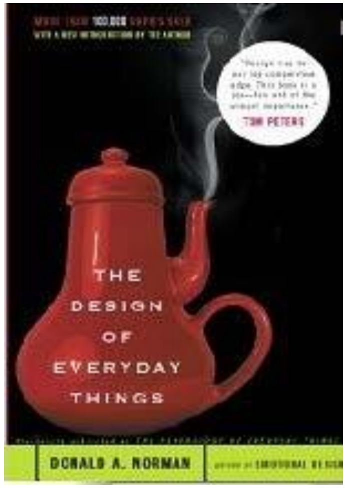
HCI Guidelines: Norman’s Seven Principles and Norman’s Model of Interaction

Professor Ram Mohana Reddy Guddeti
Information Technology Department
NITK Surathkal, Mangalore, India

HCI Guidelines:
Norman’s Seven Principles

Introduction:

Donald Norman, a Researcher, Psychologists & Designer is well known for his book titled: The Psychology of Everyday things.



In 1988 Donald Norman proposed **seven principles** for the evaluation of the interaction between Humans and Computers.

Later he formulated a **model** to understand & integrate a user into the Interface design cycle.

He intended seven stages to be used to transform difficult tasks for which HCI & interface was under development into simple ones.

Norman outlined the underlying principles for his 7 stage models as shown bellow. The seven stage Interaction model is shown in subsequent slides.

Principles underlying the seven stage model

1. Use both knowledge in world & knowledge in the head

2. Simplify task structures.

3. Make things visible

4. Get the mapping right

   (User mental model = Conceptual model = Designed model)

5. Convert constraints into advantages

   (Physical constraints, Cultural constraints, Technological constraints)

6. Design for Error

7. When all else fails – Standardize.

Each of these seven principles will be discussed in the following slides.

Discussions on Norman's seven principles.

1. Use both knowledge in the world & knowledge in the head

- As a basis for his Interaction Model Norman proposed the following levels of abstraction of knowledge of the user :

  - Task Level
  - Goal Level
  - Semantic level
  - Syntax level
  - Lexical level
  - Physical Level

The User models his/her knowledge in/of the world into his / mental realm by the process of cognition.
The user’s knowledge model is not necessarily the same as the knowledge model of the world.

The question is which one should the designer take as reference –Knowledge in the World? Or Model of world knowledge in the User’s Mental realm ?

Continued....

Discussions on Norman's seven principles. Principle 1 continued.

Relying on either of them alone would lead to an incomplete abstraction of knowledge by the Designer.

Norman’s principle mandates that both types of knowledge be considered.

- **Semantic level** describes the set of objects, attributes and operations, which the ‘system’ and the ‘user’ can communicate. Semantics is about how the user interprets and makes meanings out of the system.

- **Syntactic level** describes which conceptual entities and operations may be referred to in a particular command context or system state.

- **Interaction level** describes the translation of commands and objects into the associated physical actions and the structure of the interaction, including typing / mouse / gesture / voice / tactile rules.

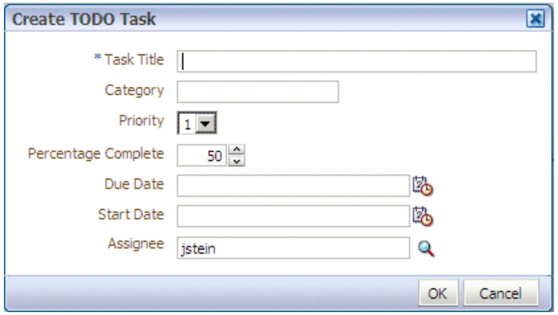
2. Simplify task structures.

- **Task Level:** task level is to analyze the user's needs and to structure the task domain in such a way, that a computer system can play its part. The task level describes the structure of the tasks which can be delegated to the computer system.

- This principle states that a ‘Task’ is to be broken down (by analysis) to their simplest action level such that at each level there is as far as possible only one action involved.

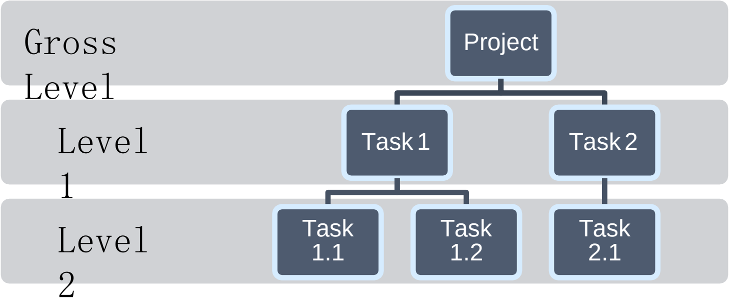
- Doing so makes mapping with the Computer’s programming language easy for a HCI designer to build Interactive Interfaces & Hierarchies.

Example: Gross level & Broken Down level of a GUI



Create TODO Task

* Task Title	|
Category	
Priority	1
Percentage Complete	50
Due Date	
Start Date	
Assignee	jstein



3. Make things visible

Objects at the Semantic level need to be **mapped** to the objects at Syntactic level, for the user. This is achieved with making the connection as ‘Visual’ as possible.

