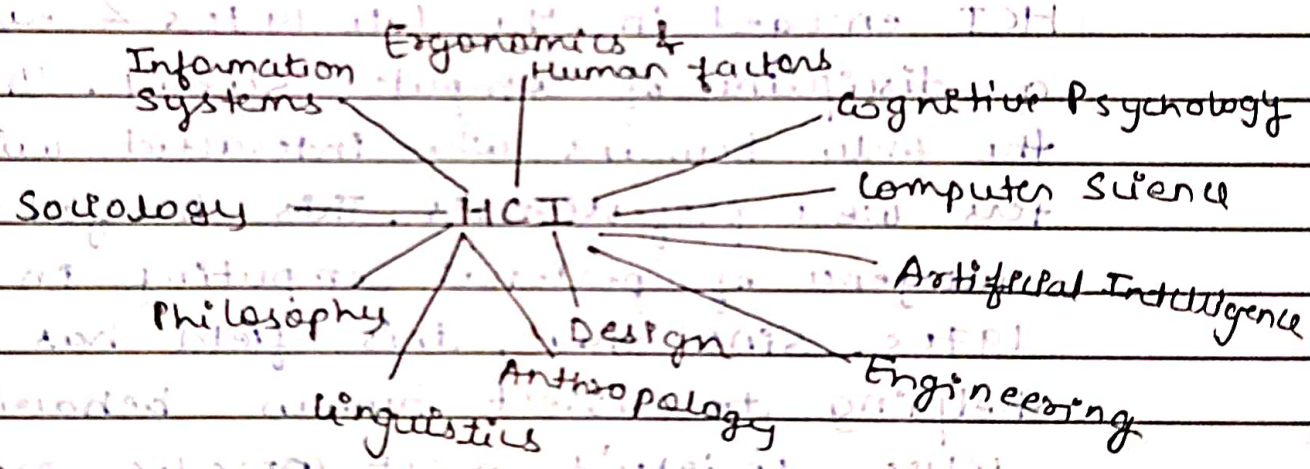
Unit-1

PAGE NO.:
DATE: / /

* Human Computer Interaction (HCI)

Human computer interaction is the study of interaction blw people (users) and computers. HCI is characterized as a dialogue or interchange blw the human & the computer because the output of one serves as the input for the other in an exchange of actions & intentions. HCI is an interdisciplinary field in which computer scientists, engineers, psychologists, social scientists and design professional play important roles.



HCI is an interdisciplinary field as it combines theories & practices from a number of fields including computer science, anthropology, ergonomics, engineering, etc.

HCI is ~~concerned~~ concerned with the design, evaluation and implementation of interactive computing systems for human use & with the study of major phenomena surrounding them.

PAGE NO.:
DATE: / /

* Need of HCI

In the past, computers were expensive and used by technical people only. Now, computers are cheap and used by non-technical people. HCI seeks to understand each user and how they interact with technology in order to improve this relationship. That's why, computer & software manufacturers have noticed the importance of making computers "user-friendly" and "easy to use".

* History of HCI

HCI emerged in the late 1970s & early 1980s as a discipline of computer science. Until 1970s, the only humans who interacted with computers were professionals. This changed with the emergence of personal computing in later 1970s. Since then, this field has evolved, helping to dissect human behaviour to solve society's most complex problems.

- 1970s - Rise of PC
- 1980s - Graphical user Interface
- 1900s - The Internet & collaborative works
- 2000s - Mobile computing & beyond

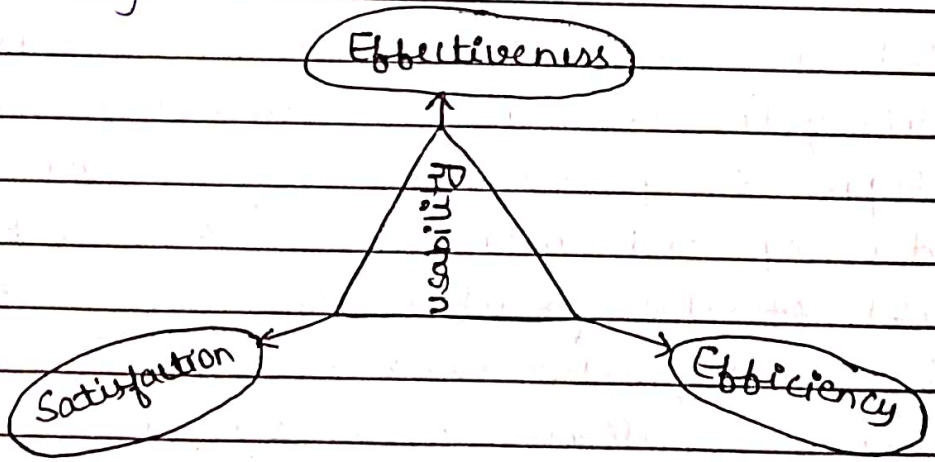
* Usability

Usability is a key concept in HCI. It refers to how well a user & a system can communicate clearly and without misunderstanding through the interface.

Usability is related to the development of interactive systems that are :

- Easy to learn
- Easy to use
- Effective
- Efficient
- Safe to use
- Enjoyable to use
- Easy to evaluate

Usability is also considered in context of the user. For example, what is usable to one user may not be usable to another user. Usability has three components with which users accomplish tasks - Effectiveness, Satisfaction & efficiency.



→ Effectiveness - The completeness with which users achieve their goals.

→ Efficiency - An effective system defines how good the system is & whether it accomplishes the tasks that it is supposed to. It illustrates how the system provides the necessary support to users to complete their tasks.

→ Satisfaction - The enjoyment and comfort experienced while interacting with the interface.

* HCI and Software Engineering

HCI is a field that focuses on how people interact with computers. It observes how humans interact with computers & design technologies that allow humans to interact with computers in new way. While, software engineering is the study of designing, development & preservation of software. It comes in contact with HCI to make the human & computer interaction more vibrant & interactive.

HCI is included in various phases of software lifecycle and it can improve software development phases like planning, design and development, management & maintenance. The use of HCI practices in testing phase helps in finding out how to improve the user satisfaction in terms of usability of a software product.

Benefits of HCI in Software Engineering:

- Elevated user satisfaction
- Decreased time & costs
- Greater system performance
- Improved efficiency & effective ability
- Quality performance

PAGE NO.:
DATE: / /

* GUI and aesthetics

Graphic user interface (GUI) is the interface from where a user can operate programs, applications or devices in a computer system. This is where the icons, menus, widgets, labels exists for the users to access.
It is significant that everything in the GUI is arranged in a way that is recognizable and pleasing to the eye, which shows the aesthetic sense of the GUI designer.

Requirements for a GUI :

→ Functional : It should be functional i.e easy to use and user should be able to operate without any additional learning about the system.

→ Communicable : It should express what it represents. GUI should be unambiguous, literally and contextually compatible.

→ Aesthetic : GUI aesthetics provides a character and identity to any product. Aesthetics is not only about beautification, it has as much to do with function as with beauty. Aesthetics is both art as well as mathematics. For a aesthetic GUI, it should be distinct, recallable, recognizable, pleasing to the eye and have highest visual quality.

- Use simple graphic design
- Avoid clutter
- Use simple & natural dialog
- Put info. in logical & natural order.

PAGE NO.:
DATE: / /

Unit-II

Model-based design & evaluation

→ Model-based design

Model-based design (MBD) is a mathematical modeling-based method for designing, analyzing and validating dynamic systems. MBD performs verification & validation through testing in a simulation environment.

