

# A House



- Without doors, windows!!!
- Utility connections!!

-Similarly, a software without any interfaces is of no use!!

# Interfaces

- Between Components
  - E.g. A gas pipe from kitchen to living room
  - E.g. A class calling method of another class
- Between Self and External Entities
  - E.g. Gas connection from SNGPL/SSGPL, water connection from WASA
  - E.g. Our software interacting with bank, NADRA
- Between Self and Human
  - E.g. Door bell, button to switch on a tube light/fan
  - E.g. A User clicking on print button, login button, post button on screen



# Interfaces

- Between Components (Software Components)
  - E.g. A gas pipe for
  - E.g. E.g. A class
- Between Self and nonhuman product
  - E.g. Gas connected from WASA, service
  - E.g. Our software
- Between Self and Human (Software and Human, User Interface)
  - E.g. Door bell, button to switch on a tube light/fan
  - E.g. A User clicking on print button, login button, post button on screen

– A User Interface can be:

- Command Line

- Graphical

# GUIs

- Reduced many interfacing problems
- Found to be difficult to learn, hard to use, confusing in some cases?
  - Any experiences?
- UIs should be easy to:
  - Learn
  - Use
  - Understand



# Designing User Interface

- An effective communication medium between a human and a computer
- Creation of a screen layout
- Study of people and how they relate to technology by answering questions like:
  - Who is the user?
  - How does the user learn to interact with the system?
  - How does the user interpret info produced by the system?
  - What will the user expect of the system?

# Golden Rules

- Place the User in Control
- Reduce User's Memory Load
- Make the Interface Consistent



# Place the user in control

- System should react to user needs
- System should help the user complete tasks
- User should not feel that the system is controlling the user
- e.g OS commands

# Place the user in control

- Design Principles:
  - Define interactions s.t. a user is not forced into unnecessary/undesired actions/modes
  - Provide flexible interaction
  - User should not feel that the system is controlling the user
  - Allow interruptible and undoable user interactions
  - Streamline interactions based on skill level, allow interactions to be customized
  - Hide technical internals from casual user
  - Provide mechanism for direct interaction with objects on screen



