in hypertext.
Using the computer for processes, h
Illustrative interactions:

x. We search for a particula
books, look for the books'
puzzle square). The e-com

xi. Professor Alan's puzzle sq



xii. Delivery technology

On the computer, the help systems a hypermedia

The same information can be obtain documentation. These applications c the internet using mobile platforms s computers.

WiFi access points or mobile phone n where

xiii. Tourist guides and directe

xiv. Rapid prototyping, creati

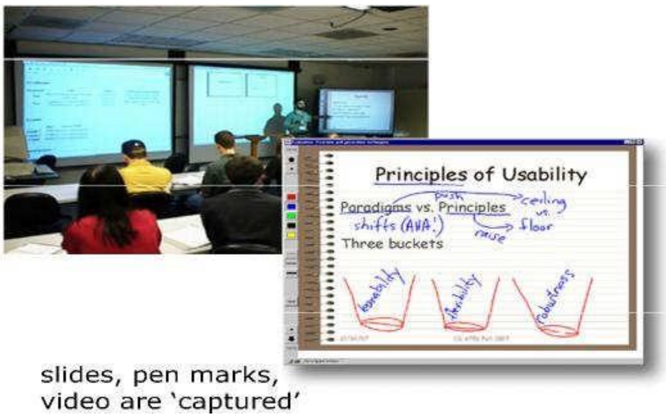
xv. The help and documentat

xvi. 'Just in time learning' (wh for a photocopier with te web. You can as well obta

xvii. In education, animation a atmosphere, and diagram structure of the web allo

xviii. E-learning provides educa

An e-Class is shown in the picture be



Lost ness in hyperspace

To avoid getting lost in hyperspace w history, bookmarks, indices, director The non-linear structure of the web i There are two aspects of lost ness: C also a lack of information integration Good design helps navigation and str

3.4 Designing structure:

Designing good structure entails task structures. In making navigation easi current location.

Also provide recommended routes as guided tour or bus tour metaphor and as a linear path through a non-linear structure.

To support printing your navigated information, you need a linearised content that links back to source.

History and bookmarks also ease navigation because they allow 'hub and spoke' access with lots of revisiting of pages. Bookmarks and favourites are good for longer term revisiting. Frames are difficult to bookmark, search and link to, except there are good reasons for its use, it is not recommended.

Using indices, directories and search

Indexes are often found on help, documentation and books. Selective but non-exhaustive list of words are used in index. Directories on web index would be very huge; so, manually choose site of navigation e.g. an open directory project or Yahoo. Using web search engines make you 'crawl' the web by following links ' from page to page. Search engines build full word index but ignore common 'sto words to carry out its search. It looks up your request in in ex when you enter keywords to find the pages.

3.5 Conducting complex search:

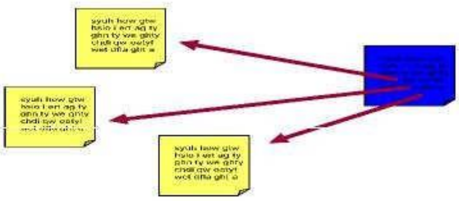
Conducting a complex search involve too many pages for a single word search, there is a need to be more selective. You can use a Boolean search method that combines words with logic e.g. if you want to find facts about engine only, write 'engin AND NOT car' . 'AND NOT 'is the Boolean expression.

In creating a link structure, Goggle, a search engine, uses richness of in and out links to rank pages.

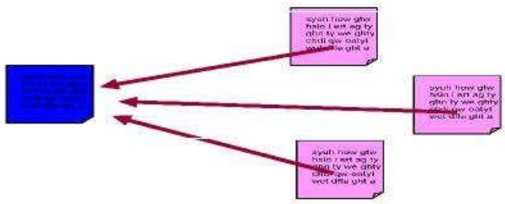
To be search engine friendly, add 'Meta' tags, relevant title, keywords and description. Note that it is hard to index generated pages for a hidden web.

Finding research literature involves special portals and search sites such as citeseer<citeseer.nj.nec.com>.

Searches for literature papers require scanning the papers for bibliography to build up citation index, such as in the diagrams below.



bibliography backwards in time



citation forwards in time

3.6 Web technology and web isses:

The web contains protocols, browses, web servers, clients and a lot of networki s g.

