Human Computer Interaction - An Introduction

Dr. Pradeep Yammiyavar
Professor,
Dept. of Design,
IIT Guwahati,
Assam, India

Dr. Samit Bhattacharya
Assistant Professor,
Dept. of Computer Science
and Engineering,
IIT Guwahati, Assam, India

Indian Institute of Technology Guwahati
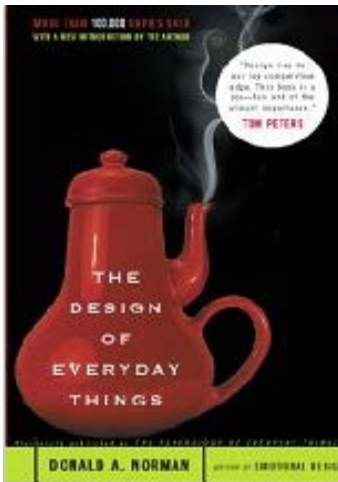
Module 4:
Guidelines in HCI

Lecture 2:
Norman's seven principles

Dr. Pradeep Yammiyavar

Introduction:

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

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

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 constrains into advantages
   (Physical constraints, Cultural constraints, Technological constraints)

6. Design for Error

7. When all else fails – Standardize.

Each of these seven principles is discussed in the following slides.

Discussions on Normans 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 <u>abstraction of knowledge of the user</u> :

  - 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 cognation.
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 ?

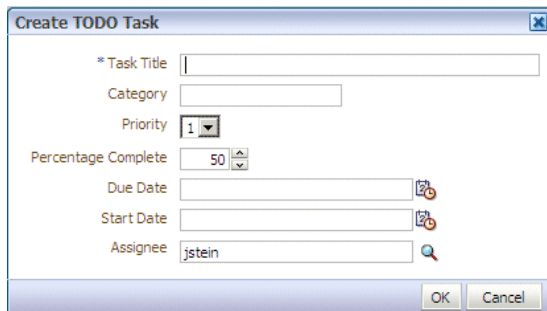Relying on either of them alone would lead to an incomplete abstraction of knowledge by the Designer.
Hence Norma's principle mandates that both types of knowledge be considered.

• **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 langue easy for a HCI designer to build Interactive Interfaces & Hierarchies.
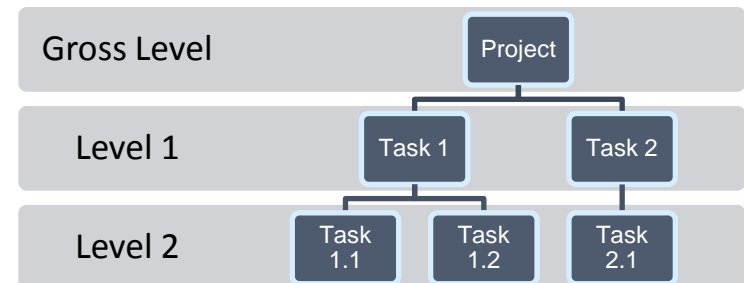
Example: Gross level & Broken Down level of a GUI

Create TODO Task

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

OK   Cancel

Gross Level	Project
Level 1	Task 1 Task 2
Level 2	Task 1.1 Task 1.2 Task 2.1

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.*

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':
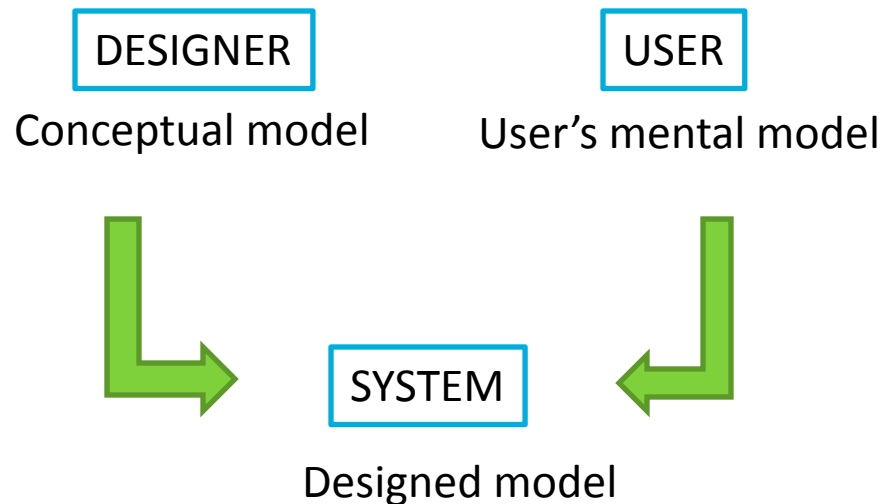Windows 8 Interface.
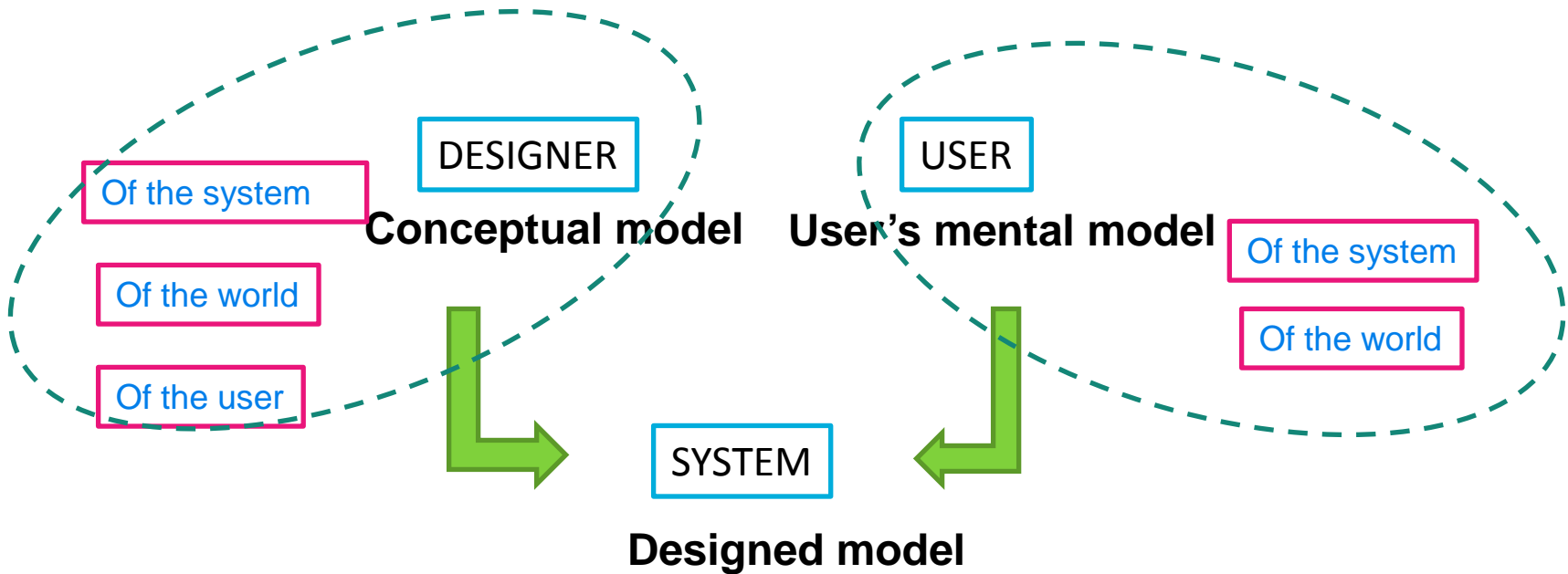
4. Get the mapping right