State the Problem → Model Physical Processes → Characterize the problem → Derive a control algorithm → Select models of computation → Specify hardware → Simulate → Construct → Synthesize software → Verify, validate & test

Stages of MBD

→ Model-based evaluation

Model-based evaluation is a method that uses a model of how a person would use a system to predict usability. Evaluation should be considered at all stage in the design life cycle. Evaluation occurs in laboratory, field or in collaboration with users. Model based evaluation evaluate both design & implementation.

Goals of evaluations

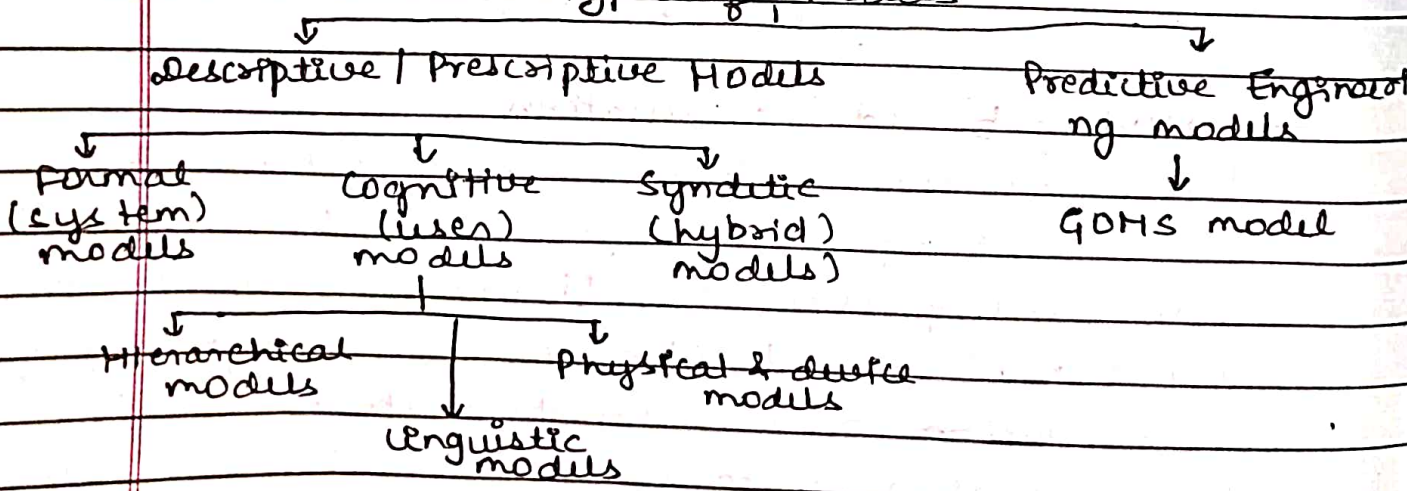
(1) To assess the extent & accessibility of the system's functionality.
(2) To assess users' experience of the interaction
(3) To identify any specific problems with the system.

Evaluation approaches through expert analysis :-
- Cognitive walkthrough
- Heuristic evaluation
- The use of model
- Use of previous work

→ Different type of Models

In HCI, models are used to guide the creation of an interface. Models are often less technical than theories.

Types of Models



→ Descriptive / prescriptive models

Descriptive models help understand how a

users will interact with a computer interface. They can also help to improve the design of the interface. Descriptive models uses behavioral, structural & other descriptions. These descriptions establishes logical relationships about the system. These models help in formulating guidelines for interface design.

→ Predictive Models

These models can predict behaviour of a user in quantitative terms. The predictive engineering models used in HCI are of three types:

(1) Formal (system) models

In these models, the interactive system (interface & interaction) is represented using formal specification techniques.

eg - the interaction modeling using state transition networks.

(2) Cognitive (user) models

These models capture the user's thought (cognitive) process during interaction. These models represent users of interactive systems.

Types of cognitive models

- Hierarchical models: These models represent a user's task and goal structure. eg - GOMS model, CCT
- Linguistic models: These models represent the user-system grammar. eg - BNF, Task-action grammar
- Physical & device models: These models represent human motor skills. eg - Three state model, Keystroke level model

(3) Synthetic (hybrid) models

In these models, both the system (external aspect) and the cognitive activities (internal aspect) are combined and represented using formal specification.

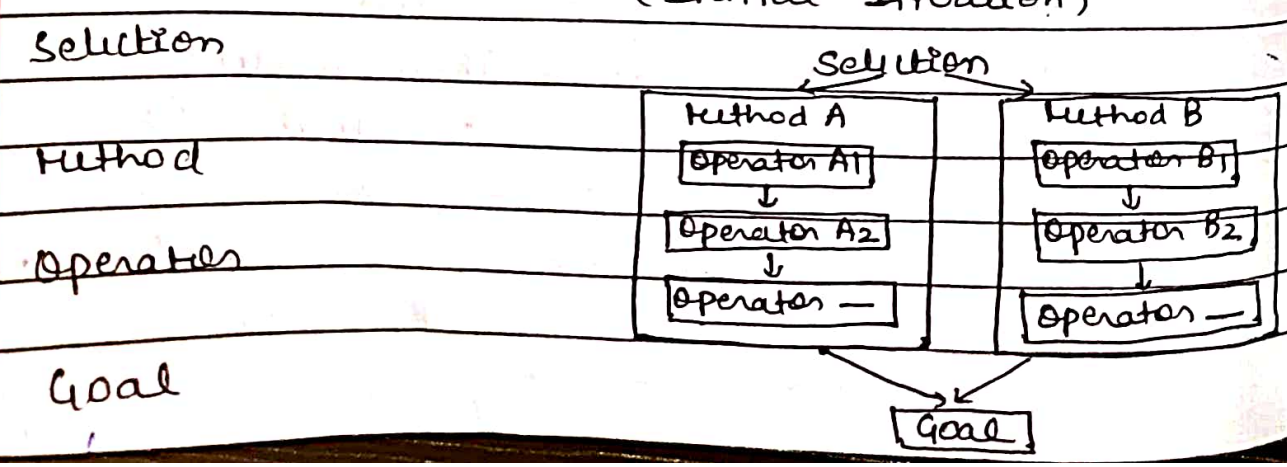
* GOMS Family of Models

GOMS is a family of models that predicts human performance. It stands for Goals, Operators, Methods & Selection rules.

- Goal: What the user wants to achieve
- Operators: Basic actions user performs
- Methods: A series of steps consisting of operators that the user performs.
- Selection Rules: Means of choosing b/w the competing methods.

GOMS family consists of FOUR models:

- Keystroke Level Model or KLM
- CMN-GOMS (Original GOMS model)
- Natural GOMS language (NGOMSL)
- Cognitive Perceptual Motor (CPM GOMS)



→ Keystroke level Model (KLM) :-

KLM predicts how long it takes an expert user to complete a task without errors. It is based on CMN-GOMS. For example, this model can predict how long it takes to close the PPT using close menu option.

It composed of six elements or operators :-

→ K - Keystroke or button press :-
This operator counts the no. of times keyboard buttons are pressed as well as mouse clicks.

→ P - Pointing with a mouse
This operator represents the action of moving the mouse to point the cursor to a desired place on the screen.

→ M - Mental Preparation
This operator represents the time needed for thinking or planning an action or decision making. Time = 1.35 sec

→ R - System Response time
This is the time that the user must wait on the system before he or she can proceed. Thus, R is unique to the system. It is sometimes also called W or waiting time.

