

# Software Engineering

Sobia Iftikhar  
[Sobia.Iftikhar@nu.edu.pk](mailto:Sobia.Iftikhar@nu.edu.pk)

Week 09



Class 25  
14-APRIAL-2021

# Interface Analysis

- understand the problem before you attempt to design a solution. In the case of user interface design, understanding the problem means understanding
  - (1) the people (end users) who will interact with the system through the interface,
  - (2) the tasks that end users must perform to do their work,
  - (3) the content that is presented as part of the interface,
  - (4) the environment in which these tasks will be conducted.

# Elements Of Interface Analysis

1. User Analysis
2. Task Analysis and Modelling
3. Analysis of Display Content
4. Analysis of the Work Environment

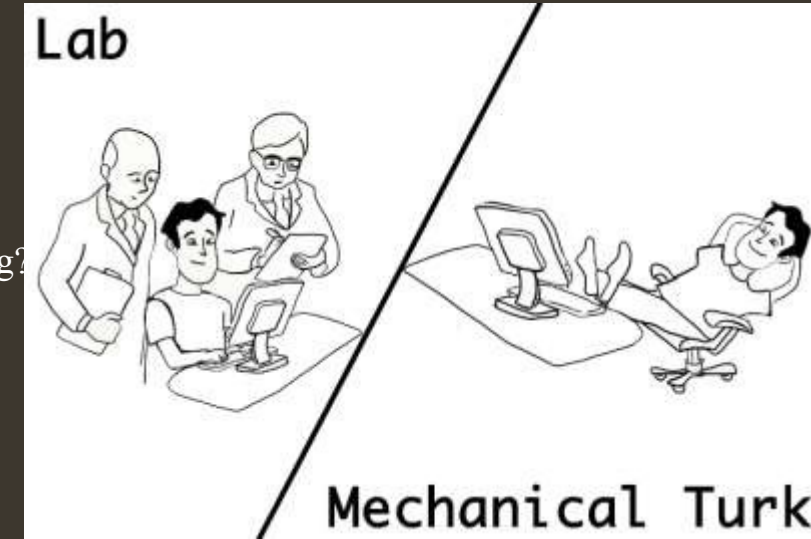
# 1. User Analysis

- The phrase “user interface” is probably all the justification needed to spend some time understanding the user before worrying about technical matters. .
- each user has a mental image of the software that may be different from the mental image developed by other users.
  - **User Interviews.** The most direct approach, members of the software team meet with end users to better understand their needs, motivations, work culture, and a myriad of other issues. This can be accomplished in one-on-one meetings or through focus groups.
  - **Sales input.** Sales people meet with users on a regular basis and can gather information that will help the software team to categorize users and better understand their requirements.
  - **Marketing input.** Market analysis can be invaluable in the definition of market segments and an understanding of how each segment might use the software in subtly different ways.
  - **Support input.** Support staff talks with users on a daily basis. They are the most likely source of information on what works and what doesn't, what users like and what they dislike, what features generate questions and what features are easy to use.



# The following set of questions (adapted from [Hac98]) will help you to better understand the users of a system:

- Are users trained professionals, technicians, clerical, or manufacturing workers?
- What level of formal education does the average user have?
- Are the users capable of learning from written materials or have they expressed a desire for classroom training?
- What is the age range of the user community?
- Will the users be represented predominately by one gender?
- Do users work normal office hours or do they work until the job is done?
- Is the software to be an integral part of the work users do or will it be used only occasionally?
- What is the primary spoken language among users?
- What are the consequences if a user makes a mistake using the system?
- Are users experts in the subject matter that is addressed by the system? Do users want to know about the technology that sits behind the interface?



## 2. Task Analysis and Modelling

- The goal of task analysis is to answer the following questions:
  - What work will the user perform in specific circumstances?
  - What tasks and subtasks will be performed as the user does the work?
  - What specific problem domain objects will the user manipulate as work is performed?
  - What is the sequence of work tasks—the workflow?
  - What is the hierarchy of tasks?