Graphic User Interface designers use ‘ Metaphors’ to make the connection. Example:

Delete action (at the Semantic Level) (Command Line)

= Waste paper basket to dump. (Syntactic level object) (natural language)

= Visual :

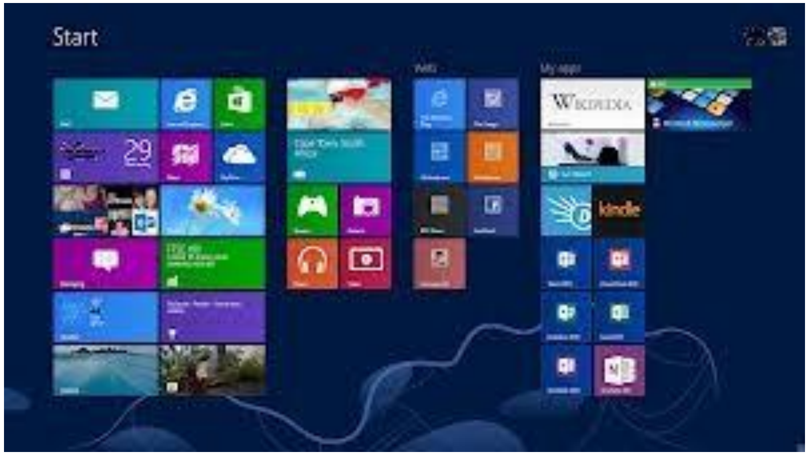
Mapping: the link between what *you want to do* and what *is perceived possible.*

Continued...

Discussions on Norman's seven principles. Principle 3 of 7

HCI Designers use this principle of ‘Making it Visual ‘ to the maximum while designing Interfaces.

Interaction styles such as
WIMP (Windows – Icons – Menus – Pointer)
Three Dimensional Interfaces as in
Virtual Realty can be found in current software interfaces.

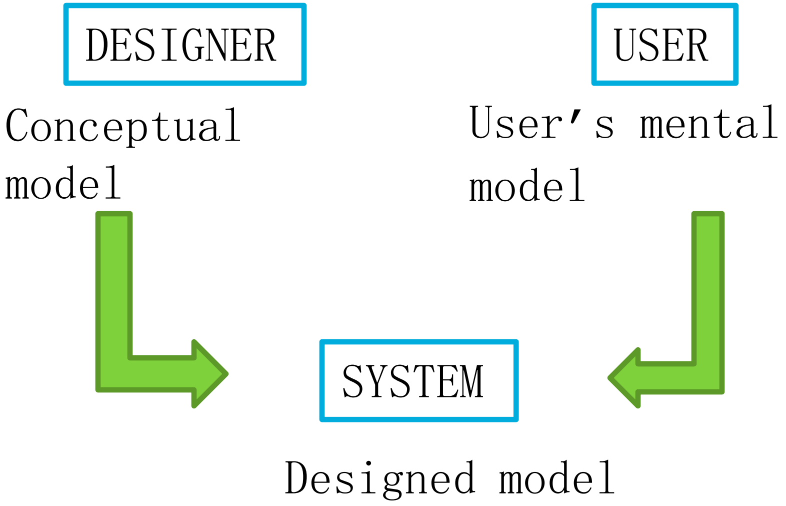


Example of ‘Visual’:
Windows 8 Interface.

4. Get the mapping right

Mapping: the link between what *user wants to do* and what *is perceived as possible - by the user based on the user’s own logic.*