→ H - Homing hands on the keyboard or other device.
Time = 0.40 sec

→ D - Manually drawing
This is used when drawing a line with a mouse. It is not frequently used. It's time taken is variable as it depends on the length of the line.

Thus, the predicted time it takes to execute a given task is
$T_{execute} = T_K + T_P + T_H + T_D + T_H + T_R$

→ CMN - GOMS

CMN - GOMS stands for Card, Moran and Newell GOMS. It is the original version of the GOMS technique. It was developed by Stuart Card, Thomas P. Moran and Allen Newell.

In CMN-GOMS, a hierarchical cognitive (thought) process is assumed unlike the linear thought process of KLM. CMN-GOMS can predict operator sequence & execution time. CMN-GOMS builds on the Keystroke level model (KLM) by adding subgoals and selection rules. CMN GOMS allows us to model the tasks and user actions in terms

consists
of four actions:

- Goals: It represents what the user wants to achieve, at a higher cognitive level. The notion of goal allows us to model a cognitive process hierarchically.
- Operators: Operators are basic actions user performs. Operators in CHN-GOMS are similar to KLM. The major difference is that in KLM, only seven operators are defined and in CHN-GOMS, the operators are not restricted to seven.
- Methods: These are series of steps consisting of operators that the user performs. To accomplish a sub-goal.
- Selection Rules: Sometimes, there can be more than one method to accomplish a goal. Selection rules provides a mechanism to decide among the methods in a particular context of interaction.

Limitations:

- Like KLM, CHN-GOMS also models only skilled (expert) user behaviour. That means user does not make any error.
- Cannot capture the full complexity of human cognition such as learning effect, parallel cognitive activities & emotional behaviour.

Limitations of KLH

- Model only expert users behaviours
- User errors cannot be modelled.

- Analysis should be done for representative tasks otherwise, the prediction will not be of much use in design and finding representative tasks is not easy.

* Fitts' Law

It is one of the earliest predictive models used in HCI. It was proposed by PM Fitts in 1954. The Fitts' law is a model of human motor performance. It mainly models the way we move our hands and fingers. Fitts' law is both discriptive and a predictive model. It is discriptive as it provides throughput, which is discriptive measure of human motor performance. It is a predictive model as it provides prediction eqⁿ for the time to acquire a target, given the distance & size of the target.

Characteristics of Fitts' law:

- The movement is related to some 'target acquisition task' i.e the human wants to acquire some target at some distance from the current hand / finger position.
- The movement is rapid & aimed.
- The movement is error-free.

As to Fitts' law, the time to move

& point to a target of width W at a distance A is,

$$MT = a + b \log_2 \left(\frac{2}{W} D + \frac{1}{6} \right)$$

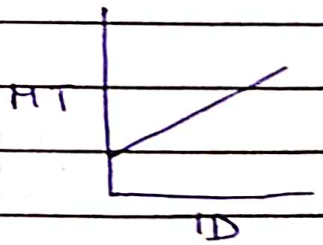
- MT → movement time
- a & b are constants, a → intercept, b → slope
- D → The distance by which the user needs to move his hand / finger. This is also called amplitude (A) of the movement. The larger the D is, the harder the task becomes.
- W → Width of the target. As the width increase, the task becomes easier.

Index of difficulty (ID) measures the task difficulty.

$$ID = \log_2 \left(\frac{D}{W} + 1 \right) \text{ (units in bits)}$$

Index of performance (IP) or called throughput is given by

$$TP = \frac{ID}{MT} \text{ (bits / seconds)}$$



* Hick - Hyman's Law

Hick - Hyman law was given by Edmund Hick and Ray Hyman. It is a psychological principle which states that the more options are available to a person, the longer

it will take him/her to make a decision about which option is best. If the user is bombarded with choices then he/she have to take time to interpret and decide.

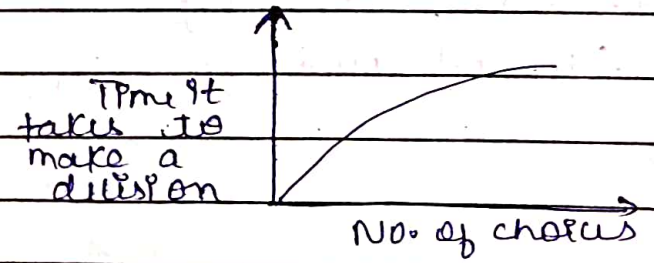
The law states that the reaction time increases with uncertainty about the judgement or decision to be made. The law provides a logarithmic function which models the reaction time (RT) required to make a decision.

$$RT = a + b \log_2 n$$

RT → reaction time

n → no. of choices

a, b → constants which depends on the circumstances of the decision being made.



Application - Reduces no. of choices, simplify decision-making

PAGE NO.:
DATE: / /

Unit-III

Guidelines in HCI

* Shneiderman's Eight Golden Rules

Ben Shneiderman's Eight Golden Rules are a set of eight guidelines for designing user interfaces that are easy to learn & use. The rules:

→ Strive for consistency: This rule emphasizes on use of consistent terminology, fonts, and design elements throughout the interface which reduces the need to learn new operations.

→ Cater to Universal usability: Design the interface to be accessible to users with a wide range of abilities.

→ Offer informative feedback: Provide users with feedback on their actions so that they know what is happening.

→ Design dialogs to yield closure: Make sure that users know when a dialog box is complete and what they need to do to proceed.

→ Prevent errors: Design the interface to prevent errors from happening in the first place.

→ Permit easy reversal of actions: Allow users to easily undo their mistakes.

→ Support internal locus of control: Make users feel like they are in control of the system.

→ Reduce short-term memory load: Minimize

the amount of information that users need to keep in their heads in order to use the system.

* Norman's Seven Principles

To assess the interaction blw human and computers, Donald Norman in 1988 proposed seven principles. He proposed the seven stages that can be used to transform difficult tasks.

Seven principles of norman are :-

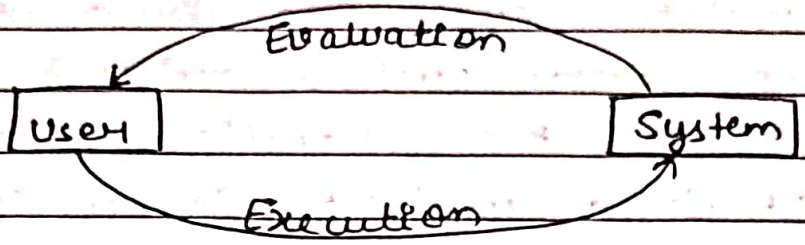
- Use both knowledge in world & knowledge in the head
- Simplify task structures.
- Make things visible.
- Get the mapping right.
- Convert constraints into advantages
- Design for error
- When all else fails - standardize.

* Norman's Model of Interaction

Norman's model of interaction also known as the execution - evaluation cycle, is a model of human - computer interaction. It is considered as one of the most influential models in HCI. This model describes the steps a user takes when interacting with a computer system.

Norman viewed the interaction as a cycle

with two components: execution & evaluation.



Norman's interaction cycle

Acc. to Norman, the execution component can be further divided into:

- Establish a goal that needs to be accomplished.
- Form the intention that will accomplish the goal.
- Specify the action sequence to implement the intentions.
- Execute the action.

The evaluation component is divided into:

- Perceive the state of system resulting from the action.
- Interpret the system state.
- Evaluate the system state / evaluate your interpretation against the expectation based on your intentions.