## 2. Task Analysis and Modelling (cont)

- Techniques Are Applied To The User Interface.
  - Use Case
  - Task elaboration.
  - Object elaboration
  - Workflow analysis



## 2. Task Analysis and Modelling (cont.)

- *Use Case*
  - Show how an end user performs some specific work-related task
  - Enable the software engineer to extract tasks, objects, and overall workflow of the interaction
  - Helps the software engineer to identify additional helpful features
- *Task elaboration.*
  - functional decomposition or stepwise refinement as mechanism for refining the processing tasks that are required for software to accomplish some desired function.
  - For example, using information contained in the use case, furniture layout can be refined into the following tasks:
    - (1) draw a floor plan based on room dimensions,
    - (2) place windows and doors at appropriate locations,
    - (3a) use furniture templates to draw scaled furniture outlines on the floor plan,
    - (3b) use accents templates to draw scaled accents on the floor plan,
    - (4) move furniture outlines and accent outlines to get the best placement,
    - (5) label all furniture and accent outlines,
    - (6) draw dimensions to show location,
    - (7) draw a perspective-rendering view for the customer

## 2. Task Analysis and Modelling (cont.)

- *Object elaboration*
  - To identify the classes
  - To identify the objects
- For example, the furniture template might translate into a class called Furniture with attributes that might include size, shape, location, and others. The interior designer would select the object from the Furniture class, move it to a position on the floor plan (another object in this context), draw the furniture outline, and so forth.

## 2. Task Analysis and Modelling (cont.)

- ***Workflow analysis***
  - 1. Each user implements different tasks via the interface; therefore, the look and feel of the interface designed for the patient will be different than the one defined for pharmacists or physicians.
  - 2. The interface design for pharmacists and physicians must accommodate access to and display of information from secondary information sources (e.g., access to inventory for the pharmacist and access to information about alternative medications for the physician).
  - 3. Many of the activities noted in the Swimlane diagram can be further elaborated using task analysis and/or object elaboration (e.g., Fills prescription could imply a mail-order delivery, a visit to a pharmacy, or a visit to a special drug distribution center).
- ***Hierarchical representation.***
- For example, consider the following user task and subtask hierarchy.
- User task: Requests that a prescription be refilled
  - • Provide identifying information.
    - • Specify name.
    - • Specify userid.
    - • Specify PIN and password. • Specify prescription number. • Specify date refill is required.

### 3. Analysis of Display Content

- The user tasks identified lead to the presentation of a variety of different types of content.
- For modern applications, display content can range from character-based reports (e.g., a spreadsheet), graphical displays (e.g., a histogram, a 3-D model, a picture of a person)

