User Interface Design

3.4 User Interface Design: Rules, User Interface Analysis and
Steps in Interface Design, Design Evaluation

User Interface Design Rules
Golden Rules

Theo Mandel coined three *golden rules:*

1. Place the user in control.

2. Reduce the user's memory load.

3. Make the interface consistent.

The basis for a set of user interface design principles that guide this important aspect of software design.

User’s Expectations!!!!

- “What I really would like,” said the user solemnly, “is a system that reads my mind. It knows what I want to do before I need to do it and makes it very easy for me to get it done. That’s all, just that.”

User's Expectations!!!!

- "What I really would like," said the user solemnly, "is a system that reads my mind. It knows what I want to do before I need to do it and makes it very easy for me to get it done. That's all, just that."

- She wanted a system that :
  - ❑ Reacted to her needs
  - ❑ Help her get things done.
  - ❑ Control the computer, not have the computer control her.

Place User in control

Mandel defines a number of design principles that allow the user to maintain control:

- **Define interaction modes in a way that does not force a user into unnecessary or undesired actions.**
  - ❑ An interaction mode is the current state of the interface.
  - ❑ For example, if *spell check* is selected in a word-processor menu, the software moves to a spell-checking mode. The user should be able to enter and exit the mode with little or no effort.

Place User in control

- **Provide for flexible interaction.**
  - ❑ Different users have different interaction preferences.
  - ❑ For example, software might allow a user to interact via keyboard commands, mouse movement, a digitizer pen, a multi touch screen, or voice recognition commands**.**
- **Allow user interaction to be interruptible and undoable.**
  - ❑ Sequence of actions can be interrupted or undoable
- **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.

Place User in control

- **Hide technical internals from the casual user.**
  - The user should not be aware of the operating system, file management functions, or other computing technology.
- **Design for direct interaction with objects that appear on the screen.**
  - For example, an application interface that allows a user to “stretch” an object (scale it in size) is an implementation of direct manipulation.

Reduce the User's Memory Load

- The more a user has to remember,
  - the more error-prone the interaction with the system will be.
- Do not tax the user's memory.
- Whenever possible, the system should "remember" pertinent information and assist the user with an interaction scenario that assists recall.

Reduce the User's Memory Load

Mandel defines design principles that enable an interface to reduce the user's memory load:

- **Reduce demand on short-term memory**
  - No need to remember past actions, inputs, and results
  - Providing visual cues that enable a user to recognize past actions, rather than having to recall them
- **Establish meaningful defaults.**
  - A "reset" option should be available, enabling the redefinition of original default values.
- **Define shortcuts that are intuitive.**
  - When mnemonics are used to accomplish a system function.
  - e.g., alt-P to invoke the print function, the mnemonic should be tied to the action in a way that is easy to remember

Reduce the User's Memory Load

- **The visual layout of the interface should be based on a real-world metaphor.**
  - For example, a bill payment system should use a check book and check register metaphor to guide the user through the bill paying process.
- **Disclose information in a progressive fashion.**
  - An example, common to many word-processing applications, is the underlining function, is **A TEXT STYLE MENU**. However, every underlining capability is not listed. The user must pick underlining; then all underlining options (e.g., single underline, double underline, dashed underline) are presented.

Make the Interface Consistent

The interface should present and acquire information in a consistent fashion. This implies that:
(1) all visual information is organized according to design rules that are maintained throughout all screen displays,
(2) input mechanisms are constrained to a limited set that is used consistently throughout the application, and
(3) mechanisms for navigating from task to task are consistently defined and implemented.

Make the Interface Consistent

- **Allow the user to put the current task into a meaningful context.**
  - ❑ Provide indicators (e.g., window titles, graphical icons, consistent color coding) that enable the user to know the context of the work at hand.
  - ❑ Offers theme that can use to cater to specific accessibility needs, such as high contrast themes for users with visual impairments.
  - ❑ Accessibile Themes->Accessibility
- **Maintain consistency across a family of applications.**
  - ❑ Implement the same design rules so that consistency is maintained for all interaction
- **If past interactive models have created user expectations, do not make changes unless there is a compelling reason to do so.**
  - ❑ e.g., the use of alt-S to save a file), the user expects this in every application he encounters. A change (e.g., using alt-S to invoke scaling) will cause confusion.

USER INTERFACE ANALYSIS AND DESIGN

Interface Analysis and Design Models

Four different models come into play when a user interface is to be analyzed and designed:

- A human engineer establishes a user model.
- The software engineer creates a design model.
- The end user develops a mental image that is often called the user's mental model or the system perception,
- The implementers of the system create an implementation model.

Interface Analysis and Design Models

- Unfortunately, each of these models may differ significantly.