When people try something they face two Gulfs:

- The Gulf of execution: They try to figure out how it operates. It is the gap b/w the user's goal & its computerized implementation.
- The Gulf of evaluation: They try to figure out what happened. It is the gap b/w the computerized implementation of the user's goal and its evaluation by the user.

PAGE NO.:
DATE: / /

* Heuristic Evaluation

Heuristic evaluation is a process where experts use certain rules & principles to measure the usability of user interfaces. It is a usability inspection method that helps to identify usability problems in the user interface design against a set of heuristic principles (also known as heuristics).

* Nielsen's ten Heuristics

Nielsen's ten heuristic principles are a set of ten guidelines for designing user interface that are easy to use. The principles are :

(1) **Visibility of system status :** Keep users informed about what is happening with the system.

eg- When you finish watching an episode of a series on Netflix, the system provides you a small screen telling you how long it will take to load the next episode.

(2) **Match b/w the system & the real world :** Use words, phrases & concepts that are familiar to the user, rather than system-oriented terms. Follow real-world conventions making information appear in a natural & logical order.

eg- When you read on Kindle, the pages turn with a swipe which imitates the experience of reading a physical book.

(3) User control & freedom : The system should not impose action on the user or make decisions for them and allow users to easily recover from errors & undo actions.

eg- Google Docs has undo & redo options.

(4) Consistency & standards : This heuristic is about keeping the same language and conventions throughout the system. Icons, buttons and terminology should behave predictably.

eg- In Gmail, you will see icons like ☆ stared, ▷ Sent, 🕒 Snoozed, ✉ All mail, etc which are completely predictable.

(5) Error prevention : This heuristic proposes that a good design should always prevent problems from occurring.

eg- Creating a warning message to confirm the decision of deleting a file.

(6) Recognition rather than recall : This heuristic minimize the user's memory load by making objects, actions & options visible. The user should not have to remember all the actions or functions of the system.

eg- Menu items should be visible or easy to access when needed.

(7) Flexibility & efficiency of use : The design should benefit both inexperienced and

experienced users. Provide shortcuts, accelerators and efficient workflows for experienced users, while not hindering novices.

eg- Keyboard shortcuts in software applications

(8) **Aesthetic and Minimalist Design**: Strive for a clean, simple and visually pleasing design. Avoid unnecessary elements that can distract or confuse users.

eg - Google search, Apple iOS

9) **Help users recognize, diagnose & recover from errors**: The design should help the user identify & find the errors. Error messages should be expressed in plain language, precisely indicating a problem and suggesting a solution.

eg - Kuala's 404 page gives users the option to go back (undo mistake) to where they were.

(0) **Help & documentation**: This heuristic concerns documentation that will help users understand how to perform their tasks. Always deliver help documentation that is easy to search for & focused on user's tasks. Provide - tooltips, chatbots, tutorials, etc.

eg - Apple's assistance for iPhone with actionable steps.

PAGE NO.:
DATE: / /

* Contextual Inquiry

Contextual inquiry is a semi-structured interview method to obtain information about the context of use, where users are first asked a set of standard questions & then observed & questioned while they work in their own environments. Because users are interviewed in their own environments, the analysis of data is more realistic than laboratory data. Contextual inquiry is based on set of principles that allow it to be molded to different situations. This technique is generally used at the beginning of the design process and is good for getting information about work practices.

The four principles of contextual inquiry are:

→ Focus: Plan for the inquiry, based on a clear understanding of your purpose.

→ Context: Go the customer's workplace and watch them do their own work.

→ Partnership: Talk to customers about their work and engage them in different aspects of work.

→ Interpretation: Develop a shared understanding with the customer about the aspects of work that matter.

The results of contextual inquiry can be used to define requirements, improve a process, learn what is important to users & customers.

PAGE NO.:
DATE: / /

* Cognitive Walkthroughs

Cognitive walkthrough is a usability evaluation method. They are designed to see whether or not a new user can easily carry out tasks within a given system. A cognitive walkthrough begins by defining the task or tasks that the user would be expected to carry out. It is these tasks that the cognitive walkthrough will examine for usability.

Questions to be asked during cognitive walkthrough:

- Will the user try & achieve the right outcome.
- Will the user notice that the correct action is available to them?
- Will the user associate the correct action with the outcome they expect to achieve?
- If the correct action is performed, will the user see that progress is being made towards their intended outcome?

Unit - IV

Empirical Research methods in HCI

PAGE NO.:
DATE: / /

* **Empirical Research**

Empirical research is defined as any research where conclusions of the study is strictly drawn from empirical evidence. Therefore, in empirical research the data is collected using evidence that is collected through observation or experience.

eg - Pharmaceutical companies use empirical research to try out a specific drug on controlled groups to study the effect & cause. This way, they prove certain theories they had proposed for the specific drug.

Need for Empirical Research

Empirical research is important in today's world because most people believe in something only that they can see, hear or experience. It is used to validate multiple hypothesis & increase human knowledge.

* **Methods of Empirical Research**

Empirical research can be conducted and analysed using qualitative or quantitative methods

- **Quantitative Research**

Quantitative research methods are used to gather information through numerical data. It is used to quantify opinions, behaviours or other defined variables. These are

predetermined & are in a more structured format. Some commonly used methods are survey, longitudinal studies, polls, etc.

eg - A researcher who conducted a quantitative survey among parents of children aged 1-8 years to study how many parents are fine with their children using phones.

- Qualitative Research

Qualitative research methods are used to gather non numerical data. It is used to find meanings, opinions, or the underlying reasons from its subjects. These methods are unstructured or semi structured. The sample size for such a research is usually small & it is a conversational type of method to provide more insight or in-depth information about the problem. Some of the most popular forms of methods are focus groups, experiments, interviews, etc.

eg -

* Empirical Research Methodology cycle

A. D de Groot, a famous dutch psychologist and a chess expert conducted some of the most notable experiments using chess in the 1940s. During his study, he came up with a cycle which is consistent & now widely used to conduct empirical research. The five phases of this cycle are:

→ **Observation:** Data, related articles, facts, etc are gathered to understand the background survey of the research topic.

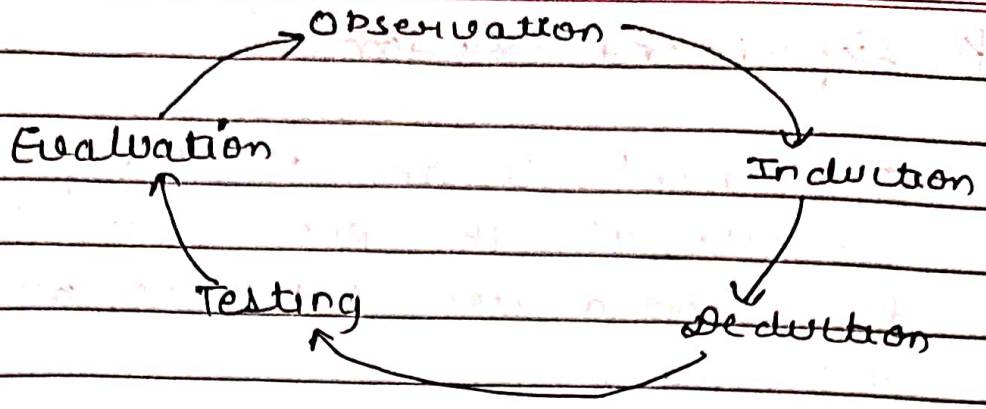
→ **Induction:** Inductive reasoning is then carried out to form a general conclusion from the data gathered through observation.

→ **Deduction:** This phase helps the researcher to deduce a conclusion out of his experiment.