Web activities involve using protocoland standards. Protocols such as HTTP(Hy r pertext Transmission Protocol) that carry information ovthe internet, HTML (Hyper-Text Mark-up Language) , XML (Extended Mark-up Language) and gaphics formats for content browsers to view the results, and a lot of plug-ins.

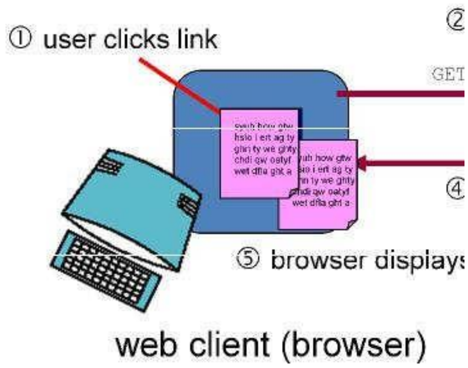
Changing use of the Internet:

Initially, the internet was created for research by CERN for their high energy physics research. But all over the world, it is now used by corporate, government, commerce, entertainment, and the advertising community.

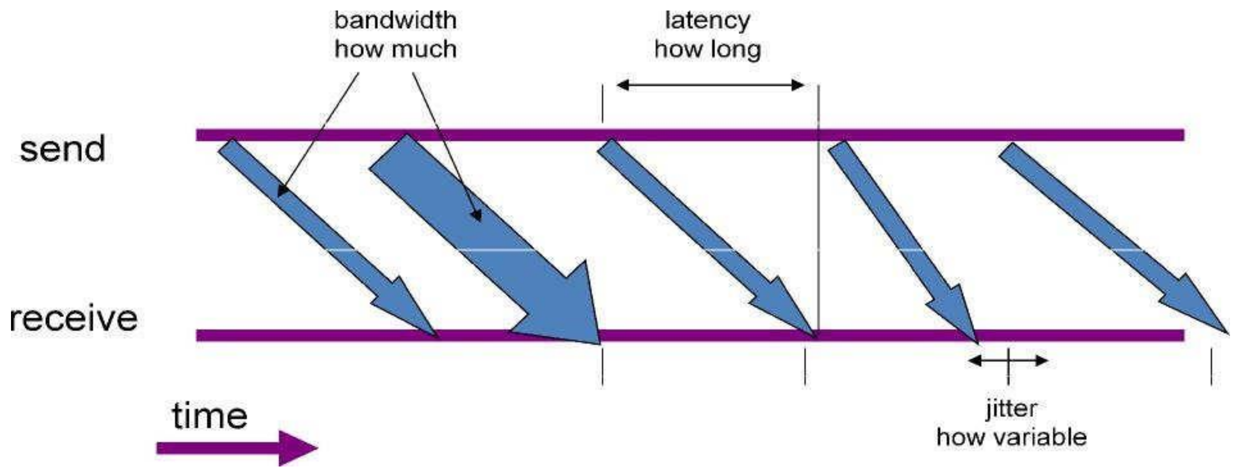
The challenges remain a loss in hyperspace and information overload.

Web servers and clients:

The web is distributed with different machines far across the world. Pages are stored on servers, the browsers (the clients) ask for pages that are sent to and from across the internet as illustrated in the picture below:

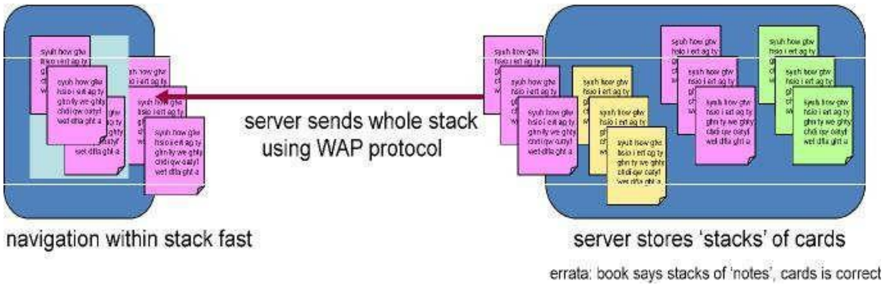


3.7 Network issues — Timing and volume of data transmission
QOS (quality of service): This term dscribes the quality of service provided by th e network.
The following comprise the quality oF service:-
Bandwidth: This is how much of infomation transmitted per second
Latency: This is how long the transmission takes (otherwise called delay)
Jitter: This is how consistent the delay is
Reliability: Here some messages may be lost and needed to be resent. This increa ses jitter and the connection set-up, hence a need to ' handshake' to start.
The illustration of bandwidth, latenc and jitter is given in the accompanying diag ram below.