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	USER
Conceptual model	User's mental model

SYSTEM

Designed model

The three models are elaborated upon in the next slide.

```
┌──────────┐                        ┌────────┐
│ DESIGNER │                        │  USER  │
└──────────┘                        └────────┘
┌───────────────┐                                    ┌───────────────┐
│ Of the system │   Conceptual model  User's mental  │ Of the system │
└───────────────┘                      model         └───────────────┘
┌───────────────┐                                    ┌───────────────┐
│ Of the world  │                                    │ Of the world  │
└───────────────┘                                    └───────────────┘
┌───────────────┐           ┌────────┐
│ Of the user   │           │ SYSTEM │
└───────────────┘           └────────┘

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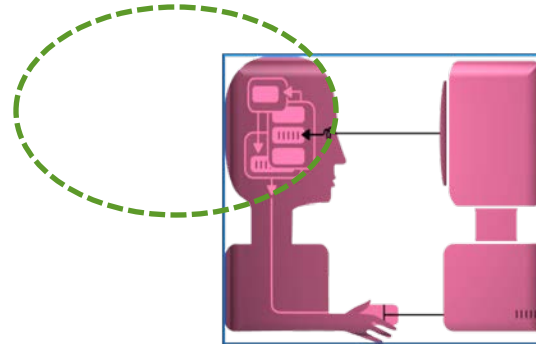
The **User's Mental Model** is the model of a system's working that a user creates when learning and using a computer. It is not technically accurate. It may also be not stable over time. User's mental models keep changing & evolving as learning continues.

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.

DESIGNER
Conceptual model

USER
User's mental model

SYSTEM
Designed model

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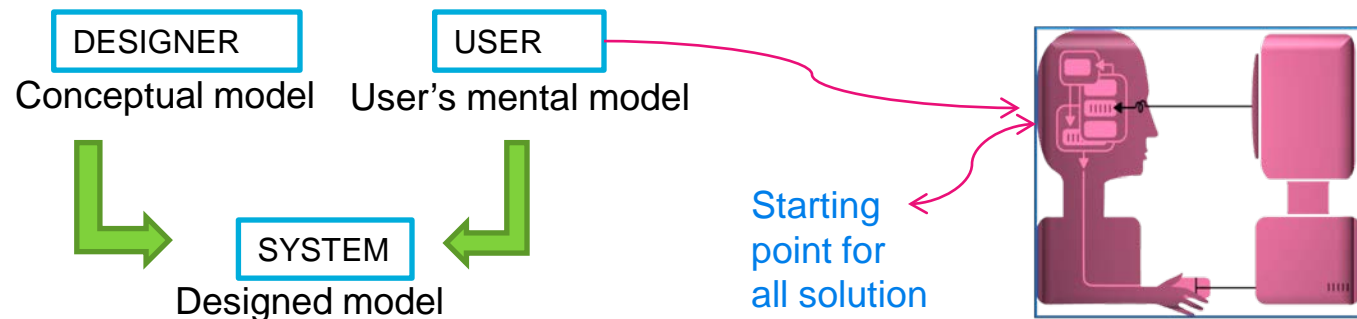
 A good device / system will emerge when the starting point of the design process is the user- his/her mental model' (in turn derived through user research- task analysis, walk –through, Contextual inquiry etc) being the basis of the system image and its conceptual model.

 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 constrains 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)

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 <u>Errors by users cannot be blamed on Users.</u>
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:

In this Lecture 2 of Module 4 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.

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 york 1988.

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

Assignment:

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.