User Interface Design

Chapter 11

Typical Design Errors

- lack of consistency
- too much memorization
- no guidance / help
- no context sensitivity
- poor response
- Arcane/unfriendly

Golden Rules

- Place the user in control
- Reduce the user’s memory load
- Make the interface consistent

Place the user in control

- Define interaction modes in a way that does not force a user into unnecessary or undesired actions.
  - For example, if spell check is selected in a word- processor menu, the software moves to a spell-checking mode. Do not force to stay that mode.
- Provide for flexible interaction.
  - For example, software might allow a user to interact via keyboard commands, mouse movement, a digitizer pen, a multitouch screen, or voice recognition commands.
- Allow user interaction to be interruptible and undoable.
  - interrupt the sequence to do something else (without losing the work that had been done)

Place the user in control

- Streamline interaction as skill levels advance and allow the interaction to be customized.
  - Users often find that they perform the same sequence of interactions repeatedly. Allow advanced user to customize the interface.
- Hide technical internals from the casual user.
  - move the user into the virtual world of the application. The user should not be aware of the operating system, file management functions.
- Design for direct interaction with objects that appear on the screen.
  - The user feels a sense of control when able to manipulate the objects. Like resize the window.

Reduce the user's memory load

- Reduce demand on short-term memory.
  - providing visual cues that enable a user to recognize past actions, rather than having to recall them
- Establish meaningful defaults.
  - The initial set of defaults should make sense for the average user, but a user should be able to specify individual preferences. However, a "reset" option should be available.
- Define shortcuts that are intuitive.
  - When mnemonics are used to accomplish a system function (e.g., alt-P to invoke the print function)
- The visual layout of the interface should be based on a real world metaphor.
  - For example, a bill payment system should use a checkbook and check register metaphor to guide the user through the bill paying process. This enables the user to rely on well-understood visual cues, rather than memorizing an arcane interaction sequence.
- Disclose information in a progressive fashion.
  - functions under a text style menu in ms word.