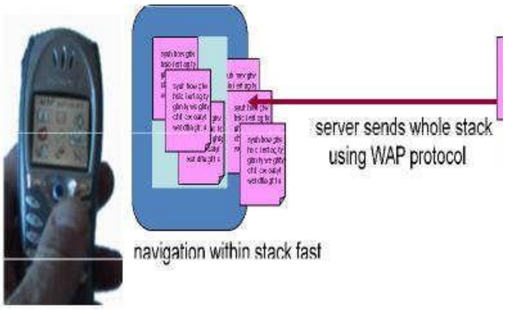


3.8 Design implications of the W b:
While designing, you should considethe bandwidth and hence about the downl oad time, e.g. an image of 100K bytes may be transmitted in 1 s ec (also called broadband), and a 56K bytes modem may transmit in 18 seconds. Hence the need to save graphics in appropriate format and size at the same graphics could be reused in the browser cach so th e after first load.
To reduce the connection time, use ne big file at a time in a data transfer than u sing several small ones.
Beware of 'fit on one screen' rule be cause scrolling is fast! Think before breaking big graphic into bits.
In latency, think about feedback.

3.8.1 WAP (Wireless Advance Prot col): This describes the web activities on th e phone as illustrated in the diagram below:



The phone is made up of very small screen and the scrolling of data could be very painful because the screen displays small 'pages' at a time. A GSM connection is considerably slow with big chunks. Operation is carried out using a WML (wirelessmark-up language) whose content is delivered in 'stacks' of 'cards'. The cards are the 'pages' the user viws but navigation within the stack is fast.
N.B. With larger screens and faster connections, WML (Wireless Mark-up Language) is giving way to small HTML pages. See illustration below.



3.8.2 Static web content:
This is a medium and a message comprising text, graphics, movies and sound.
The message and the medium "content is king" because it is the catch phrase of the dot.com era but widely ignored.
The message content should be appropriate to the audience, should be timely, reliable, and generally worth reading!
The medium page and site design should be a good design that is essential to attract readers. A bad design may mean good material that is never seen.
The content should be printable!

3.8. 3 Text:
The text style should be universal generic styles such as serif, sans, fixed, bold and italic. You can use specific fonts too, but these should vary between platforms.
Use cascading style sheets (CSS) for fine control but beware of older browsers and fixed font sizes for compatibility.
Because colours are often abused, b careful about your choice of colour.
Text positioning should be easy; it could be left, right justified or centred. You should remember that DHTML (Dynamic Hypertext Mark-up Language) requires precise positioning so beware of platforms.
You should also take note of the screen size.
Remember that mathematical oriented texts require special fonts and layout.

8.4 Graphics:

Graphics should be used with care considering for example, the file size and download time. For example, the image above is made up of 1000 words of text and is affected by size, number of colours, and file format.
Add little backgrounds because too any backgrounds often make text hard to read.
Speeding up transmission of graphic require caching and to be able to reuse the same graphics.

Using progressive formats make the image to appear in low resolution and the image also gets clearer as shown above.

8.5 Formats:

Use JPEG for photographs as shown bove, for higher but 'lossy' compression.
Use GIF (Graphics Image Format) for sharp edges and lossless compression.
PNG formats are supported by current web browsers.
For action, use animated gifs for simple animations and image maps for images you can click on.

3.8.6 Icons:
Sample icons are displayed below

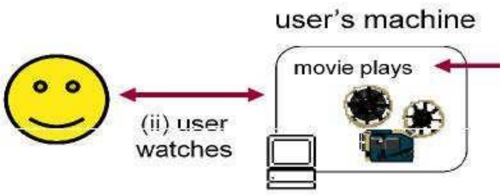
They are just small images on the web used for bullets and decoration or to link to other pages.
There are lots available!
The design of icons is just like any other interface that needs to be understood. So icons should be designed as collection to fit the web. A web site under construction is a sign of the inherent incompleteness of the web.

