

## Lecture 21: User Interface Design

Kenneth M. Anderson  
Object-Oriented Analysis and Design  
CSCI 6448 - Spring Semester, 2002

## Credit where Credit is Due

- Some material presented in this lecture is taken from section 7 of Maciaszek's "Requirements Analysis and System Design". © Addison Wesley, 2000

## Goals for this Lecture

- Cover the information presented in Section 7 of the textbook
  - Discuss
    - User Interface Design
      - Background Information
      - UI Guidelines
      - OO Models for UI Design
        - » Window Navigation

## Last Lecture

- Covered
  - Design Tutorial
    - Fred Brooks "No Silver Bullet"
      - Summary: Design is Hard! :-)
    - Architectural Design
    - Detailed Design

## Quiz

1. Draw a sequence diagram for the following scenario  
A registrar object creates a student object and adds the student to its school. The registrar object instructs the student to register for classes, so the student creates an empty schedule object and adds three courses to its schedule. The registrar object then sets the student's status to "registered"
2. Create the equivalent collaboration diagram

## User Interface Design

- Modern software development often requires significant user interface design activities
  - Research in UI design has been conducted by the Computer-Human Interaction (CHI or HCI) community since the early 80's
    - Previously handled by psychology, social science, and ergonomics
- UI design starts with sketches in analysis
  - which are used for requirements gathering, storyboarding sessions, prototype development, and for inclusion in use case documents
- During design, sketches are refined and connected to boundary classes that will implement them

## UI Design is Multidisciplinary

- UI design (done right) requires a team of skilled professionals
  - including graphic artists, HCI experts, usability experts, developers, domain experts and users (!!)
- Initial designs flow from the tasks that the system must support; use cases are useful in identifying a system's tasks
  - Note: It is impossible to cover the topic of UI Design in one lecture; Tammy Sumner takes an entire semester to cover it in CSCI 6838!!

## Maciaszek's process

- An analyst develops sketches during requirements analysis; sketches are placed with use cases
- A designer specifies which collaborations will be involved with handling the tasks of each use case
  - sketches are updated to match the particular GUI technology selected for system implementation (see page 246)
  - A technology expert may be consulted to ensure proper use of the GUI toolkit's elements
- Prototypes are developed by graphic artists and "social and behavioral scientists"
  - This is outdated terminology, UI designer, HCI expert, usability expert more likely used today
  - Prototypes are tested on users and the design is evolved

## Maciaszek's process, cont.

- A programmer implements the final interfaces developed by the prototype sessions
  - This involves integrating boundary and control classes with the relevant entity classes
- More information:  
<<http://www.acm.org/sigchi/>>

## UI Guidelines

- GUI design centers around users
  - lots of guidelines have been developed to help design user interfaces that build on the importance of the user
- 1. User in Control
  - Modern systems make users feel in control
    - “no mothering” principle
    - users initiate actions; receives feedback; can cancel; can undo (or be warned when undo is not possible)

## UI Guidelines, continued

- 2. Consistency
  - UI adheres to platform's GUI standards
    - adherence to standards allows users to transfer knowledge and interaction skills from one program to another
      - Xerox Star, Apple Macintosh are early examples
- 3. Personalization and Customization
  - UI should allow user to “tailor” interface to their desires (resize columns, for example)
  - UI should also allow more extensive customization to handle different types of users (novice/expert or co-branding)

## UI Guidelines

- 4. Forgiveness
  - A good interface allows user to experiment, which requires the ability to “roll back” or “undo” previous actions
    - this is really just part of “user in control”
  - Forgiveness is hard to implement
    - lots of work on developer to make things easy on user
- 5. Feedback
  - Another part of “user in control”
  - System must provide user feedback on all actions (and allow them to cancel actions that are taking too long)

## UI Guidelines

- 6. Aesthetics and Usability
  - Aesthetics concern a system's visual appeal
  - Usability refers to a system's "ease of use"
    - Both can be achieved only via iteration
    - Usability tests involve users in a "controlled environment" often with a "one-way mirror" allowing developers to watch users struggle with their interfaces!

## UI Widgets

- Maciaszek surveys different types of windows, widgets, and interaction styles including
  - primary and secondary windows, row browsers, tree browsers, web pages, dialog boxes, tab folders, drop-down lists, message boxes, single document and multiple document interfaces
- Since most of this content is MS Windows specific, I do not require any of this information in this class

## Window Navigation

- Once a UI has been designed, activity diagrams can be used to model the navigation between UI elements (mostly windows) of a system
  - This can be useful information for a developer who has to implement boundary and control classes
- States in an activity diagram can be stereotyped with UI elements that allow them to be adapted to modeling window navigation
  - See figure 7.21 on page 265 and the taxonomy of elements on pages 266 and 267

## Using Activity Diagrams

- For each UI element, activities that can be initiated are represented as sub-states
  - See figure 7.22 on pages 266
- Once states and activities have been specified, transition lines can be used to indicate navigation among UI elements
  - See figures 7.23, 7.24, and 7.25 on pages 268-271