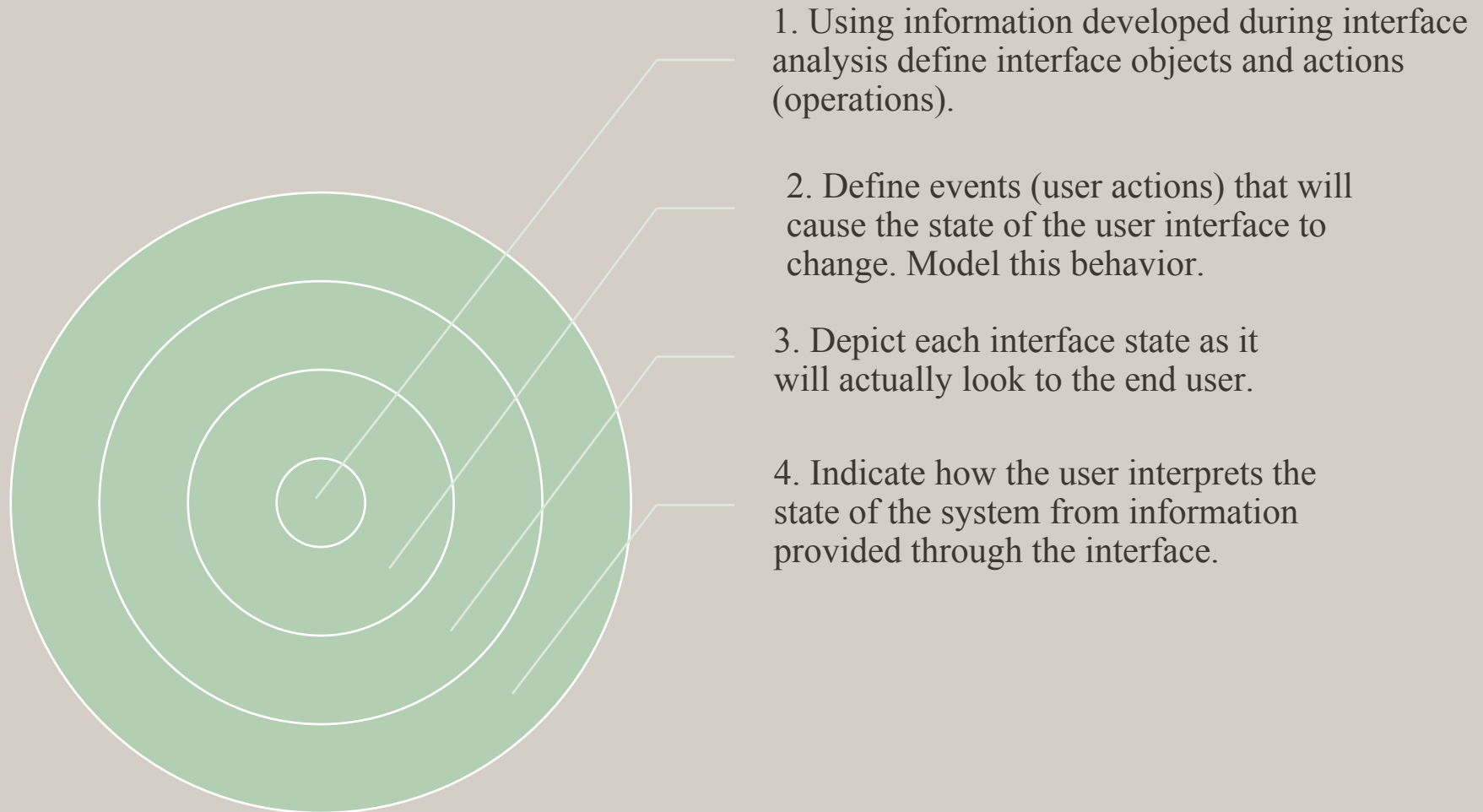
## 4. Analysis of the Work Environment

- People do not perform their work in isolation. They are influenced by the activity around them.
- In some applications the user interface for a computer-based system is placed in a “user-friendly location” (e.g., proper lighting, good display height, easy keyboard access), but in others (e.g., a factory floor or an airplane cockpit), lighting may be suboptimal, noise may be a factor.
- Will system interaction be measured in some manner (e.g., time per transaction or accuracy of a transaction)?
- Will two or more people have to share information before an input can be provided?
- How will support be provided to users of the system?

Class 26  
15-APRIAL-2021



# Interface Design Step



# During all of these steps, the designer must

- 1) always follow the golden rules discussed,
- (2) model how the interface will be implemented
- (3) consider environment (e.g., display technology, operating system, development tools) that will be used the environment

# Interface Design Step

- Applying Interface Design Steps
  - Interface objects and actions are obtained from a grammatical parse of the use cases and the
    - software problem statement
  - Interface objects are categorized into types: source, target, and application
    - A source object is dragged and dropped into a target object such as to create a hardcopy of a report
    - An application object represents application-specific data that are not directly manipulated as part of
      - screen interaction such as a list
  - After identifying objects and their actions, an interface designer performs screen layout which involves
    - Graphical design and placement of icons
    - Definition of descriptive screen text
    - Specification and titling for windows
    - Definition of major and minor menu items
    - Specification of a real-world metaphor to follow