→ **Testing:** Quantitative & qualitative empirical data are gathered. The data is examined, often with statistical analysis. The results can support, refute or be neutral to hypothesis.

→ **Evaluation:** The reasoning, methodology & findings of the experiment are written down & the conclusions of the researcher are presented. Information relating to any difficulties, challenges & limits of test are also included.

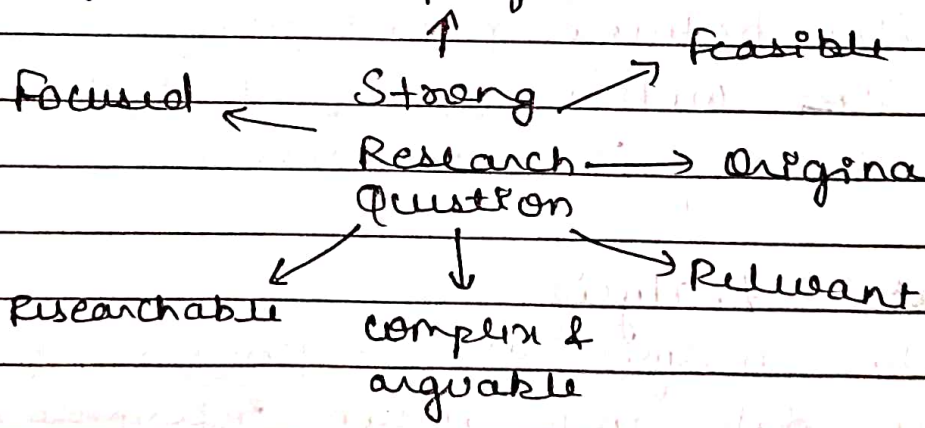
eg:-
- Observation: I do not sneeze at home, I do sneeze at my sister's home & my sister owns a cat, while I do not have a cat.
- Induction: I may be allergic to cats.
- Deduction: I hypothesize that, if I go to the pet store & pick up a cat, I will start sneezing.
- Testing: I went to the pet store & when I picked up the cat, I started sneezing.
- Evaluation: My trip to the pet store supports the idea that I am allergic to cats.



Empirical Research Cycle

* Research Question Formulation

Formulating research questions is an essential step before starting any research. It involves exploring an existing uncertainty and identifying a need for further investigation.



Steps of Question Formulation :-

- Design a question focus (QFocus)
- Produce Questions
- Rework Questions
- Prioritize Questions
- Explore further
- Reflect

PAGE NO.:
DATE: / /

* Experiment design

Experiment design is the process of putting together all the pieces needed to test hypothesis on a user interface or interaction technique.

Steps for designing an experiment are :-

- Determine the research question
- Define the variables
- Form a hypothesis
- Choose a design type
- Create a test group
- Collect research
- Repeat the experiment

Three commonly used designs :

→ The between-group design
→ The within-group design
→ The split plot design

* Data Analysis

Data analysis is the process of converting raw data into useful information for decision-making.

* Analysis of Variance (ANOVA)

ANOVA stands for analysis of variance. It is a statistical test commonly used in HCI research. It is a tool that splits the

data set's variability into two parts: systematic & factors & random factors. The systematic factors have a statistical influence on the given data set, while the random factors do not. Analysts use the ANOVA test to determine the influence that independent variables have on the dependent variable in a regression study.

ANOVA was developed by Ronald fisher in 1918. It is the extension of t-test & z-test.

The formula for ANOVA is:

$$F = \frac{MST}{MSE}$$

where,

F = ANOVA coefficient

MST = Mean Sum of squares due to treatment

MSE = Mean Sum of squares due to error

Anova is used to analyze the differences b/w the means of more than two groups. It is commonly used in three ways.

- One-way ANOVA
- Two-way ANOVA
- N-way ANOVA

One-way ANOVA evaluates the impact of a solo factor on a solo response variable. It determines whether all the samples are the same. It is used to determine whether

there are any statistically significant differences b/w the means of three or more independent groups.

Two-way ANOVA is an extension of one-way ANOVA. In two-way ANOVA, there are two independent variable affecting a dependent variable.

Assumptions of ANOVA

- Samples follow normal distribution.
- Samples have been selected randomly & independently.
- Each group should have common variance.

Null Hypothesis

The means for all groups are the same (equal)

$H_0 : \mu_1 = \mu_2 = \mu_3 = \cdots = \mu_n$

Alternate Hypothesis

The means are different for at atleast one pair of groups.

$H_1 : \mu_1 \neq \mu_2 \neq \mu_3 \neq \cdots \neq \mu_n$

Unit-V

Task modelling and analysis

PAGE NO.:
DATE: / /

* **Task**

Task is referred as an activity or action that is performed by the user while interacting with a service / application.

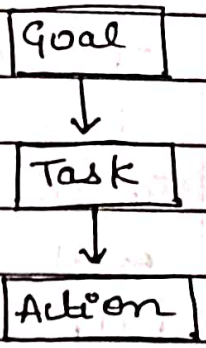
Task analysis

Task analysis is the study of how a user will complete any certain task successfully.

Goal: state of the system that a human wants to accomplish.

Task: activities required, used, or deemed necessary to achieve a goal.

Action: steps required to complete the task.



* **Hierarchical Task Analysis (HTA)**

Hierarchical task analysis involves decomposing tasks into subtasks & analyzing the logic of sequence needed to execute the task to achieve the set goal in an optimal way. It is a structured & objective approach. It gives an understanding of the steps taken by users to achieve a certain goal. It helps in achieving the goal in best possible way.

HTA steps of execution:

- Define task being analyzed
- Conduct data collection

- Determine the overall goal of the task.
- Determine task sub-goals.
- Perform sub-goal decomposition
- Develop plans analysis

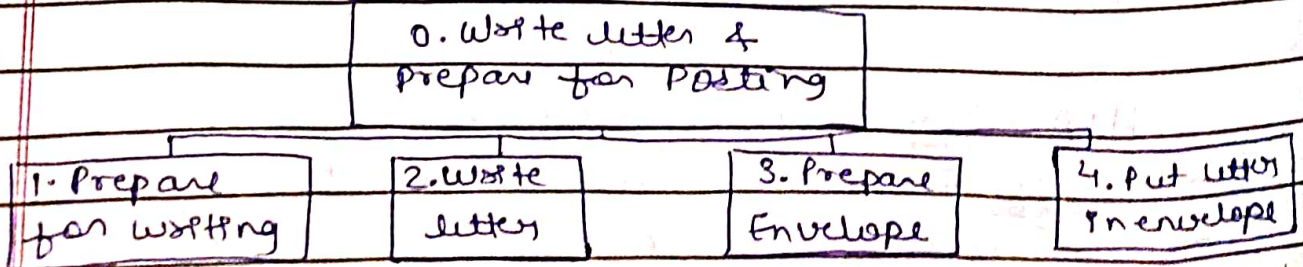
Advantages

- It lets you compare different approaches to the same task.
- It provides extensive information about a task.
- Evaluates systems against usability or functional requirements.
- Helpful in error analysis.

Disadvantages

- HTA diagrams can become quite complex
- There is no strict rule for creating an HTA diagram so different analysts will generate inconsistent hierarchies at varying level of detail.
- Time consuming.
- It is not a predictive tool. It focuses on existing tasks.

