CHAPTER 12 USER INTERFACE DESIGN

Overview

This chapter introduces the principles of user interface design as it relates to the development of software products. Proper interface design begins with careful analysis of the user, the task and the environment. Once the user's tasks are identified, user scenarios are created and validated. Good user interfaces are designed, they don't happen by chance. Prototyping is a common approach to user interface analysis and design. Early involvement of the user in the design process makes her or him more likely to accept the final product. User interfaces must be field tested and validated prior to general release.

Place User in Control

- Define interaction in such a way that the user is not forced into performing unnecessary or undesired actions
- Provide for flexible interaction (users have varying preferences)
- Allow user interaction to be interruptible and reversible
- Streamline interaction as skill level increases and allow customization of interaction
- Hide technical internals from the casual user
- Design for direct interaction with objects that appear on the screen

Reduce User Memory Load

- Reduce demands on user's short-term memory
- Establish meaningful defaults
- Define intuitive short-cuts
- Visual layout of user interface should be based on a familiar real world metaphor
- Disclose information in a progressive fashion

Make Interface Consistent

- Allow user to put the current task into a meaningful context
- Maintain consistency across a family of applications
- If past interaction models have created user expectations, do not make changes unless there is a good reason to do so

Usability Assessment by Observing Users

- Is system usable without continual help of instruction?
- Do rules of interaction help a knowledgeable user work efficiently?
- Do interaction rules become more flexible as users become more knowledgeable?
- Has system been tuned to the physical and social user environment?
- Is the user aware of the state of the system?
- Does the user know where he or she is at all times?
• Is the interface structured in a logical and consistent manner?
• Are interaction mechanisms, icons, and procedures consistent across the interface?
• Does the interaction anticipate errors and help users correct them?
• Is the interface tolerant of errors made?
• Is the interaction simple?

User Interface Design Models

• Design model (incorporates data, architectural, interface, and procedural representations of the software)
• User model (end user profiles - novice, knowledgeable intermittent user, knowledgeable frequent users)
• User's model or system perception (user's mental image of system)
• Implementation model (look and feel of the interface and supporting media)

User Categories

• Novices - have no syntactic knowledge of the system and little knowledge of the application or general computer usage
• Knowledgeable intermittent users - have reasonable semantic knowledge of the application, but low recall of syntactic information necessary to use the interface
• Knowledgeable frequent users - good semantic and syntactic knowledge of the application, look for interaction shortcuts

User Interface Design Process (Spiral Model)

• User, task, and environment analysis and modeling
• Interface design
• Interface construction
• Interface validation

Analysis of Physical Work Environment

• Where will the interface be located physically?
• 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?

Validation Considerations

• Does the interface implement every user task correctly, accommodate all task variations, and achieve all general user requirements?
• To what degree is the user interface easy to use and easy to learn?
• Do users' accept that the user interface as a useful tool in their daily work?
Interface Analysis Goals

- Understand people who interact with system through the interface
- Understand tasks end-users must perform to do their work
- Understand the content presented as part of the interface
- Understand the nature of the environment in which tasks will be performed

User Analysis Sources

- User interviews - designers meet with end-users individually or in groups
- Sales input - sales people help designers categorize users and better understand their needs
- Marketing input - marketing analysis can help define market segments and help understand how each segment might use the software
- Support input - support staff can provide good input what works and does not, what users like, what features generate questions, and what features are easy to use

Task Analysis and Modeling

- Use-cases
  - What work will the user perform in specific circumstances?
- Task elaboration
  - What tasks and subtasks will be performed as the user does the work?
- Object elaboration
  - What specific problem domain objects will the user manipulate as work is performed?
- Workflow analysis
  - What is the sequence of work tasks?
- Hierarchical representation
  - What is the hierarchy of tasks?

Interface Layout and Content Review Questions

- Are different types of data assigned to consistent geographical locations on the screen?
- Can user customize screen locations of content?
- Is proper on-screen identification assigned to all content?
- How should large report be partitioned for ease of understanding?
- Will mechanisms be available for moving directly to data summary information for large data collections?
- Will graphical output be scaled to fit bounds of display device used?
- How will color be used to enhance understanding?
- How will error messages and warnings be displayed to the user?

Interface Design
• Use the information developed during interface analysis to define interface objects and actions (operations)
• Define events (user actions) that will cause the state of the user interface to change and model this behavior
• Depict each interface state as it will appear to the end-user
• Indicate how the user interprets the state of the system from information provided through the interface

User Interface Design Pattern Examples

• Top-level navigation
• Card stack
• Fill-in-the-blanks
• Sortable table
• Bread crumbs
• Edit-in-place
• Simple search
• Shopping cart
• Progress indicator

Interface Design Issues

• System response time - time between the point at which user initiates some control action and the time when the system responds
• User help facilities - integrated, context sensitive help versus add-on help
• Error handling - messages should be non-judgmental, describe problem precisely, and suggest valid solutions
• Menu and command labeling - based on user vocabulary, simple grammar, and have consistent rules for abbreviation
• Application accessibility - ensuring that interface design contains mechanisms for accommodating user with special needs (vision, hearing, mobility, speech, learning impairments)
• Internationalization - interface accommodates needs of different locales and languages

User-Interface Development System Capabilities

• Managing input devices
• Validating user input
• Error handling and displaying error messages
• Providing feedback
• Proving help and prompts
• Handling windows and fields
• Establishing connections between application software and interface
• Insulating application from interface management functions
• Allowing user to customize the interface
User Interface Evaluation Cycle

1. Preliminary design
2. Build first interface prototype
3. User evaluates interface
4. Evaluation studied by designer
5. Design modifications made
6. Build next prototype
7. If interface is not complete then go to step 3

User Interface Design Evaluation Criteria

- Length and complexity of written interface specification provide an indication of amount of learning required by system users
- Number of user tasks and the average number of actions per task provide an indication of interaction time and overall system efficiency
- Number of tasks, actions, and system states in the design model provide an indication of the memory load required of system users
- Interface style, help facilities, and error handling protocols provide a general indication of system complexity and the degree of acceptance by the users