- Your role, as an interface designer, is:
  - to reconcile these differences and derive a consistent representation of the interface.

Users Can Be Categorized As:

- *Novices*
  - No syntactic knowledge of the system and little semantic knowledge of the application or computer usage in general.
- *Knowledgeable, intermittent users*
  - Reasonable semantic knowledge of the application but relatively low recall of syntactic information necessary to use the interface.
- *Knowledgeable, frequent users*
  - Good semantic and syntactic knowledge that often leads to the "power-user syndrome"; that is, individuals who look for shortcuts and abbreviated modes of interaction.

User's *Mental Model*

- The (system perception) is the image of the system that end users carry in their heads.

- For example, if the user of a particular word processor were asked to describe its operation, the system perception would guide the response.

Implementation Model

The *implementation model* combines :

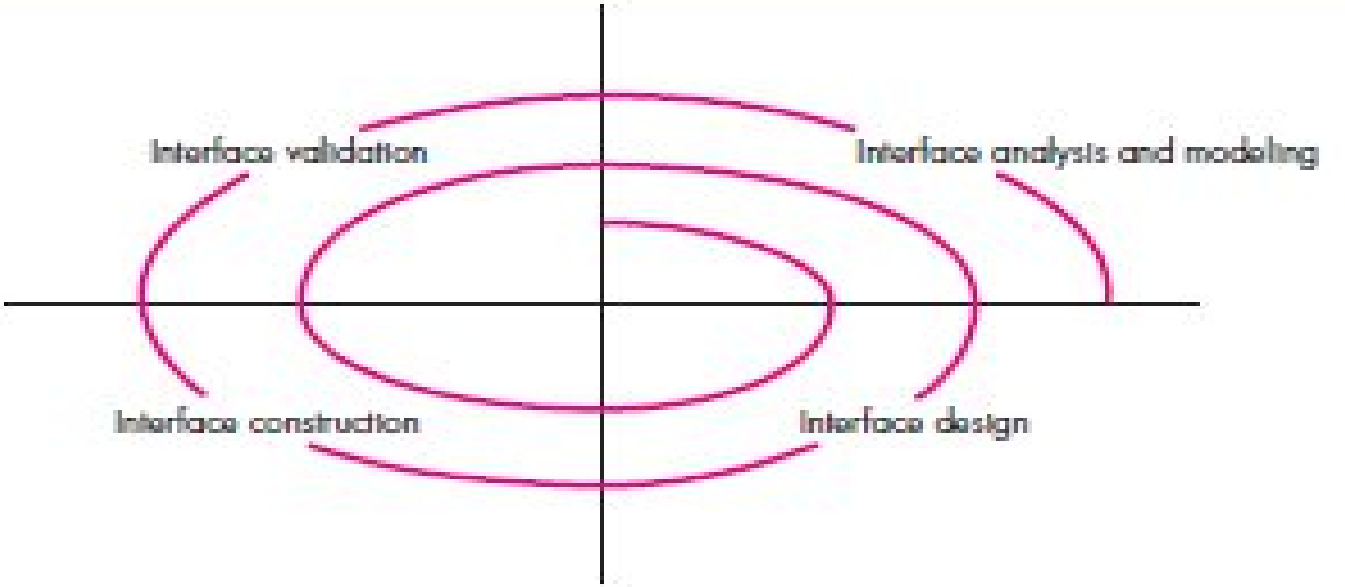
- the outward manifestation of the computer based system :
  - the look and feel of the interface
- coupled with all supporting information:
  - Books
  - Manuals
  - videotapes
  - help files

  that describes interface syntax and semantics.

- When the implementation model and the user's mental model are coincident, users generally feel comfortable with the software and use it effectively.