User Mental Model = Conceptual Model = Designed Model



DESIGNER

Conceptual model

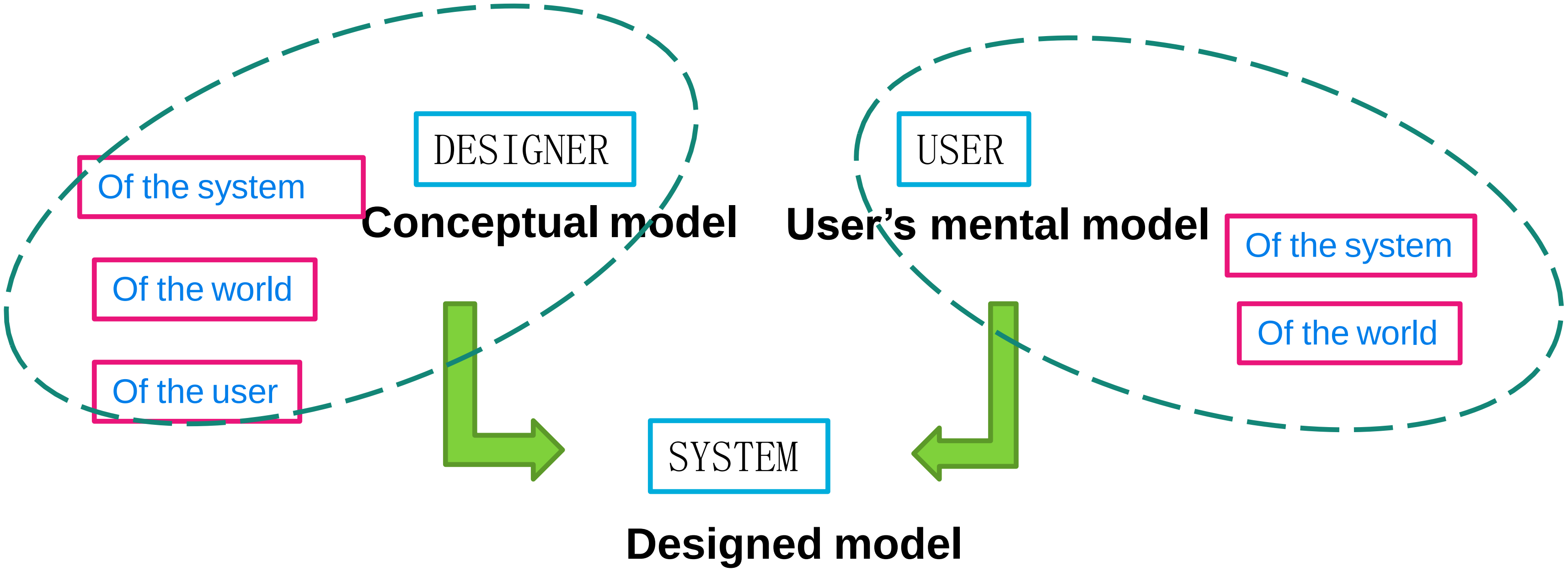
USER

User’s mental model

SYSTEM

Designed model

The three models are elaborated upon in the next slide.

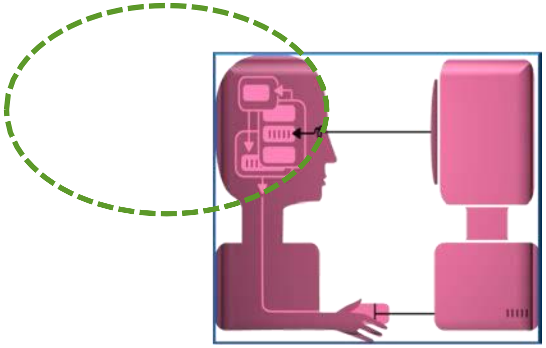


The **User's Mental Model** is the model of a system's working that a user creates when learning and using a computer. It is not technically accurate. It may also be not stable over time. User's mental models keep changing & evolving as learning continues.

In a way Mental Models are models people have of themselves, others and environment. It is their inner understanding.

The mental model of a device is formed by interpreting its perceived actions and is a visible physical structure. Some times the word 'System image model' is also used to imply the real world physical model.

The Conceptual Model.

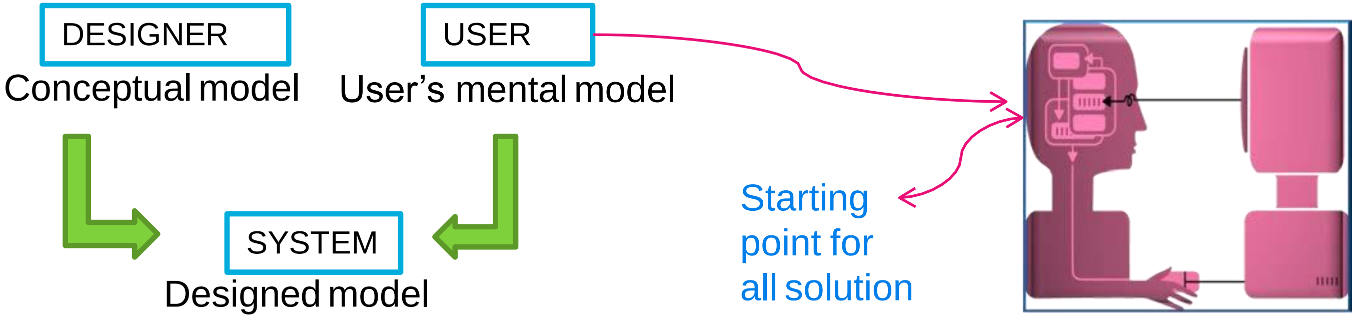


- •This is a technically accurate model of the computer / device / system created by designers / teachers/researchers for their specific internal technical use.

- •Users too have a Conceptual model but it is their mental model unless the user is as technically qualified as the evaluator. In a way as far as the user is concerned mental models and conceptual models are inherent to each other.

- •Designer's too have Mental models of the system. So a Conceptual model of the system needs to be as close as possible to the System's Image Model.

Continued…



DESIGNER

Conceptual model

USER

User’s mental model

SYSTEM

Designed model

Starting point for all solution

A good device / system will emerge when the starting point of the design process is the user- his/her mental model’ (in turn derived through user research- task analysis, walk –through, Contextual inquiry etc.) being the basis of the system image and its conceptual model.

The Conceptualised solution the Designer had in his/her mind is called the *design model* .

The User model (what the user develops in the self to explain the operation of the system) and the system image (the system’s appearance, operation way it responds is usually a blend of the users mental model and conceptual model all rolled into one. (unless the user happens to be an expert)

Ideally, the design model and user model have to be as close as possible for the systems acceptance. The designer must ensure that the system image is consistent with and operates according to the proper conceptual model.

5. Convert constraints into advantages

This is about how to ensure that the user knows what to do next when there are more than one possibility or more than one given option.

In other words how a designer needs to embed constraints in the sequence of operations in an interface such that the user is guided to the right sequence choice by reducing the chance of error in choosing the wrong option

As a principle Interfaces need to have any of the three type of constraints Physical; Technological; Cultural

Physical constraints :

Based on shape, size, area (example mouse over area demarcation)

Continued...