3.8.7 Web colour:
Decide from the beginning how many colours your web site is to contain. The PCmonitor is made to display millions of colours each comrising 24 bits per pixel but the 'same' colour may look very different.
Web colours are usually expressed a dips per inch —dpi. Between 72 to 96 dpi ae common.
Older computers, PDAs and phones can tolerate perhaps only 16 bits or 8 bits per pixel in 256 colours or even in greyscale.
From the colour palettes, you can choose up to 256 different useful colours, although Netscape 'web safe' 216 colours are common. Each GIF (raphics Image Format) used for fast download has its own palette.

3.8.8 Movies and sound

The problems of size and download a re worse in movies and sound compared wi th graphics. So they require special plug-ins. The problems are not prominent with audio how ever, because some have some compact formats such as MIDI.

With streaming video, you can play hile downloading hence can be used for 'br oadcast' radio or TV (see illustration below).



Dynamic web content

This content shows what happens w ere, with its technology and security, it ena bles a local interaction, and effective search through remote access and batch generation of pages of dat a. It is of dynamic content.

The active web

In the early days of the web, pages t at contained text were static pages. Some gateways such as FTP, GOPHER enabled usability thereby m aking interaction easy on one simple model.

A dynamic content has a model or m etaphor with passive pages but active interf ce. Each leads to different user understanding.

The architectural design of the web is about what happens where. The design has (i) a feedback in which a user can see results of his own actios, (ii) a feed through effect in which effects of other people's actions are seen. Note that the effect of the design is reflected on the complexity of imp lementation and its maintenance.

The concerns of the designer from th e user's point of view are the changes to be made in the design during use in terms of the media stream, th e presentation, and the content. He should be guided by the following questions:

i. 'Are these done automatically?', ii. ' By whom would the changes be made?' - The site author, the user, or other users through a feed through? and iii. How often is the pace of change: in days, in months, or in seconds?

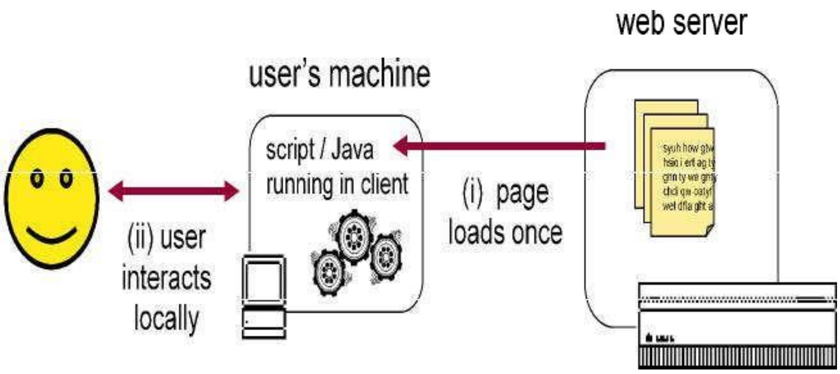
The technology of design changes:

Where does the change happen? Is i through the client using toolkits such as the applets , Flash, JavaScript and DHTML ,or through the server using server toolkits such as the CGI scripts, J va servlets , JSP, ASP, PHP, etc,? Or through another machine such as the author's machine, the database server, the proxy server, or through people adopting s ocio-technical solutions?

Security of the web:

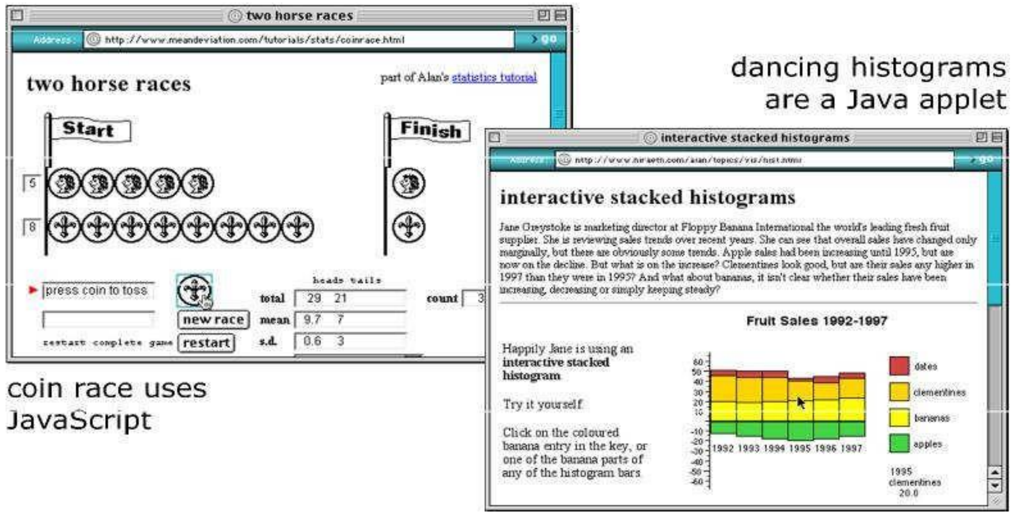
For computational functions, the codes and data should not be at the same place! The problems that need to be addressed on security include the security of data, the safety of the web-server and the client machine that is the most vulnerable, and of course the entire networks.

Local interaction at the client side: S e the illustration below



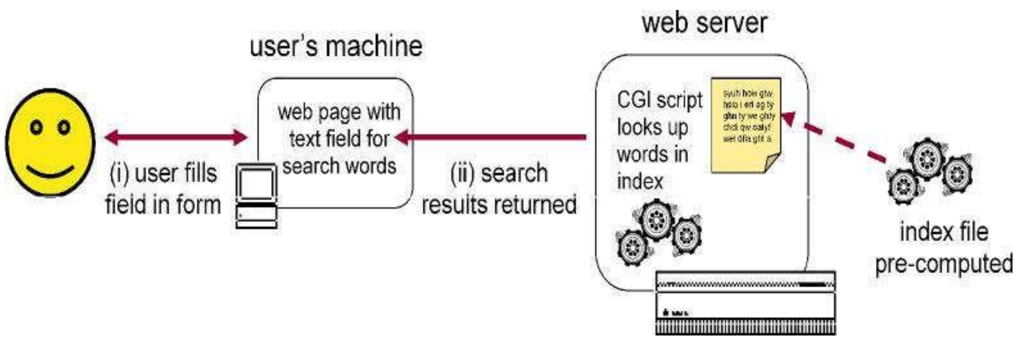
For a fixed content interaction, the u ser interacts locally through his machine by he use of Java applets, Flash, JavaScript plus DHTML (Dynam ic Hyper-Text Mark-up Language).
In fixed content interaction, a rapid f eedback occurs but interaction is local and t here is no feed through.

examples



The picture above shows that the 'co in race' uses Java Script while the 'dancing' istograms (set of interactive stacked histograms) that depict the sales trends for each fruit type, ar e Java applets.

The picture below demonstrates the processes of the user conducting a web sea ch.

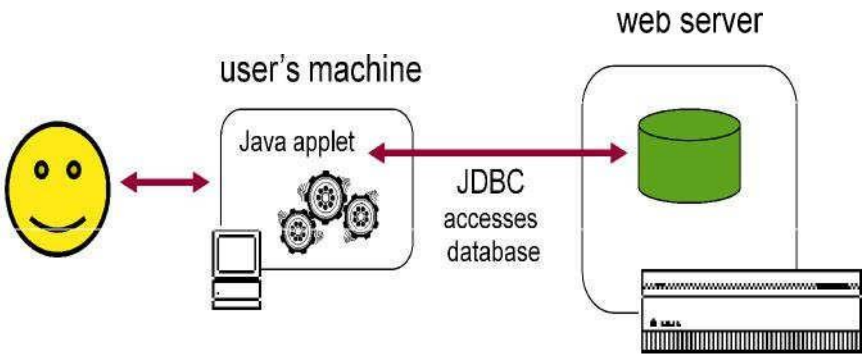


Indices are created off-line before co mmencement of search.

For automatic generation of data, yo u need database driven sites. The available ptions of tools that can be used are the client-end applet or Flash access remote DB using the server-end CGI that is driven by web forms with limited user interface.