# Example

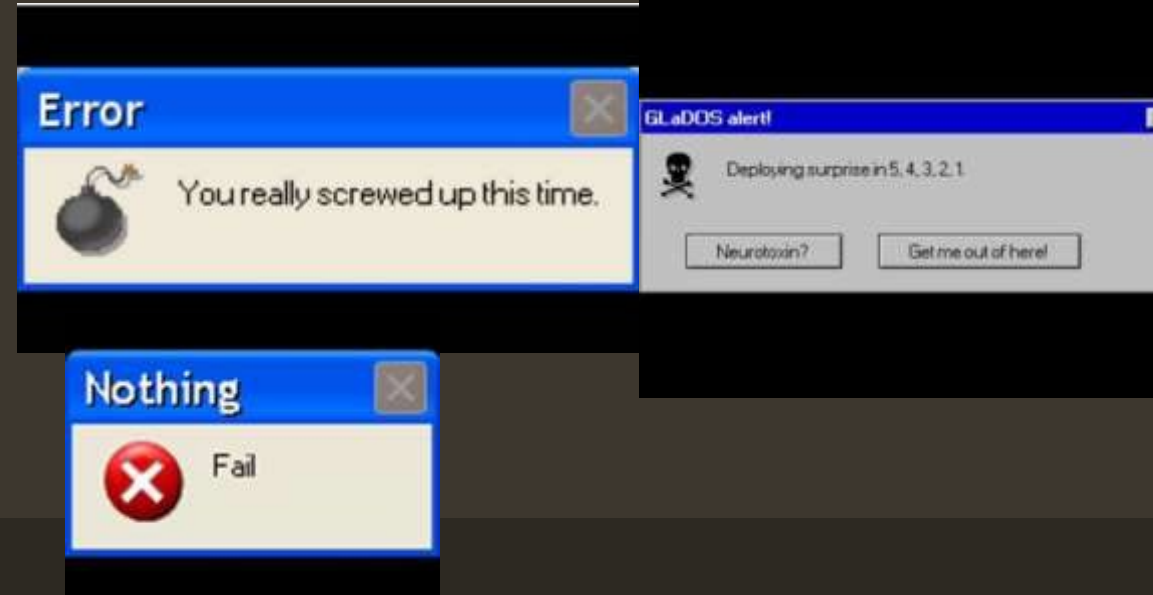
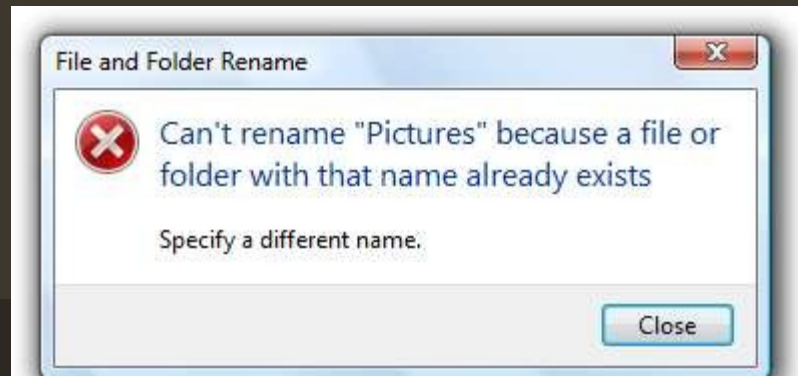
Objects (boldface) and actions (italics) are extracted from this list of homeowner tasks

- *accesses* the **SafeHome** system
- *enters* an **ID** and **password** to allow remote access
- *checks* **system status**
- *arms* or *disarms* **SafeHome** system
- *displays* **floor plan** and **sensor locations**
- *displays* **zones** on floor plan
- *changes* **zones** on floor plan
- *displays* **video camera locations** on floor plan
- *selects* **video camera** for viewing
- *views* **video images** (four frames per second)
- *pans* or *zooms* the **video camera**

