User Interface Design

Instructor: Mehroze Khan

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

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
- Identification of interface objects and actions
- 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?

User Interface

- As technologists studied human interaction, two dominant issues arose.
  - First, a set of ***golden rules*** were identified. These applied to all human interaction with technology products.
  - Second, a set of ***interaction mechanisms*** were defined to enable software designers to build systems that properly implemented the golden rules.
- These interaction mechanisms, collectively called the ***user interface****,* have eliminated many problems associated with human interfaces.

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

Place the user in control

- Design Principles:
  - Define interactions such that 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
  - Define shortcuts that are intuitive
  - Visual layout must be based on real world metaphor
  - Disclose information in a progressive manner

Make the interface consistent

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

References

1. Roger S. Pressman, Software Engineering A Practitioner’s Approach, 9th Edition. McGrawHill