MODULE 1: SURVEY OF HUMAN-COMPUTER INTERACTION

CONCEPTS, THEORIES AND PRACTICE

UNIT 1 CONCEPTS, THEORIES AND HISTORY

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

3.0 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. 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 the user clicks on the "delete" item (interpretation).

The user view of interaction technique:
Here, an interaction technique is a way). 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 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.

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 comuting

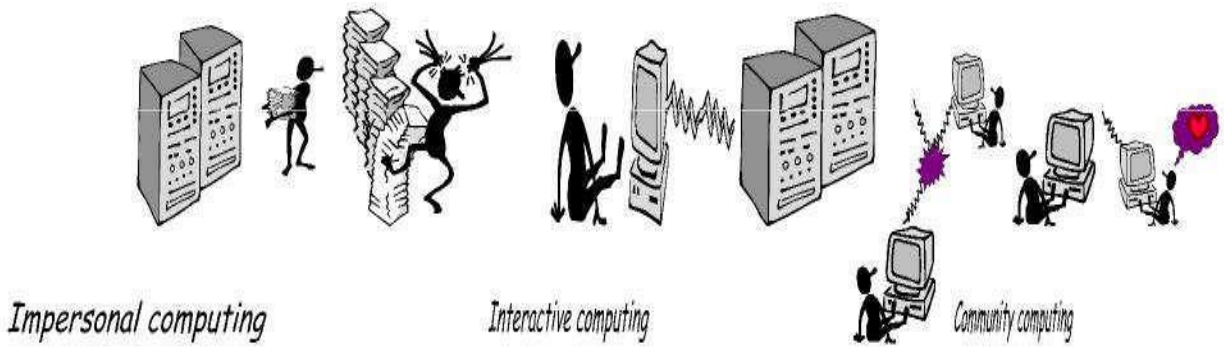
The paradigm shifts commenced fron timesharing processing system that signifinified an interactive computing.

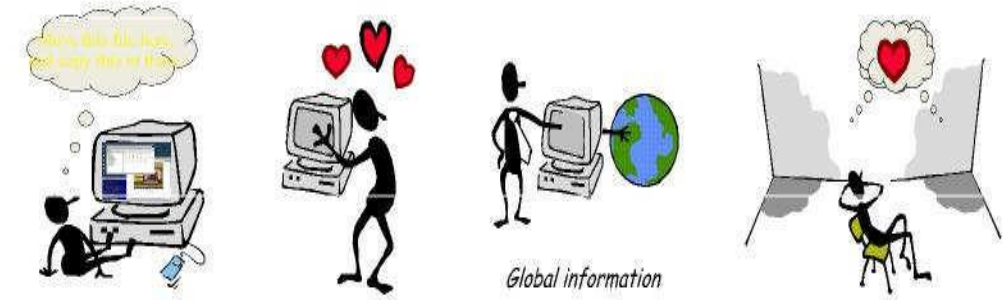
This was followed by another paradigm shift in networking that represented a community computing.The graphical display was an innovation whose era indicated a paradigm shift to direct manipulation of devices.The Micro processor innovation ided opportunity for personal computing as an example of another prov 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 significat 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 everday activities.

Pictorial representations of the shifts are illustrated below:





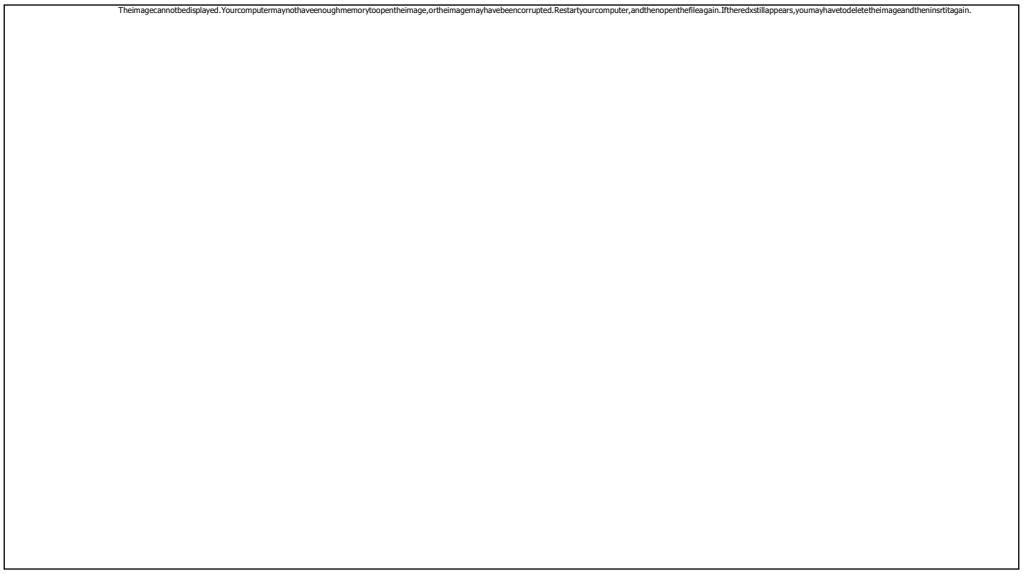


Figure 1: Approximate time lines showing where work was performed on some major technologies. .

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
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 improvements more than anything else has triggered this explosive growth.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 appropriat***zz***, 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