- Four common design issues usually surface in any user interface
  - System response time (both length and variability)
  - User help facilities
    - When is it available, how is it accessed, how is it represented to the user, how is it structured, what happens when help is exited
  - Error information handling
  - How meaningful to the user, how descriptive of the problem
  - Menu and command labeling (more on upcoming slide)
    - Consistent, easy to learn, accessibility, internationalization

# Guidelines for Error Messages

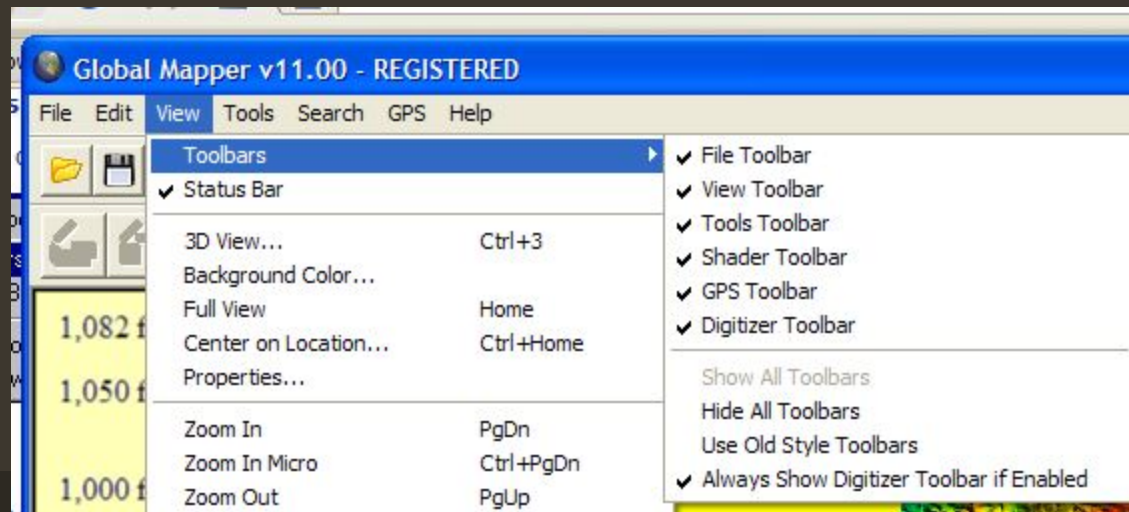
- The message should describe the problem in plain language that a typical user can understand
- The message should provide constructive advice for recovering from the error
- The message should indicate any negative consequences of the error (e.g., potentially corrupted data files) so that the user can check to ensure that they have not occurred (or correct them if they have)
- The message should be accompanied by an audible or visual cue such as a beep, momentary flashing, or a special error color
- The message should be non-judgmental
  - The message should never place blame on the user