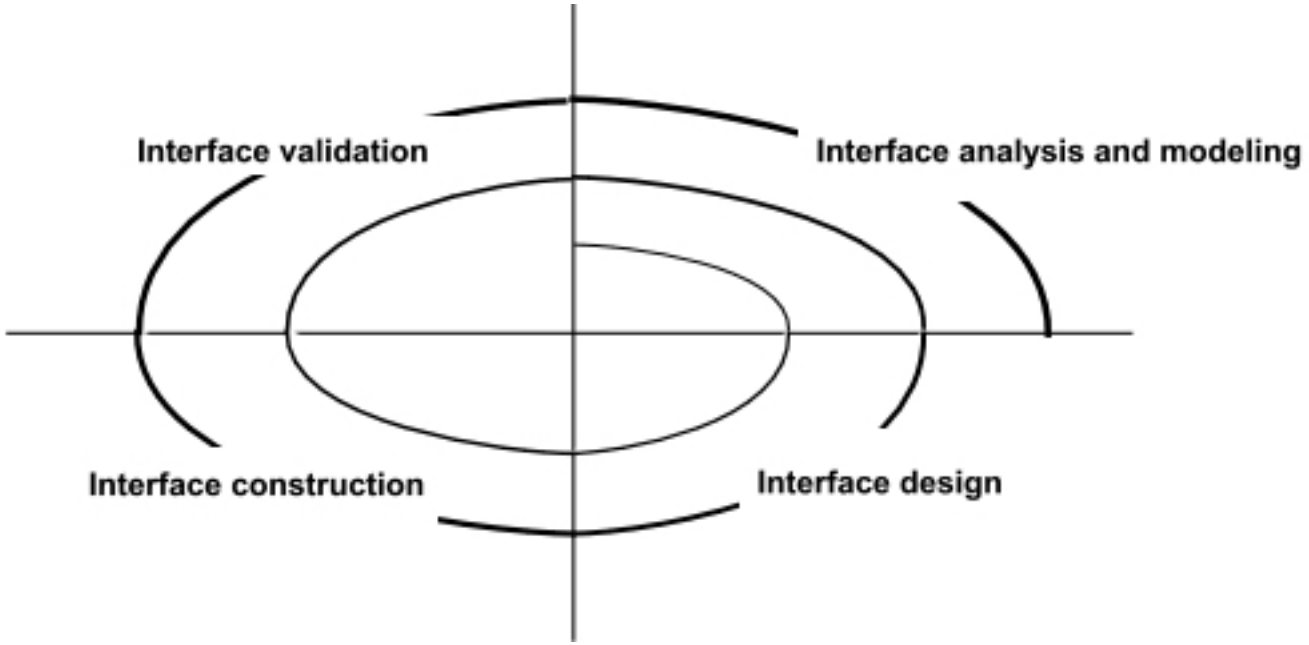
Make the interface consistent

- Allow the user to put the current task into a meaningful context.
  - It is important to provide indicators (e.g., window titles, graphical icons, consistent color coding) that enable the user to know the context of the work at hand.
  - what alternatives exist for a transition to a new task (sidebar/navigation bar)
- Maintain consistency across a family of applications.
  - Word, excel, power-point
- If past interactive models have created user expectations, do not make changes unless there is a compelling reason to do so.
  - ctlr+s to save

User Interface Design Models

- User model — a profile of all end users of the system
- Design model — a design realization of the user model
- Users' Mental model (system perception) — the user's mental image of what the interface is.
- Implementation model — the interface "look and feel" coupled with supporting information that describe interface syntax and semantics
  - syntactic knowledge refers to the mechanics of interaction that are required to use the interface effectively.
  - Semantic knowledge refers to the underlying sense of the application—an understanding of the functions that are performed, the meaning of input and output, and the goals and objectives of the system.

User Interface Design Process



Interface Analysis

- Interface analysis 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.

User Analysis

- 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?
- Are users expert typists or keyboard phobic?
- What is the age range of the user community?

Task Analysis and Modeling

- Use-cases define basic interaction
- Task elaboration refines interactive tasks
- Object elaboration identifies interface objects (classes)
- Workflow analysis defines how a work process is completed when several people (and roles) are involved
- Hierarchical representation

Analysis of Display Content

- Are different types of data assigned to consistent geographic locations on the screen (e.g., photos always appear in the upper right hand corner)?
- If a large report is to be presented, how should it be partitioned for ease of understanding? Will mechanisms be available for moving directly to summary information for large collections of data.
- Will graphical output be scaled to fit within the bounds of the display device that is used?
- How will color to be used to enhance understanding?
- How will error messages and warning be presented to the user?

Analysis of the Work Environment

- 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, a keyboard or mouse may not be an option, display placement may be less than ideal.
- The interface designer may be constrained by factors that mitigate against ease of use.

Interface Design Steps

- Using information developed during interface analysis, define interface objects and actions (operations).
- Define events (user actions) that will cause the state of the user interface to change. Model this behavior.
- Depict each interface state as it will actually look to the end-user.
- Indicate how the user interprets the state of the system from information provided through the interface.