Cultural constraints : Culturally & semantically practiced rituals, symbols, color codes.

Ex: Always start from Right & stop on Left lower end in a word document.

Technological constraints: Example: Closing a file without saving –user needs to be warned every time this is likely to be operated by the user.

Program it such that it is mandatory to press the save button before close button.

Visibility and feedback:

Visual design also can suggest constraints.

Ex: If a number of identical buttons are required for diverse functions, Visually building differences such as colour or grouping , it is possible to ‘constrain’ a user in not pressing randomly the identical looking buttons placed in close proximity.

6. Design for Error

Errors are not taken as human faults in users in HCI.
This means Errors by users cannot be blamed on Users.
Users are not the cause for errors.
Often errors are **‘slips**’ - intend to do one thing but end up doing another accidentally.
Errors happen when there is a mismatch between User’s mental model, designers' understanding of User’s mental model; system limitations.

Research literature revels that Errors can be classified as:

Description Errors: Two objects physically alike are described / taken mistakenly for each other.
One solution employed is ‘ highlighting’ the object which is in line of next action so that ‘ attention’ is drawn to that right object from amongst similar looking group of objects.

Data Errors: Could be perception errors or selection errors. A solution could be reversal of action without penalty and ‘ affordance’ by the user to correct the error by retracing action steps.

Associative Action Errors:

Associative Errors are those that involve activating one sequence in place of another and realising it when the wrong /unexpected response results.

Associative Errors also happen when short term memory is overloaded or long term memory fails. Forgetting to do something as prescribed or reversing the sequence - Pressing the second button first instead of the first button etc.

•'Slips' & Errors need to be taken care of in Design by providing feed back (either pre or post action). Example : Prompting.

•The cause of the error needs to be understood more than the error.

• Retracing actions must be provided for.

•Assume Task to be imperfect and assume that users will always ' approximate' their actions.

7. When all else is unsuitable – Standardize.

In certain situations / contexts wherein the nature of the task is critical, the user needs to be ‘ forced’ to follow a the only choice as given (afforded) by design. Example of such situations are: Medical Devices; Warfare equipment; Nuclear Equipment; Power Plant Controls, Energy Grids; Air Traffic Controls etc.

In such critical application contexts ‘STANDARDISATION’ practice is followed.

However very stringent usability testing & evaluation practice is followed before ‘**standardising’** the format for both INPUT as well as the OUTPUT.

Standardisation comes under the ‘ Best Practice’ adaptation wherein specific rules are the basis;

Where as PRINCIPLES are abstract design rules with navigation and UCD focus , ‘ GUIDELINES’ allow more freedom to the designer.

Between ‘ Standardisation, Principles and Guidelines, the context and the level of the user (expertise) are the determining factors.

Conclusions:

Here, we discussed some of the more popular HCI norms such as:

•Norman’s seven principles of Design

were explored in detail in terms of their background, theory and applications in design.

Since ‘ design’ involves both qualitative as well as quantitative aspects - guide lines, rules & principles are more suited than absolute laws as in science.

Assignments:

1.Chose a Software interface and conduct an evaluation using Norman's seven principles.

2.Draw an' interaction model' based on Norman's model for the following Interface: Assume all data.

An interface for checking number of Leaves (absence with permission) availed off by a student and their type (Medical, Vacation; Conference visits;) . Refer to the student leave rules of your institution for necessary constraints and other relevant data.

HCI Guidelines:
Norman’s Model of Interaction

Introduction

Let us first understand the word “ INTERACTION”

All man-made objects offer the possibility for interaction.

When an object is designed for a purpose (function) it affords interaction. Interaction is a way of framing the relationship between people and objects designed for them.

Interaction is thus a way of framing the relationship between the object & User.

All Design activities can be viewed as design for interaction. In fact not only objects but space & messages (communication) too involve interaction. Interaction is a key aspect of function, and function is a key aspect of design.

However often one notices that designers often use the word ‘INTERACTION’ rather carelessly .

Untrained Designers often tend to confuse ‘Interaction’ with ‘Reaction’.

For example: Designers claim to be designing “Interactive web pages”.

The fact is clicking on links to navigate to a new webpage is NOT an “INTERACTION”.
It is ‘reaction’ of input by the hyperlinked pages. The computer is automatically reacting to input because it has been programmed to do so.
This programmed action couples ‘ input’ to ‘output’ in a fixed way.

Interaction is however a dynamic action that through a dialogue (involving feed back) adjusts to input and gives appropriate output.

In HCI – the feed back loop model of interaction treats a person as closely coupled with a dynamic system.

In HCI – Interaction is simply stated as two way communication between the “User and System”

USER

SYSTEM
(computer / device / object / cyberspace)

It should however be noted that due to the human complex cognitive system, representing interaction between a person and a dynamic system as a simple feedback loop can only be a good first approximation.