# Reduce the user's memory load

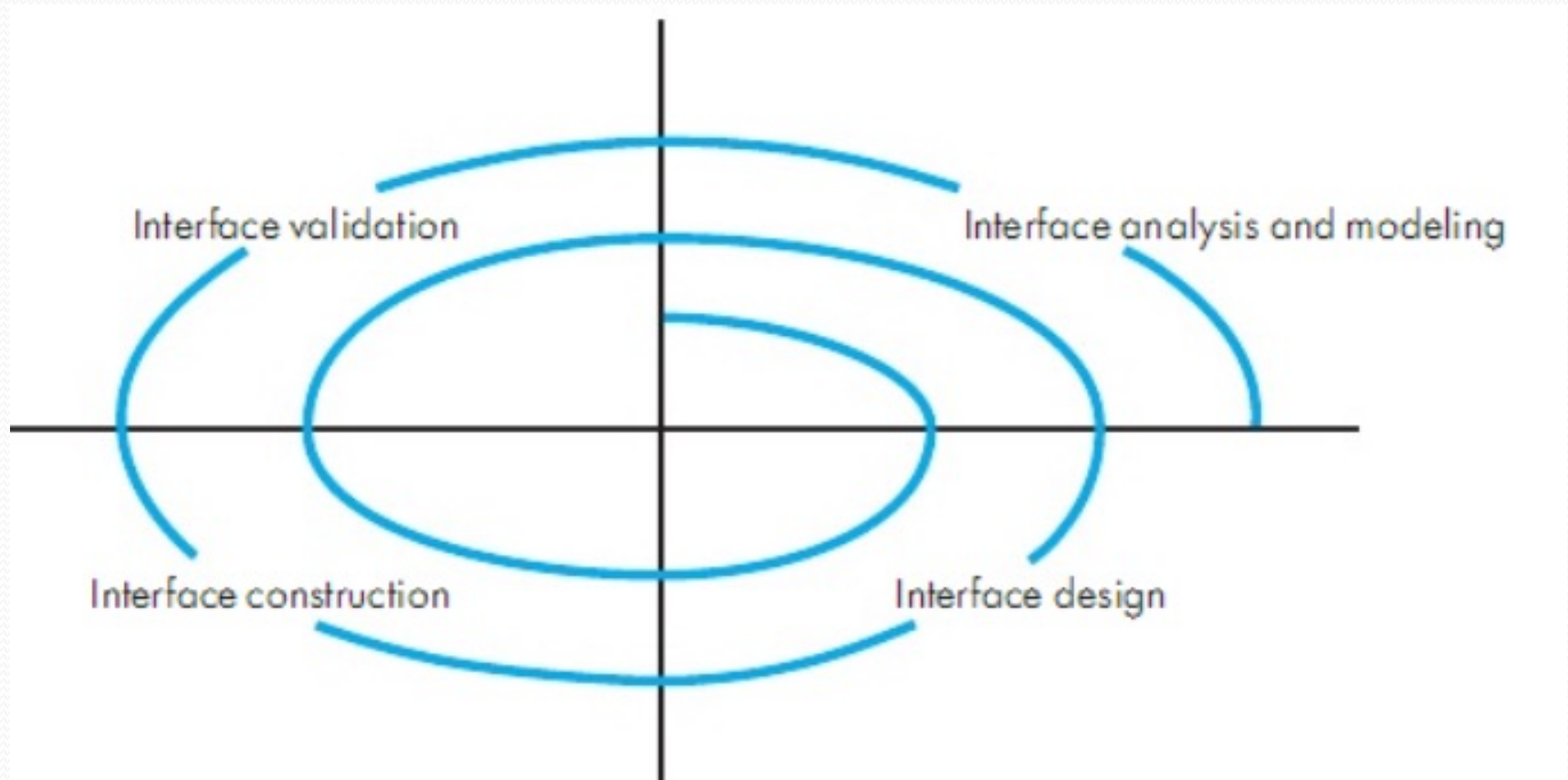
- The more a user has to remember, the more error-prone the interaction
- Design principles:
  - Reduce demand on short term memory
  - Establish meaningful defaults example reset
  - Define shortcuts that are intuitive: Alt:P
  - Disclose information in a progressive manner. Example  
Select text and show text styles

# Make the interface consistent

- Maintain design rules for all screens
- Design principles:
  - Allow user to put current task into a meaningful context:  
graphical icons
  - Maintain consistency across a complete product line:  
CNTRL+C
  - Avoid violating de facto standards



# UID Process



# UID Process

- Interface Analysis
  - User Model
    - Profile of end user
      - Age, gender, education, physical abilities, cultural/ethnic background etc.
    - Novice, Knowledgeable, Knowledgeable frequent users
  - Design Model
  - Mental Model
    - User's perception of the system
  - Implementation Model



# UID Process

- Interface Design
  - Define interface objects and actions
- Interface Construction
  - Creation of prototypes to evaluate usage scenarios
- Interface Validation
  - Ability to implement every user task correctly
  - Degree to which interface is easy to use
  - User's acceptance to user interface as useful

# Interface Analysis

- User Analysis
  - Trained professionals? Technicians? Clerks? Etc.
  - Average level of formal education?
  - User capability to learn from written material/training?
  - Expert typists? Do not like keyboard?
  - Age range?
  - Users represented predominantly by one gender?
  - Routine of work? Regular? Overtime?
  - Frequency of usage?
  - Primary spoken language of users?
  - Subject matter experts?
  - Desire to know underlying technology