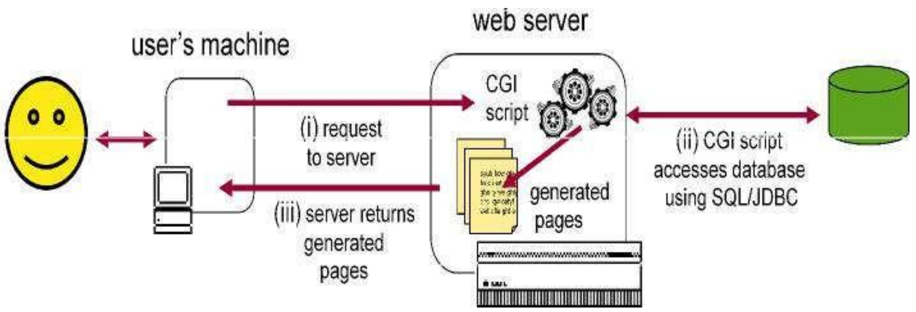
However, hybrid solutions that could be optionally utilised are the CGI generated pages that can contain JavaScript etc. The JavaScript can 'wr ite' web pages dynamically.

Look at the following picture: A situa tion of the Java applet and JDBC accessing t e database.
Java applet & JDBC



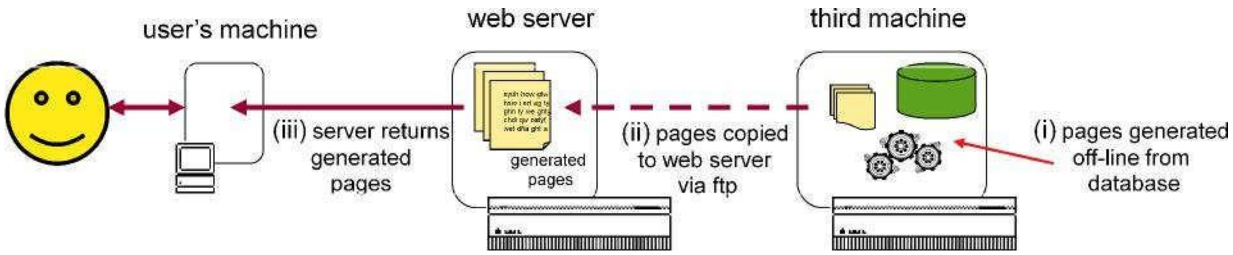
The advantage of this option is an int eractive Data Base access. However, the ava ilability of bandwidth and the security of data are problem issu es to be resolved.

The picture below depicts the situati on in which the CGI script accesses the data ase.



The advantage here is that the datab ase is always current with up to date inform ation.
Its disadvantage is the non proximity of the web server and that access is not ind ex friendly.

Batch generation of web pages of data.
Batch generation of data is for slow a nd varying data that is updated through a lo cal database. Here, pages are periodically generated before up load.
Many technologies are involved in b tch generation; and they include the use of object oriented languages such as C, Java, HyperCard and Visua l Basic.
Illustration of batch generation of w b pages



Advantages of this option are that th e generated batches are index able and secu re.
The disadvantage is its slower turnar ound.

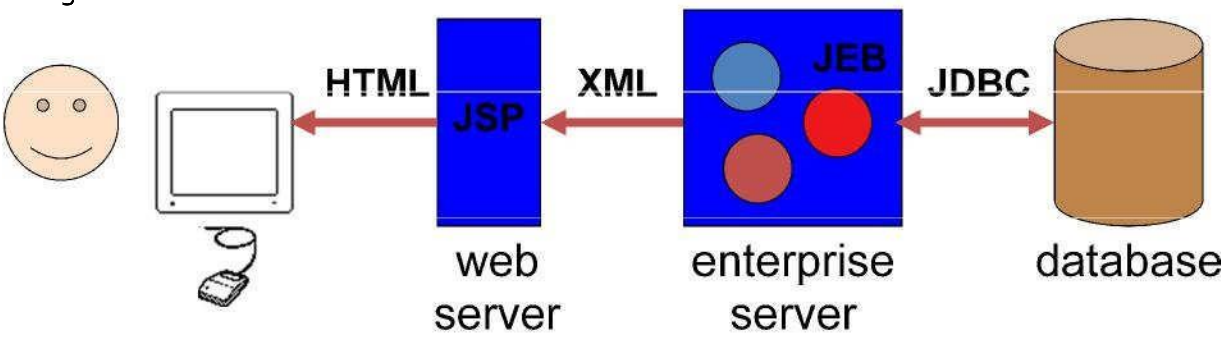
Dynamic content:

Dynamic contents are really 'active'web pages in which data is updated as well a s presented on the web. The data presentation could be made in any of the means discussed previously u sing the CGI, applet-JDBC etc.