Eg -



0: Write letter & prepare for posting

1: Prepare for writing

- 1.1 : Get Paper
- 1.2 : Get envelope
- 1.3 : Get pen
- 1.4 : Get address book

2 : Write letter

- 2.1 : Write own address
- 2.2 : Write addressee's address
- 2.3 : Write date & "Dear..."
- 2.4 : Write body text of letter
- 2.5 : Sign off

3 : Prepare envelope

- 3.1 : Write name on envelope
- 3.2 : Write addres on envelope

4 : Put letter in envelope

- 4.1 : Fold letter
- 4.2 : Plow letter into envelope
- 4.3 : Seal envelope

* Engineering Task Models

Engineering task models are detailed, formal representations of how users interact with a system to complete specific tasks. They provide a structured way to describe and analyze user activities and interactions within a software application or system.

Characteristics of ETM:

- ETM have flexible notations, which describes the possible activities clearly.
- They have organized approaches to support the requirement, analysis, and use of task models in the design.
- They support the reuse of in-condition design solutions to problems.
- They let the automatic tool accessible to support the different phase of the design cycle.
- ETM's like GOMS can estimate the time a human might need while interacting with a system or service.

* Concur Task Tree

CTT is a modeling technique and a notation used in design process & task analysis. It has been designed to analyze and model task based user interactions with a computer system / problem.

Key features:

- Focus on activities that users aim to perform.
- Hierarchical structure
- Graphical syntax
- Rich set of temporal operators

In CTT, four Task categories are defined:

- User task: These are the tasks that represent only internal cognitive activity of a

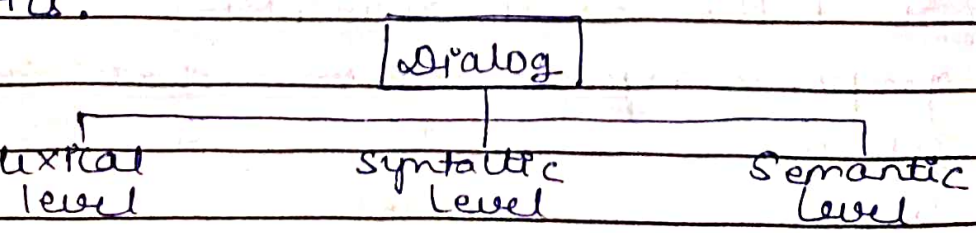
user, such as selecting a strategy to solve a problem.

- Interaction task: These are user actions with possibility of immediate system feedback, such as editing a diagram.
- Application task: These are the tasks performed by the system only, such as generating a query result.
- Abstract task: These refers to the tasks whose subtypes are of different types or the task type is not yet divided.

* Dialog

A dialog is the construction of interaction b/w two or more beings or systems. In HCI, a dialog is studied at three levels -

- Lexical: At this level, details such as the shape of icons, actual keys pressed etc are dealt with.
- Syntactic: The order of inputs & outputs in an interaction are described at this level.
- Semantic: This level offers the meaning and impact of the dialog on the application or statistics.



* Formalism in Dialogue Design

Formalism in dialogue design is a model that uses mathematical or computational models to represent or analyze dialog structures. Formal techniques for representing dialogs serves two purposes:

- Understanding the proposed design better
- Analyzing dialogs to identify usability issues

Formal models offer a scientific & rigorous technique to describe the behaviour of a dialog design. Formal models are:

→ Finite State Machines
→ State Charts
→ Petri nets

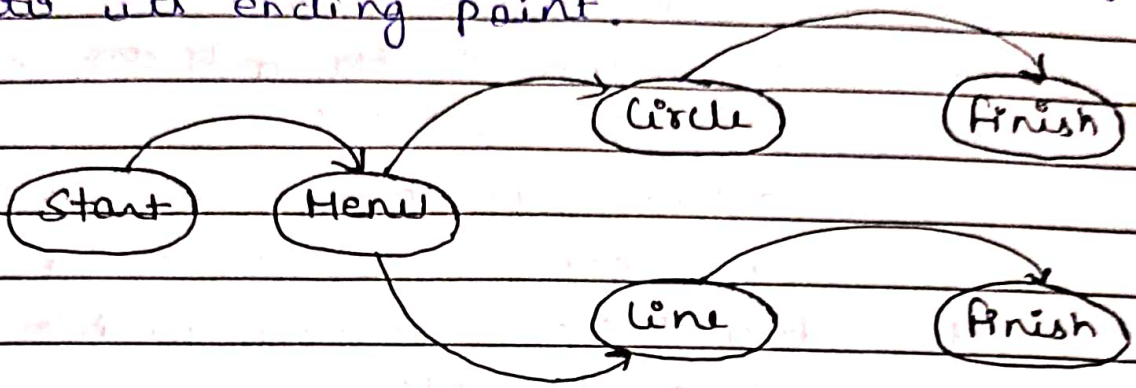
* Design using FSM

FSM are a foundational formalism for dialog design in HCI. They consist of states, transitions, and events making them appropriate for representing easy dialog systems.

In FSM, state transition network is a graphical representation that illustrates the flow of a dialog system. An STN is made up of two entities:

- Circles: A circle denotes the system system that has been given a name and is branded.

- Arcs : The circles are joined by arcs, which denote the action or occurrence that causes the arc to change from its starting point to its ending point.



State Transition Network

* Statechart

A statechart models the flow of control from one state to another for a particular object within a system. Statecharts represent complex reactive systems that extends Finite state Machines (FSM), handle concurrency and adds memory to FSM.

Statecharts has the following states :-

- Active state : The present state of the underlying FSM.
- Basic states : These are individual states & are not composed of other states.
- Super states : These states are composed of other states.

Components of state charts :-

→ State : It represents the condition of the system.

→ Transition : Transition represents the change from one state to another.

→ Action : An action is the activity of an object that is initiated by an event. An action describes what the object does in response to the event.

→ Event : Events are triggers which initiate state transition.

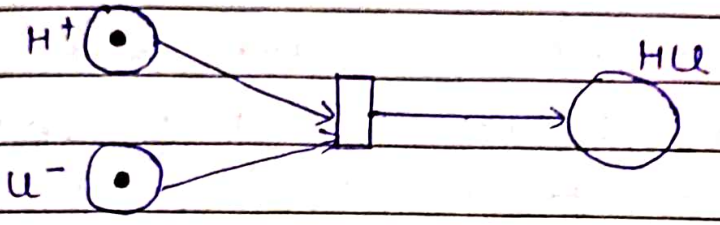
* Petri Nets

It was introduced by Carl Adam Petri. Petri nets are a graphical model that represents systems with multiple independent activities happening simultaneously. They are popular data flow modeling formalism.