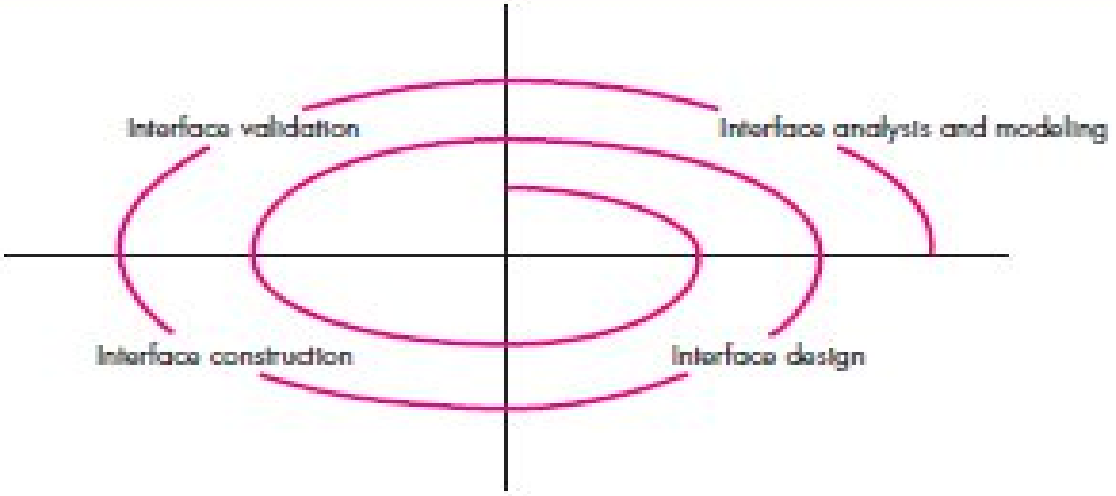
Analysis And Design Process

- The analysis and design process for user interfaces is iterative and can be represented using a spiral model.



Analysis And Design Process

The user interface analysis and design process begins at the interior of the spiral and encompasses four distinct framework activities:
(1) Interface Analysis And Modeling,
(2) Interface Design,
(3) Interface Construction,
(4) Interface Validation



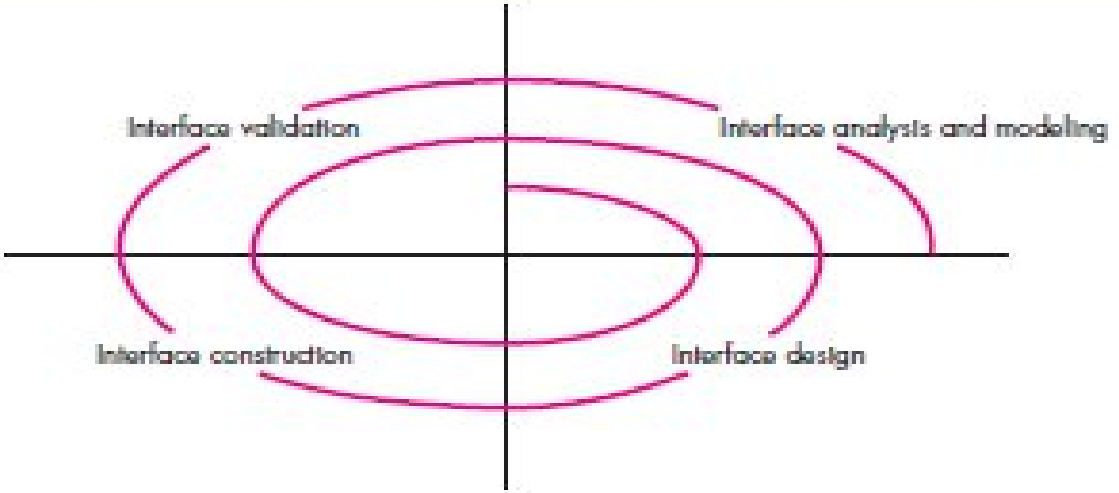
Analysis And Design Process

- The spiral implies that:
  - each of these tasks will occur more than once
  - with each pass around the spiral representing additional elaboration of requirements and the resultant design.

(1) Interface Analysis And Modeling,

- *Interface analysis* focuses on the profile of the users who will interact with the system.

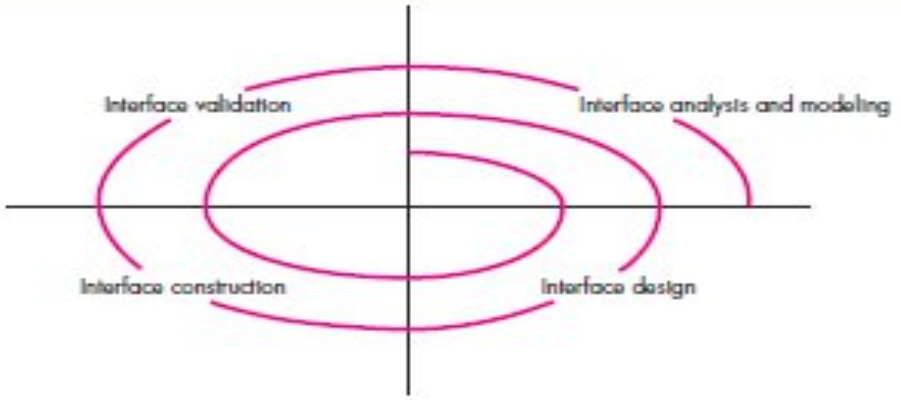
- Different user categories are defined.
  - For each user category, requirements are elicited.
  - Understand the system perception for each class of users.



(1) Interface Analysis And Modeling,

- Once general requirements have been defined, a more detailed *task analysis* is conducted.