The update is done through the webform interface using the server script that updates the data base, as in the example of that carried out in th e updating of book theatre seats.

The issues of concern in the design of this type of interaction are the authenticat on and security problems due to multiple transactions of data.

Using the n-tier architecture



As illustrated, the picture contains òe or more intermediate layers with theiness logic' in layers. The web is made up of standard com 'bus ponents and protocols.

4.0 CONCLUSION

The challenges of human computer i nteraction in hypertext, multimedia and the world-wide web remain a loss in hyperspace and information overload.

Also, the availability of bandwidth an d the security of data are problem issues to be resolved

5.0 SUMMARY

Hypertext enables you to find inform ation by navigating the hyperspace using th web technology

Multimedia is sometimes called Hypermedia and can be used for simple audio an d video effects. ch

Hypermedia is not just text but hype rtext systems containing additional media su as illustrations, photographs, video and sound.

Animation is adding motion to image s, particularly images on things that change in time.

History and bookmarks ease navigati on because they allow 'hub and spoke' acce s with lots of revisiting of pages. Bookmarks and favourites aregood for longer term revisiting

The web contains protocols, browser s, web servers, clients and a lot of networki ıg.

Web activities involve using protocols and standards. Protocols such as HTTP(Hy pertext Transmission Protocol) that carry information ovr the internet, HTML (Hyper-Text Mark-up Language) , XML (Extended Mark-up Language) and gaphics formats for content browsers to view the results; and a lot of plug-ins.

Wireless Advance Protocol (WAP) de scribes the web activities on the phone

Web icons are small images on the w eb used for bullets and decoration or to link to other pages.

Dynamic content are 'active' webges in which data is updated as well as prese nted on the web.

Batch generation of web pages ofta is for slow and varying data that is update d through a local da database

6.0 Tutor Marked Assignment

9 Distinguish between: (i) Hypertext and Hyper media, (ii) Hypermedia and Ani mation effects

10 What do you understand by "Vid eo and Audio" effects in human computer in teraction? What are their advantages and challenges arisin g from their application in web browsing and hypertext?

11 Give 5 examples of typical Human Computer interactions in the web.
12 What is the function of the Boolean search used in conducting a web search? Give 2 examples of Boolean expressions that can be used for a web search.
1 3 Describe the major components of the web technology
14 The quality of Service (Q0S) of a network is reflected in the bandwidth, latency, the jitter, and reliability of the network. What do you understand of the underlined terms?
15 Mention 2 design implications of the web. Describe how these affect the web activities using the GSM phone
16 What are the design implications to be considered in the following:
(i) Static web content, (ii)Text, (iii) Graphics, (iv)Picture formats, (v)Icons, (vi)Web colour, (vii) Movies and sound,(vii) the active web.
9. Distinguish between a fixed content interaction in the web and a dynamic interaction. What are the resource software applications utilized to carry out these interactions?
10. Using a pictorial illustration how is the batch generation of web pages carried out? What are the technologies utilized to carry out this function? What are the advantage(s) and disadvantage(s) of this option of generating data?

7.0 Further Readings / References

- Catledge, 1-ara D. and James E. Pitkow. "Characterizing Browsing Strategies in the World-Wide Web." Unpublished (?), 1994. [give UR1-?]
- Nielsen, Jakob. "Using link titles to help users predict where they are going." *Alertbox* column of January 11, 1998.
- Nielsen, Jakob. "The difference between Web design and GUI design." *Alertbox* column of May 1, 1997.
- Nielsen, Jakob. "The Tyranny of the Page: continued lack of decent navigation support in Version 4 browsers." *Alertbox* column of November 1, 1997.
- Nielsen, J., *Multimedia and Hypertext: the Internet and Beyond.* 1995, Boston: Academic Press Professional.
- van Dam, A., *et al.* "A Hypertext Editing System for the 360," in *Proceedings Conference in Computer Graphics.* 1969. University of Illinois.