Definitions of some Terms of Interaction

Domain: **expertise, knowledge in some real world Activity. In GUI domain concepts such as geometric shape, colour, Symbols etc. are involved.**

Task: **operation to manipulate concepts in a domain.**

Goal: **desired output from a performed task.**
Ex in GUI: A button

Intention: **specific action required to meet the goal**

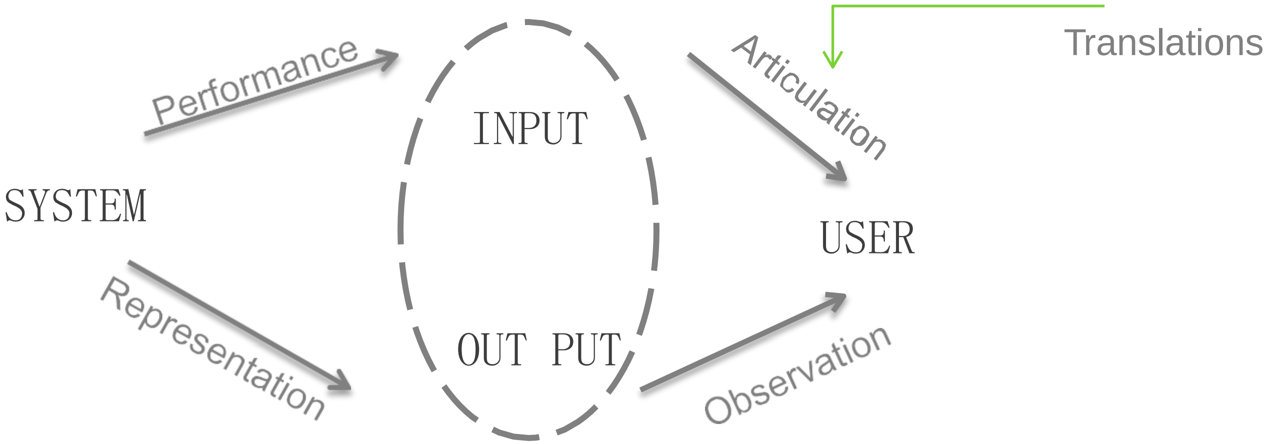
Task analysis: **Study of the problem space**

In HCI interaction models are translations between user and system

There are different Interaction Models mentioned in HCI

- Donald Norman’s Interaction Model
- Abowd & Beale’s model

A generalised Interaction Model (from Dix et al) has four components:
(i) System; (ii) User; (iii) Input & (iv) Output.



There are different Interaction Styles (nature of the dialogue)

And there are different Interaction Contexts
(Social, Organizational, Educational, Commercial etc.)

We will discuss Donald Norman’s Interaction Model in the following slides

Norman’s Model of Interaction

Donald Norman’s Interaction model concentrates on the Users Thought processes and accompanying actions.

Norman proposed that actions are performed by the users in cycle such as

(i) Establishing a goal
(ii) Executing the action
(iii) Evaluating the results

Given a need a user sets about achieving the goal of fulfilling the needs.

A series of actions are performed – one leading to another – till the result expected is obtained.

Norman’s Model of Interaction consists of seven stages as follows:

User to establish **Goal**

User to formulate **Intention**

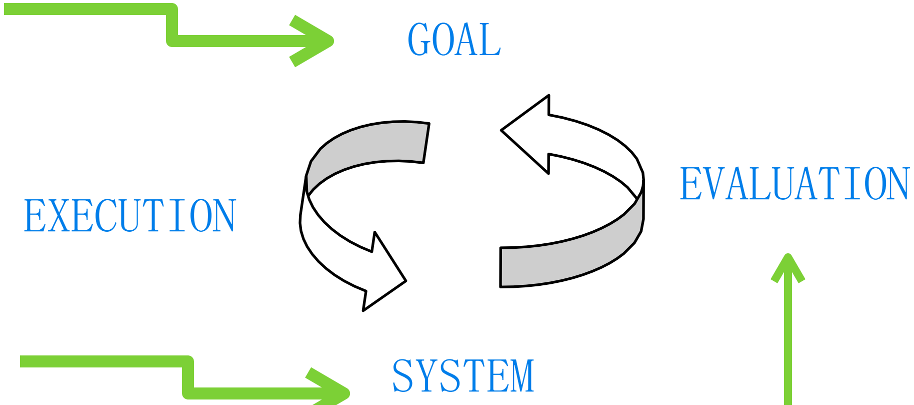
Decides on action

Executes the action

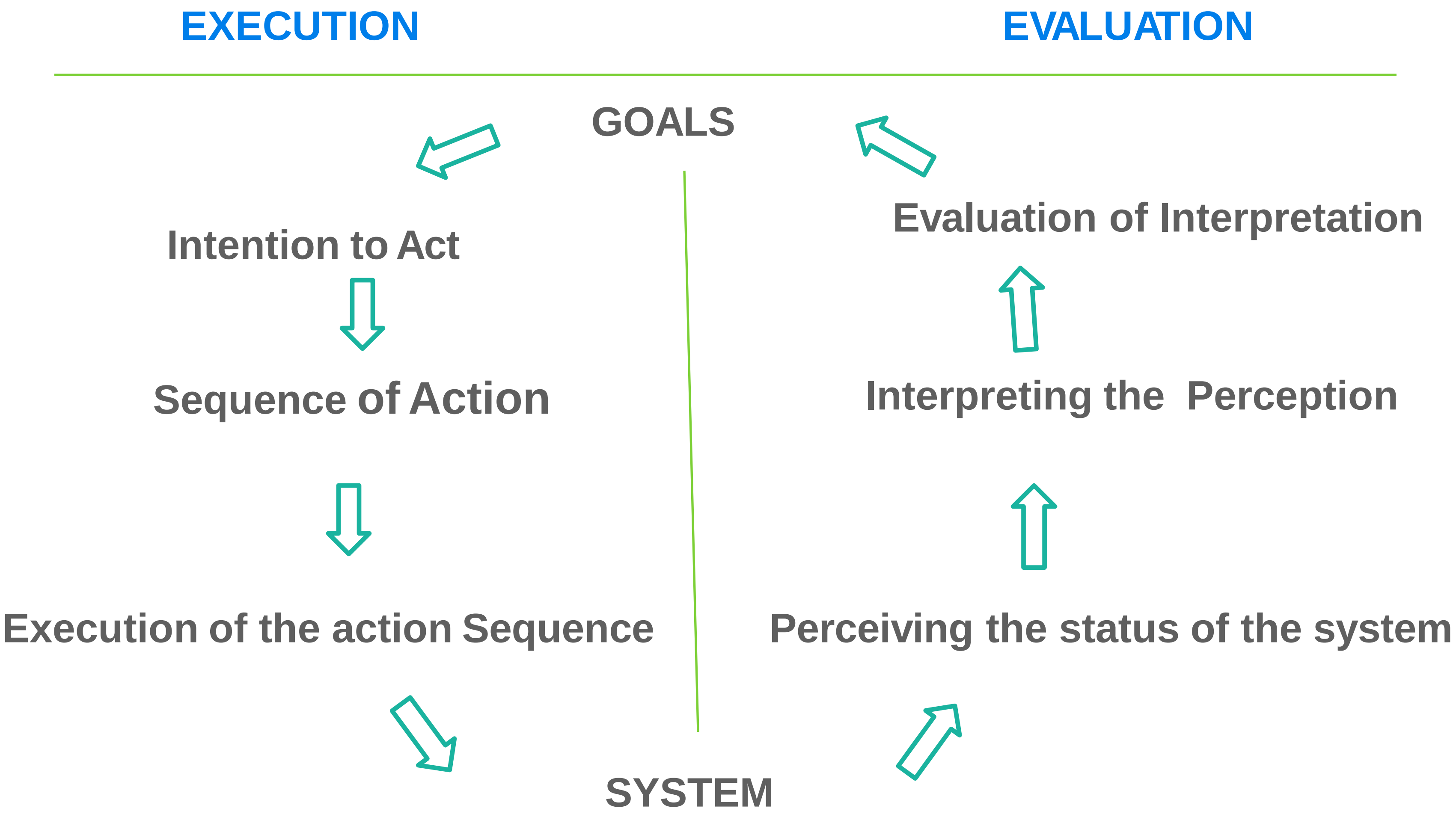
Perceives system state

Interprets system response

Evaluates system with respect to goal.