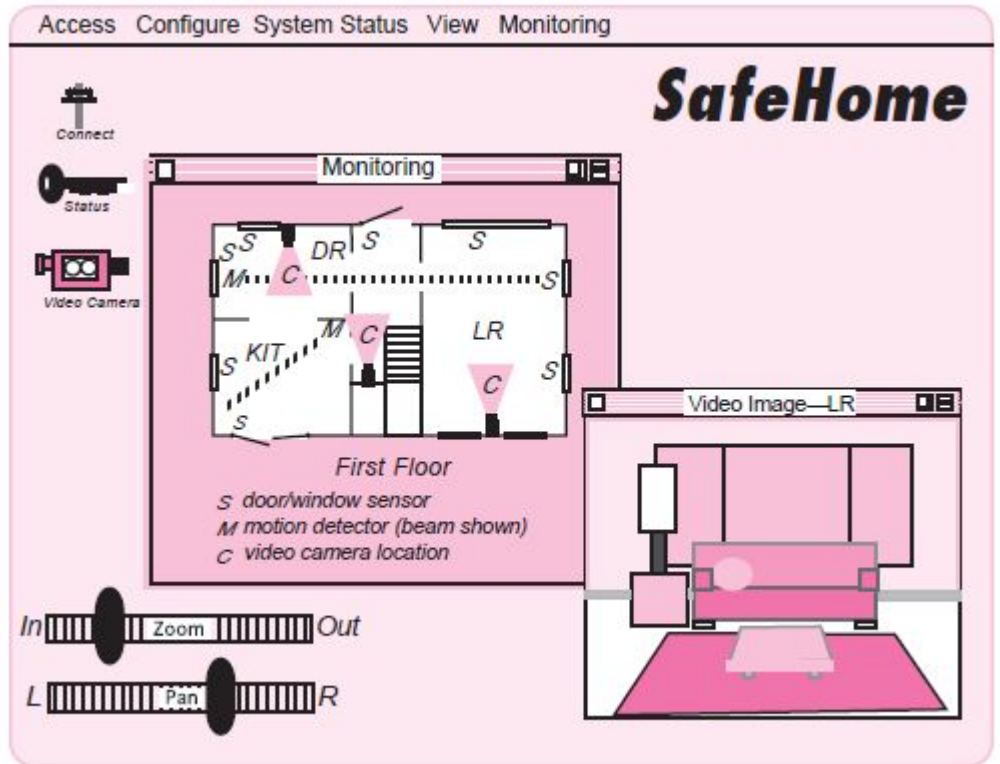
Interface Design Steps

- Preliminary use case:
  - I want to gain access to my SafeHome system from any remote location via the Internet. Using browser software operating on my notebook computer(while I’m at work or traveling),
  - I can determine the status of the alarm system,
  - arm or disarm the system,
  - reconfigure security zones, and view different rooms within the house via preinstalled video cameras.

Interface Design Steps

- Based on this use case, the following homeowner tasks, objects, and data items are identified:
  - *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**

Preliminary screen layout



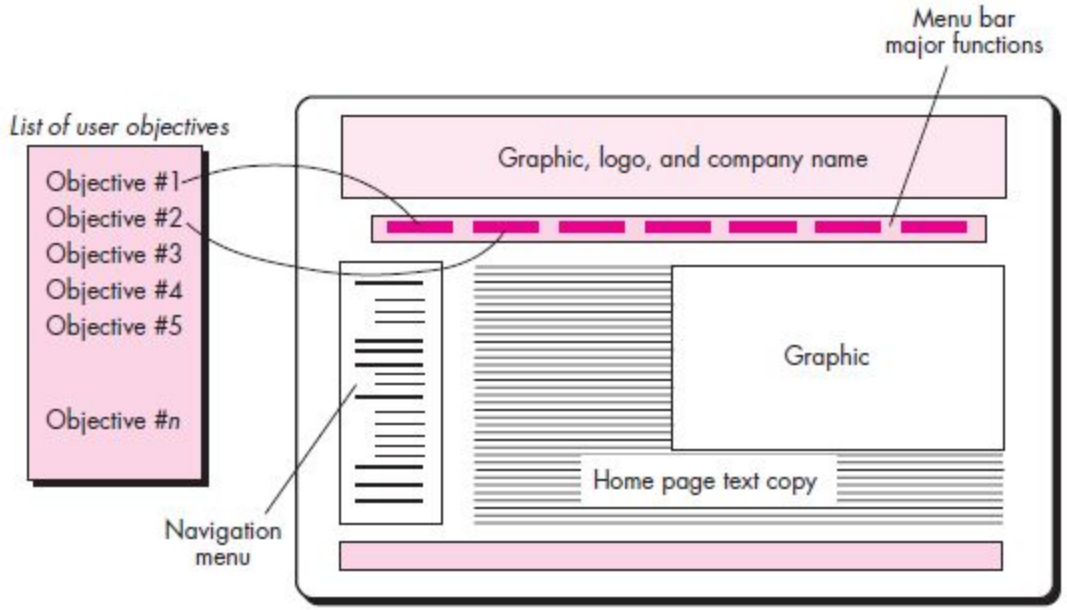
Design Issues

- Response time
- Help facilities
- Error handling
- Menu and command labeling
- Application accessibility
- Internationalization

Design Construction (for web app)

- *Where am I?* The interface should
  - provide an indication of the WebApp that has been accessed
  - inform the user of her location in the content hierarchy.
- *What can I do now?* The interface should always help the user understand his current options
  - what functions are available?
  - what links are live?
  - what content is relevant?
- *Where have I been, where am I going?* The interface must facilitate navigation.
  - Provide a “map” (implemented in a way that is easy to understand) of where the user has been and what paths may be taken to move elsewhere within the WebApp.

Design Construction



Interface Design Principles-I

- Anticipation—A WebApp should be designed so that it anticipates the use's next move.
- Communication—The interface should communicate the status of any activity initiated by the user
- Consistency—The use of navigation controls, menus, icons, and aesthetics (e.g., color, shape, layout)
- Controlled autonomy—The interface should facilitate user movement throughout the WebApp, but it should do so in a manner that enforces navigation conventions that have been established for the application.
- Efficiency—The design of the WebApp and its interface should optimize the user's work efficiency, not the efficiency of the Web engineer who designs and builds it or the client-server environment that executes it.

Design Evaluation