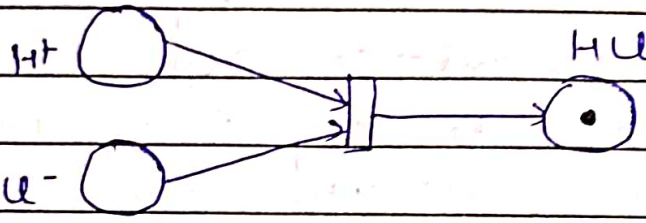
Components of Petri Nets :-

- Place - Place represents conditions or states in the system. A place is represented by circle.
- Transition - Transition represents event or activities which can exchange the system state. Transitions are represented by squares/rectangles.
- Arc - Represents directional connections b/w places & transitions. Arc is represented by arrows.
- Token - Tokens are object that circulate between place and transitions, indicating the system's state. Tokens are represented by small filled circles.

eg:- $H^+ + Cl^- \rightarrow HCl$ (A)



(B)



(C)

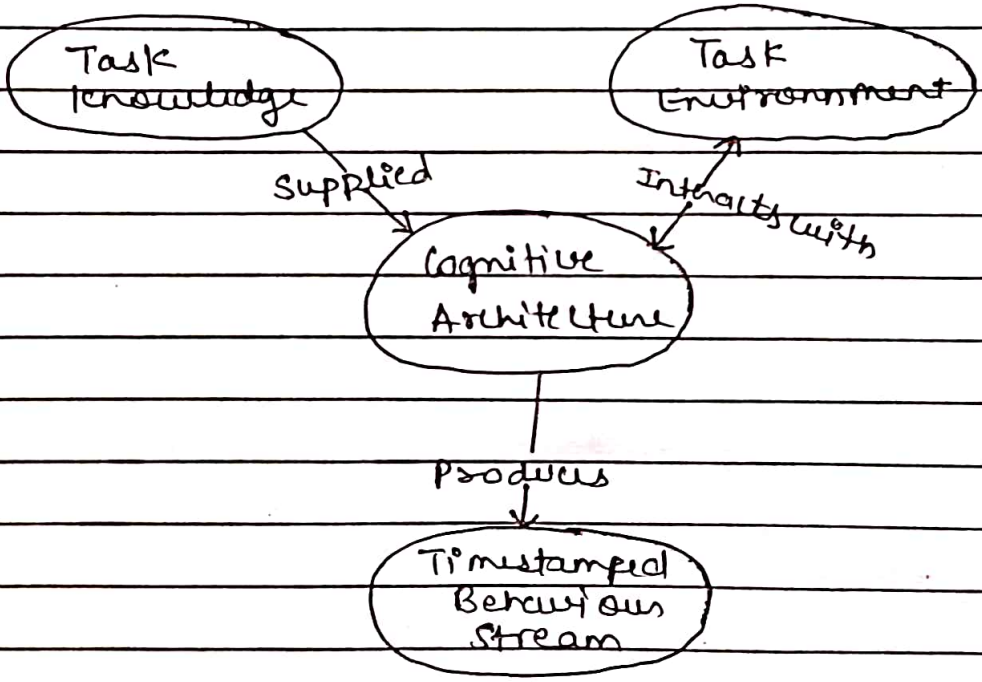
Unit-VI

Introduction to CA

PAGE NO.:
DATE: / /

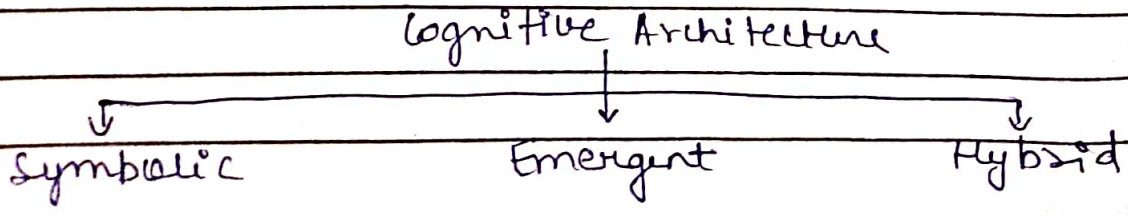
* Introduction to Cognitive Architecture

Cognitive architecture is the theory regarding the human mind, its structure, and how the various components work in sync to manage intelligent behaviours in complex environments. It aims to use cognitive psychology research to create a complete computer-based cognition model. Cognitive architecture aims to create artificial computational system processes that work like human.



* Types of Cognitive Architecture

Cognitive architecture can be symbolic, emergent and hybrid.



- Symbolic cognitive architecture : It is a framework that represent & manipulate knowledge symbolically. It involves using symbols, rules, & logic to model human cognition in computer systems.
- Emergent cognitive architecture : It involves the emergence of complex behaviours from simple interactions. In this, the system evolve & learn from user interactions over time.
- Hybrid cognitive architecture : It combines both symbolic & emergent approaches.

* Human Processor (HHP)

The human Processor model was developed by Card, Moran & Newell in 1983. It describes the cognitive process that people go through b/w perception and action.

The HHP contains three interacting subsystems :

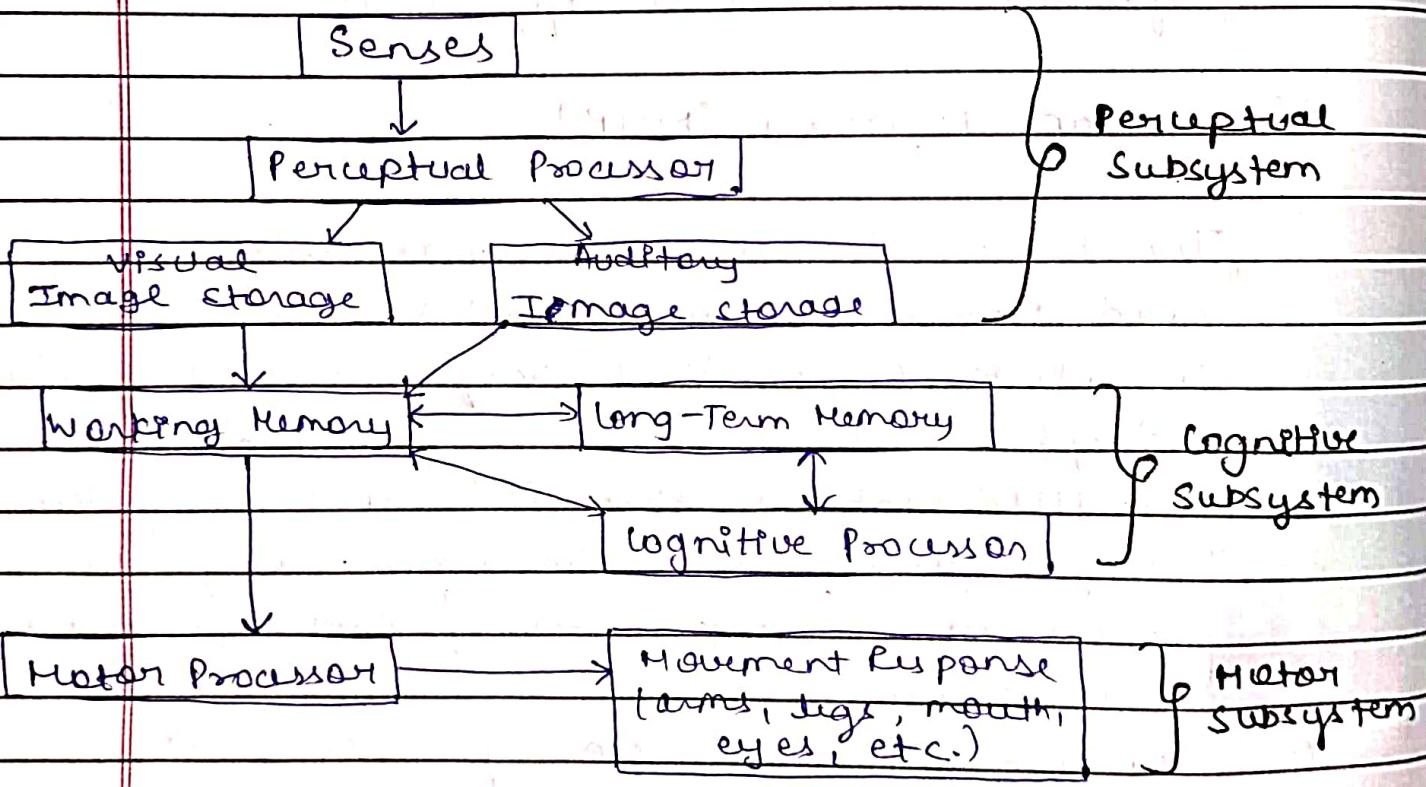
(i) The perceptual subsystem representing the process of taking sensory input from the environment.