Another way of depicting Norman's 7 stage Action model



EXECUTION

EVALUATION

GOALS

Intention to Act

Sequence of Action

Execution of the action Sequence

SYSTEM

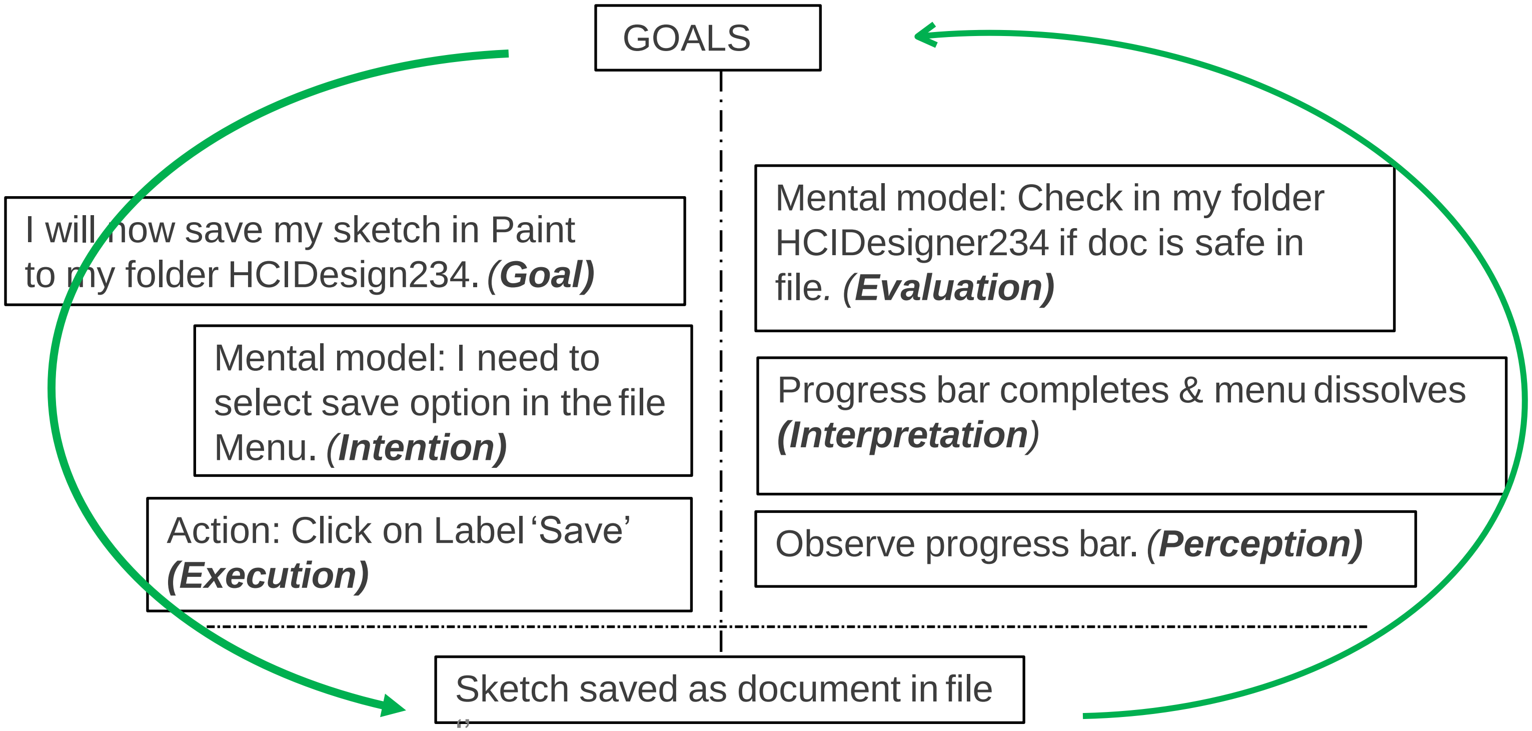
Perceiving the status of the system

Interpreting the Perception

Evaluation of Interpretation

Understanding Norman's Model with Example:

Need: Documenting work done
Task: Save My Sketch
Goal: Safely store the sketch in a place which we can fetch it from



GOALS

I will now save my sketch in Paint to my folder HCIDesign234. ***(Goal)***

Mental model: I need to select save option in the file Menu. *(**Intention)***

Action: Click on Label ‘Save’ ***(Execution)***

Mental model: Check in my folder HCIDesigner234 if doc is safe in file*.* *(**Evaluation)***

Progress bar completes & menu dissolves *(**Interpretation**)*

Observe progress bar. ***(Perception)***

Sketch saved as document in file

As a basis for his Interaction Model Norman proposed the following levels of abstraction of knowledge of the user

- **Task Level**
- **Goal Level**
- **Semantic level**
- **Syntax level**
- **Lexical level**
- **Physical Level**

Task Level: task level is to analyze the user's needs and to structure the task domain in such a way, that a computer system can play a part in it. The task level describes the structure of the tasks which can be delegated to the computer system.

Semantic level describes the set of objects, attributes, and operations, the system and the user can communicate. Semantics is about how the user interprets it and makes meanings out of the system.

Syntax level describes which conceptual entities and operations may be referred to in a particular command context or system state.

Lexical level: language, wording.

Norman’s HCI model consists of three types:

User’s Mental Model; **System Image Model**; **Conceptual Model**.

The **User’s Mental Model** is the model of a machine’s working that a user creates when learning and using a computer. It is not technically accurate. It may also be not stable over time.

User’s mental models keep changing, evolving as learning continues.

In a way Mental Models are models people have of themselves, others and environment.

The mental model of a device is formed by interpreting its perceived actions and its visible structure.

The **System image Model** is the visible physical part of the computing system / device.

The Conceptual Model.

This is the technically accurate model of the computer / device / system created by designers / teachers/researchers for their specific internal technical use.

Users too have a Conceptual model but it is their mental model unless the user is a technically qualified as the evaluator. In a way as far a the user is concerned mental models and conceptual models are inherent to each other. Designer’s too have Mental models of the system. So a Conceptual model of the system needs to be as close as possible to the System’s Image Model.

The User model *(what the user develops in the self to explain the operation of the system)* and the system image *(the system’s appearance, operation way it responds)* is usually a blend of the **users mental model and conceptual model** all rolled into one.

Interaction Model and Device / System Design

A good device / system will emerge when the starting point of the design process is the user- his/her mental model' (in turn derived through user research- task analysis, walk thoughts Contextual inquiry etc.) being the basis of the system image and its conceptual model

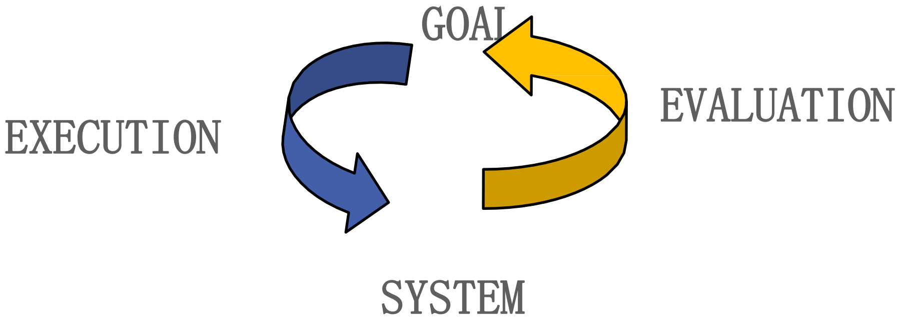
The Conceptualisation of the Designer had in his/her mind is called the *design model* .

Ideally, the design model and user model have to be as close as possible for the systems acceptance.