- Those tasks that the user performs to accomplish the goals of the system are :
  - Identified
  - described,
  - and elaborated (over a number of iterative passes through the spiral).



(1) Interface Analysis And Modeling,

- Finally, analysis of the user environment focuses on the physical work environment.
- Among the questions to be asked are
  - Where will the interface be located physically?(ATM/Kiosk/PCs)
  - Will the user be sitting, standing, or performing other tasks unrelated to the interface?
  - Does the interface hardware accommodate space, light, or noise constraints?
  - Are there special human factors considerations driven by environmental factors?

- The information gathered as part of the analysis action is used to create an analysis model for the interface.
  - Using this model as a basis, the design action commences.

(2) Interface Design

- The goal of *interface design* is to:
  - Design a set of interface objects and actions (and their screen representations)
  - that enable a user to perform all defined tasks in a manner that meets every usability goal defined for the system.

(3) Interface Construction

The construction activity involves prototyping—the only practical way to validate what has been designed.

Interface construction normally begins with:

- the creation of a prototype that enables usage scenarios to be evaluated.

As the iterative design process continues,

- a user interface tool kit may be used to complete the construction of the interface.

(4) Interface Validation

Interface validation focuses on:

(1) the ability of the interface to implement every user task:

- correctly
- to accommodate all task variations,
- to achieve all general user requirements;

(2) the degree to which the interface is easy to use and easy to learn

(3) the users' acceptance of the interface as a useful tool in their work

Interface analysis

- User Analysis
- Task Analysis

User Analysis

- ??

User Analysis

- **User Interviews**
  - members of the software team meet with end users to better understand their needs
- **Sales input**
  - Sales people meet with users on a regular basis and can gather information that will help the software team to categorize users
- **Marketing input**
  - Market analysis can be invaluable in the definition of market segments and an understanding of how each segment might use the software
- **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

Questions:: Understand The Users

- 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?
- Will the users be represented predominately by one gender?
- How are users compensated for the work they perform?

Questions:: Understand The Users

- 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?

Task Analysis and Modeling

Task Analysis and Modeling

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?

Techniques used To answer these questions

- **Use cases.**
- **Task elaboration.**
- **Object elaboration**
- **Workflow analysis**
- **Hierarchical representation**

Task elaboration

- Regardless of the overall approach to task analysis, you must first define and classify tasks.
- One approach is stepwise elaboration
- For example, let's reconsider the computer-aided design system for interior designers. By observing an interior designer at work, you notice that interior design comprises a number of major activities:
  - furniture layout
  - fabric and material selection
  - wall and window coverings selection,
  - presentation (to the customer)
  - costing, and shopping.
- Each of these major tasks can be elaborated into subtasks.

Task elaboration

For example, 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, and
(7) draw a perspective-rendering view for the customer.

Task elaboration

- Stepwise elaboration
  - also called functional decomposition or stepwise refinement
- as a mechanism for refining the processing tasks that are required for software.

Object elaboration

- Rather than focusing on the tasks that a user must perform, you can examine the use case and other information obtained from the user and
  - extract the physical objects that are used by the interior designer.
  - Objects can be categorized into classes.
  - Attributes of each class are defined, and
  - An evaluation of the actions applied to each object provide a list of operations.

Object elaboration

- 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. The tasks *select*, *move*, and *draw* are operations.

Workflow analysis

- This technique allows you to understand how a work process is completed when several people (and roles) are involved.

- Consider a company that intends to fully automate the process of prescribing and delivering prescription drugs.
  - The entire process will revolve around a Web-based application that is accessible by physicians (or their assistants), pharmacists, and patients.
- Workflow can be represented effectively with a UML swimlane or activity diagram.

Hierarchical representation

- Once workflow has been established, a task hierarchy can be defined for each user type.
- The hierarchy is derived by a stepwise elaboration of each task identified for the user.

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.*

Analysis of Display Content

- For modern applications, display content can range from:
  - character-based reports
    - a spreadsheet
  - graphical displays
    - histogram
    - 3-D model
    - picture of a person
  - specialized information
    - audio or video
    - files

Questions for 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)?

- Can the user customize the screen location for content?

- Is proper on-screen identification assigned to all content?

- If a large report is to be presented, how should it be partitioned for ease of understanding?

Questions for Analysis of Display Content

- 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 be used to enhance understanding?

- How will error messages and warnings be presented to the user?

Analysis of the Work Environment

Hackos and Redish [Hac98] discuss the importance of work environment analysis when they state:

- People do not perform their work in isolation. They are influenced by
  - the activity around them,
  - the physical characteristics of the workplace,
  - the type of equipment they are using, and
  - the work relationships they have with other people.
- If the products you design do not fit into the environment, they may be difficult or frustrating to use.

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.