(ii) The cognitive subsystem representing the process of cognition.

(iii) The motor subsystem representing the motor actions in response to some perceptual or cognitive stimulus.

Each of these subsystems has their own memory & processors.

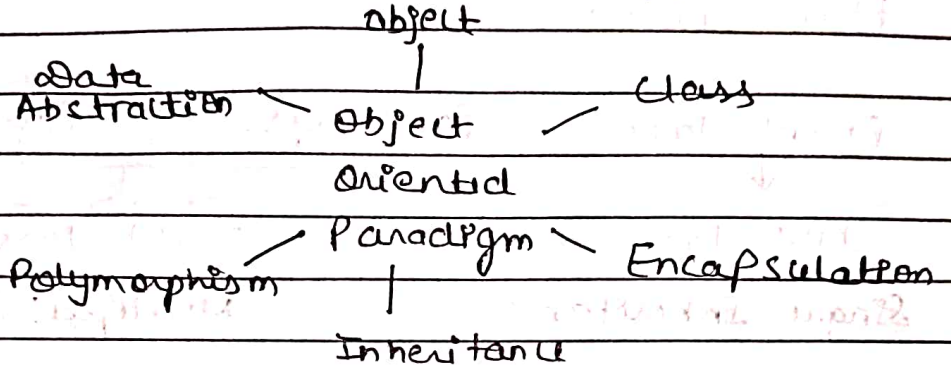
- The perceptual subsystem consists of two types of memory, namely the visual and the auditory.
- The cognitive subsystem depends on the short term (working) memory with limited storage capacity and the long term memory with infinite storage capacity.
- The working memory also acts as at the memory component of the motor subsystem. Execution of a motor action takes about 70ms.



* OOP - Introduction

The object Oriented Programming Paradigm plays an important role in human computer interface. It has different components

that takes real world objects & perform actions on them, making live interactions between man and the machine. OOPS mainly revolves around the concept of objects. Objects are the building blocks on OOPS application.

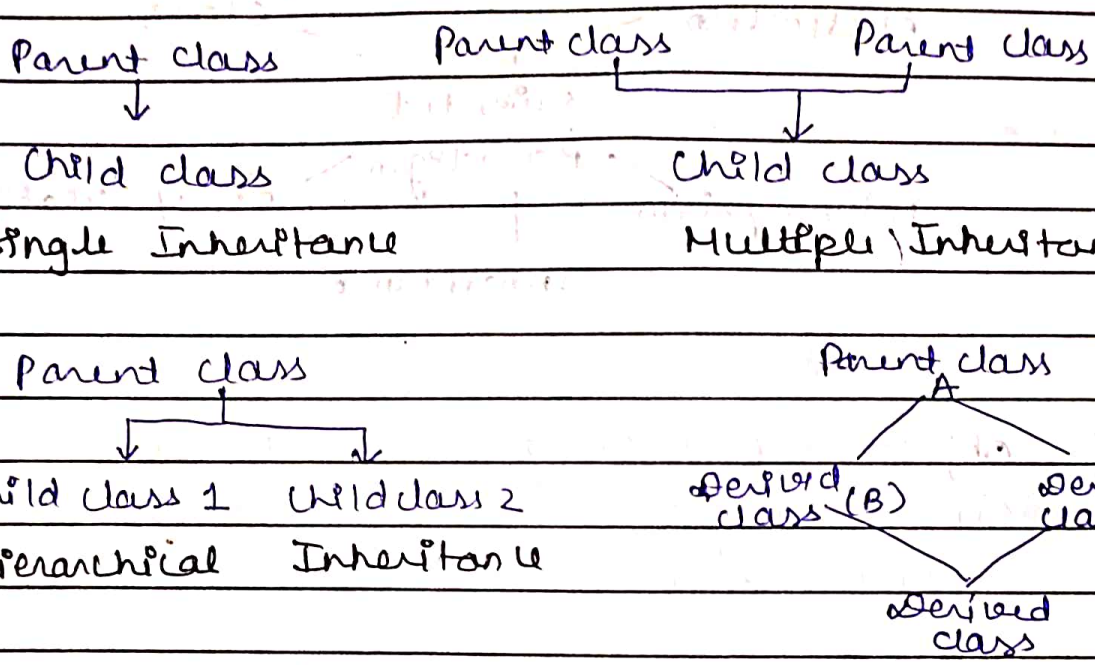


(i) **Object** : Objects are instances of a class. Objects can correspond to real-world objects or an abstract entity.

(ii) **Class** : A class is a group of objects that has mutual methods. It can be considered as the blueprint using which objects are created. They define the attributes & behaviours (method) that object of the class will use.

(iii) **Encapsulation** : Encapsulation is defined as wrapping up of data under a single unit. It is the mechanism that binds together code & the data it manipulates. In encapsulation, the variables or data of a class are hidden from any other class. It is also known as data-hiding.

W **Inheritance :** Inheritance is a mechanism that allows one class (child or subclass) to inherit the properties & behaviours of any other class (superclass or parent class). It promotes code reusability & establishes a relationship b/w classes.



V **Polymorphism :** Poly means many & morphism means form. Polymorphism is the ability of a message to be displayed in more than one form. For example – a person at the same time can have different characteristics like a man can be a husband, father, son, at a same time.

Types of Polymorphism

- Compile Time
  - Function Overloading
  - Operator Overloading
- Run Time
  - Virtual function

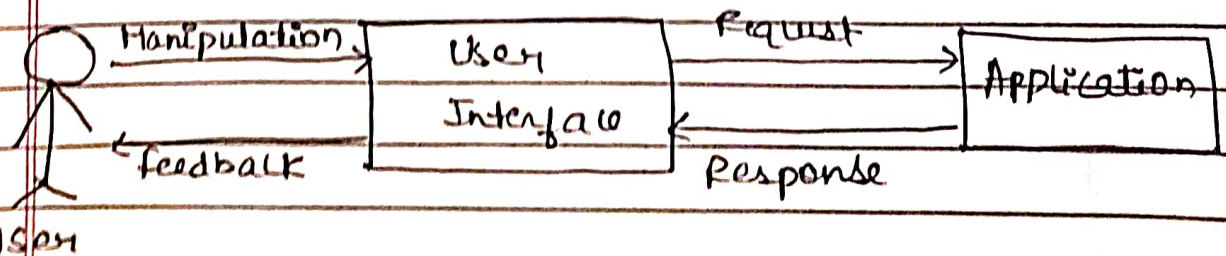
(vi) **Data Abstraction :** Data abstraction refers to providing only essential information about the data to the outside world, hiding the background details or implementation.

eg - A man driving a car knows that pressing the accelerators will increase the speed of the car but he does not know about how on pressing the acceleration the speed is increasing.

* OOH - Object Oriented Modeling of user interface design

Object oriented modeling (OOH) is a way of thinking about problems using models based on real-world concepts.

Object oriented interface unites users with the real world manipulating software objects for designing purpose.



While creating the OOM for interface design, first of all analysis of user requirements is done. The design specifies the structure and components required for each dialogue. After that, interfaces are developed and tested against the use case.

The sequence of processes documented for every use case are then analyzed for key objects. This results into an object model. Key objects are called analyses objects and any diagram showing relationships between these objects is called object diagram.