The designer must ensure that the system image is consistent with and operates according to the proper conceptual model.

Norman applies the "Gulf Model" to explain why some interfaces cause problems to the users.

Norman uses the terms “**Gulf of Execution’** and ‘ **Gulf of Evaluation’**.
Norman's model (also some times referred to as the Gulf Model) is useful in understanding the reasons of interface failures from the users point of view. The Seven stages of action model is an elaboration of the Gulf model.

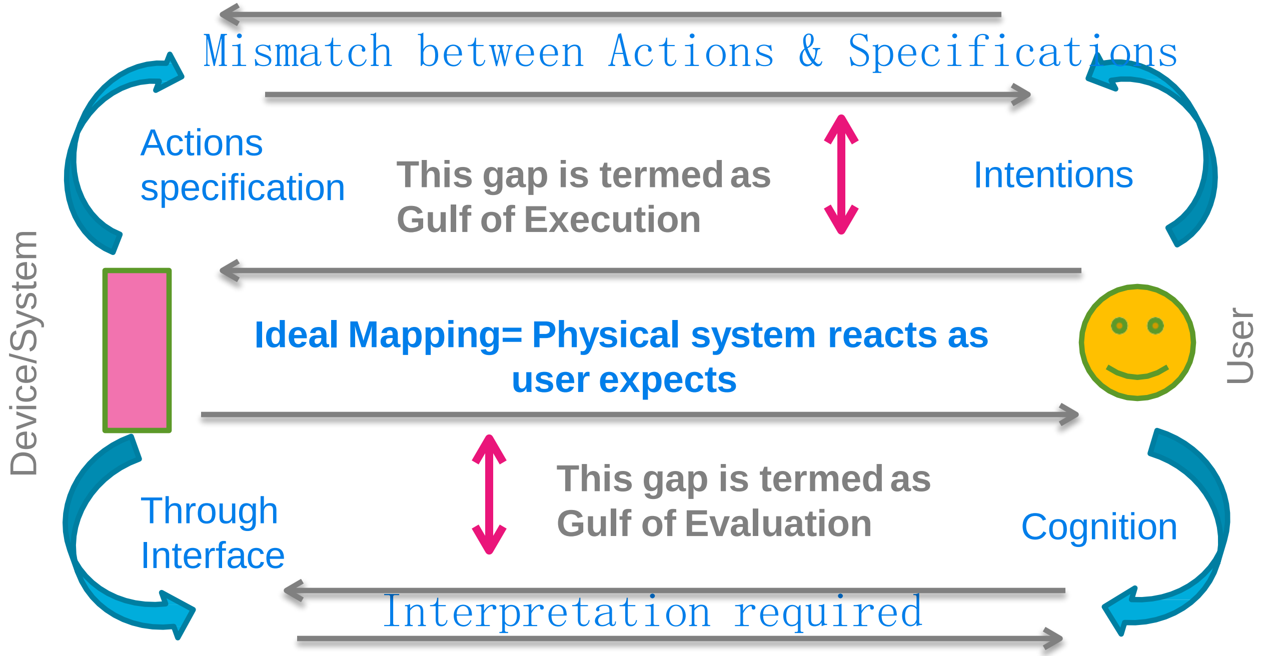


Gulf of Execution represents the difference between user’s formulation of the action to reach their goals and the actions allowed by the system.

User’s formulation of action ≠ Actions allowed by the system.

The Gulf of Evaluation is the difference between physical presentation of system state and the expectations of the user.

User's Expectation ≠ System's presentation.



Mismatch between Actions & Specifications

Actions specification

This gap is termed as Gulf of Execution

Intentions

Device/System

Ideal Mapping= Physical system reacts as user expects

User

Through Interface

This gap is termed as Gulf of Evaluation

Cognition

Interpretation required

Interaction Styles

Having understood Interaction Frame work as a model let us
Look at different Interaction Styles

Some common interaction styles

- Command line interface
- Menus
- Natural language
- Question/answer and Query dialogue (Ex: SQL)
- Form-fills and spreadsheets
- WIMP – [Windows; Icons; Menus; pointers]

- Three–dimensional interfaces
- Gestural Interfaces
- Vice operated commands
- Thought (mind) operated commands (Evolving rapidly at Laboratory level)

Conclusions

Interaction models are conceptualisations of the process of Interaction between the user and the system.

Norman's seven stage Interaction model explains interactivity from the user’s point of view.

There is a gulf (gap) between (i) EXECUTION and (ii) EVALUATION

There could be a number of reasons why an interaction can fail at a number of points in the dialogue.

The interaction model can be a useful tool for analysing as well as conceptualising dialogue between a system and user.

Assignments:

Draw the Users Mental Model for a Transfer of Money from one account to another on an ATM

Using Norman's seven principles draw a Norman's Interaction Diagram for 2 Tasks in any application software of your choice.

References:

Shneiderman; Designing the user Interface: strategies for effective human computer interaction. Addison –Wesley, Reading MA 1987.

Nielsen, Enhancing the exploratory power of usability heuristics. Proceedings of the ACM CHI’94 Conference.

D. A. Norman; The Design of Every Day Things. Doubleday, New York1988.

Dix.A, Finlay J; Abowd G.D & Beale R; Human Computer Interaction, 3rd Edition, Pearson Education 2005.