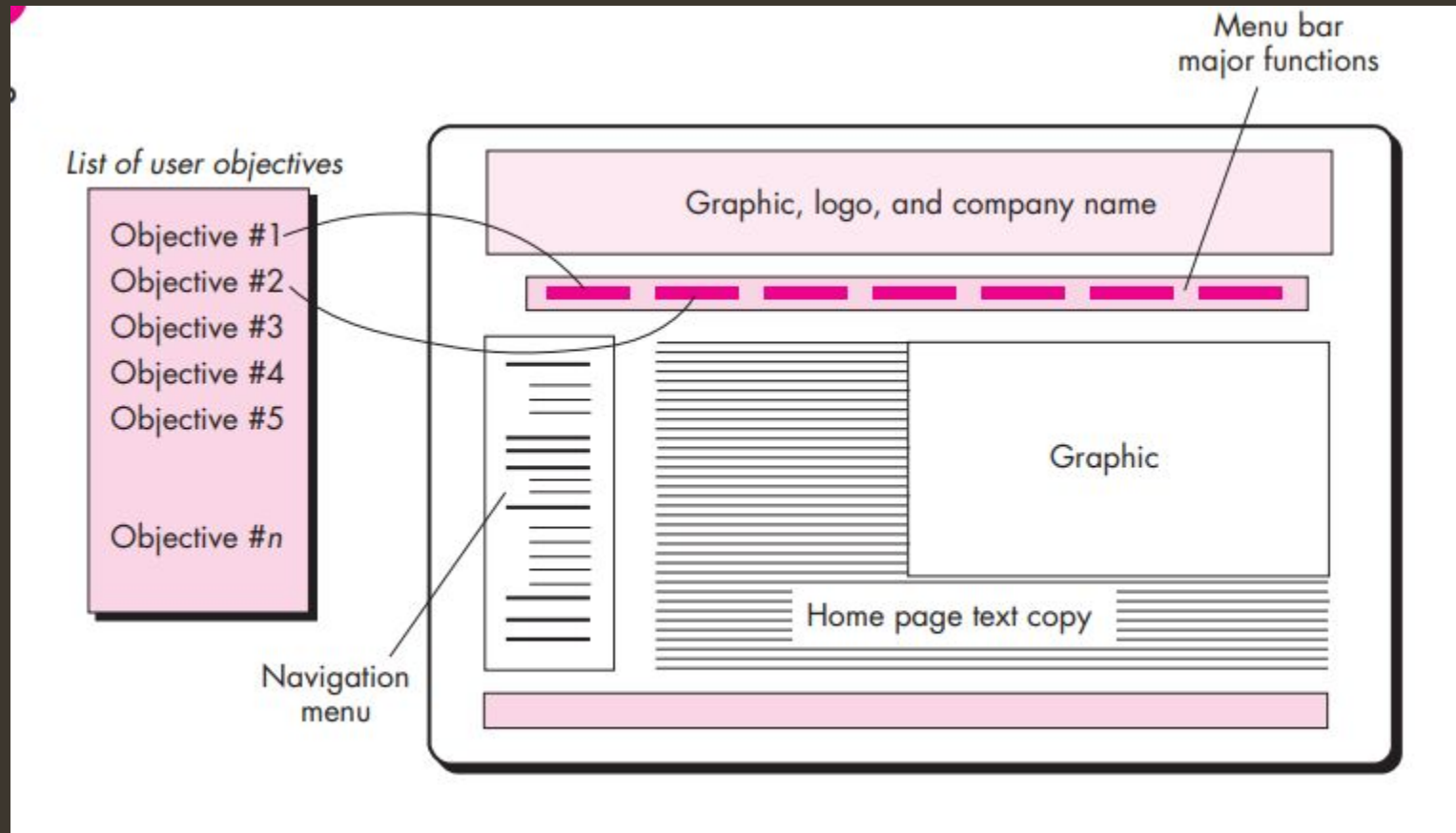


# Questions for Menu Labeling and Typed Commands

- Will every menu option have a corresponding command?
- What form will a command take? A control sequence? A function key? A typed word?
- How difficult will it be to learn and remember the commands?
- What can be done if a command is forgotten?
- Can commands be customized or abbreviated by the user?
- Are menu labels self-explanatory within the context of the interface?
- Are submenus consistent with the function implied by a master menu item?



# Example



# Interface construction and implementation:

- 1. Review information contained in the requirements model and refine as required.
- 2. Develop a rough sketch of the WebApp interface layout.
- Map user objectives into specific interface actions.
- Define a set of user tasks that are associated with each action.
- Storyboard screen images for each interface action.
- Refine interface layout and storyboards using input from aesthetic design
- identify user interface objects that are required to implement the interface.
- Describe the interface layout for each state.
- Refine and review the interface design model.

# Design Evaluation