# Interface Analysis

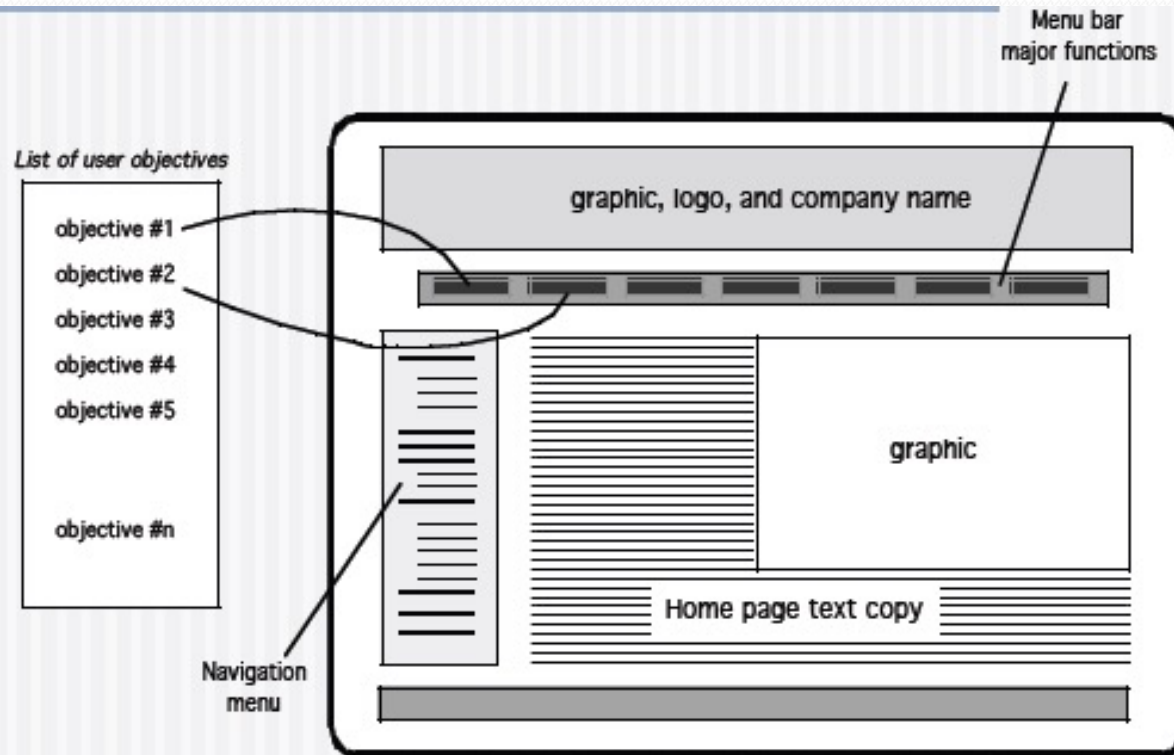
- Task Analysis
  - Work performed by user in particular circumstances
  - Tasks/subtasks performed by the user during the work being performed
  - Problem domain objects manipulated by users
  - Sequence of work tasks? Flow of actions? The workflow?
  - Hierarchy of tasks
- Techniques used to support task analysis
  - Usecases
  - Task Elaboration(Functional decomposition)
  - Object Elaboration(Classes and attributes)
  - Workflow Analysis

# Interface Analysis

- Analysis and display content
  - Partitioning of large objects (e.g. reports) for better understanding
  - Use of colors to improve understanding
  - Content of error messages?



# Mapping User Objectives



# Design Issues

- Response time
- Help
  - Help without leaving the interface?
- Error information handling



# References

- Pfleeger Book slides from UCF
- Bruegge et al. OOSE book
- Book and slides from Pressman
- <http://www.authorstream.com/Presentation/umr17-1645231-pressman-ch-12-user-interface-design/>

# Acknowledgement!

- Some slides have been reused from UCF slides for the SE course, specially for UI design
- Majorly, flow of UCF slides has been modified and a few examples have been added