Learning Objectives

- Explain the process of designing interfaces and dialogues and the deliverables for their creation.
- Contrast and apply several methods for interacting with a system.
- List and describe various input devices and discuss usability issues for each in relation to performing different tasks.
- Describe and apply the general guidelines for designing interfaces and specific guidelines for layout design, structuring data entry fields, providing feedback, and system help.

Designing Interfaces/Dialogues

- User-focused activity
- Prototyping methodology of iteratively:
  - Collecting information
  - Constructing a prototype
  - Assessing usability
  - Making refinements
- Must answer the who, what, where, and how questions
Designing Interfaces and Dialogues (Cont.)

![Figure 11-1: Systems development life cycle (SDLC)]

Deliverables and Outcomes

- Creation of a design specification
  - A typical interface/dialogue design specification is similar to form design, but includes multiple forms and dialogue sequence specifications.
  - Our exercise is based on RRC’s UI Sketch, Storyboard, and UI Flow.

Deliverables and Outcomes (Cont.)

- The specification includes:
  - Narrative overview
  - Sample design
  - Testing and usability assessment
  - Dialogue sequence
  - **Dialogue sequence** is the ways a user can move from one display to another.

Interaction Methods and Devices

- **Interface**: a method by which users interact with an information system
- All human-computer interfaces must:
  - have an interaction style, and
  - use some hardware device(s) for supporting this interaction.
  - use a good combination of ergonomics (efficiency in working environment), aesthetics (pleasing appearance), and interface technology.
Methods of Interacting

- Command line
  - Includes keyboard shortcuts and function keys
- Menu
- Form
- Object-based
- Natural language

Command Language Interaction

- **Command language interaction**: a human-computer interaction method whereby users enter explicit statements into a system to invoke operations.
- Example from MS DOS:
  - `COPY C:PAPER.DOC A:PAPER.DOC`
  - Command copies a file from C: drive to A: drive

Menu Interaction

- **Menu interaction**: a human-computer interaction method in which a list of system options is provided and a specific command is invoked by user selection of a menu option.
- **Pop-up menu**: a menu-positioning method that places a menu near the current cursor position.

Menu Interaction (Cont.)

- **Drop-down menu** is a menu-positioning method that places the access point of the menu near the top line of the display.
  - When accessed, menus open by dropping down onto the display.
  - Visual editing tools help designers construct menus.
Menu Interaction (Cont.)

Guidelines for Menu Design
- **Wording** — meaningful titles, clear command verbs, mixed upper/lower case
- **Organization** — consistent organizing principle
- **Length** — all choices fit within screen length
- **Selection** — consistent, clear and easy selection methods
- **Highlighting** — only for selected options or unavailable options

FIGURE 11-8
Menu building with Microsoft Visual Basic .NET

Form Interaction

**Form interaction**: a highly intuitive human-computer interaction method whereby data fields are formatted in a manner similar to paper-based forms
- Allows users to fill in the blanks when working with a system.

FIGURE 11-9
Example of form interaction from the Google Advanced Search Engine *(Source: Google.)*
Object-Based Interaction

- **Object-based interaction**: a human-computer interaction method in which symbols are used to represent commands or functions
- **Icons**: graphical pictures that represent specific functions within a system
  - Use little screen space and are easily understood by users

Natural Language Interaction

- **Natural language interaction**: a human-computer interaction method whereby inputs to and outputs from a computer-based application are in a conventional spoken language such as English
- Based on research in artificial intelligence.
- Current implementations are tedious and difficult to work with, not as viable as other interaction methods.

Principle: User-Centered Design

- Understand the Business
- Maximize Graphical Effectiveness
- Think Like a User
- Use Models and Prototypes
- Focus on Usability
- Invite Feedback
- Document Everything
Designing the User Interface

- Design a *Transparent* (easy to perceive) Interface
  - Facilitate the system design objectives, rather than calling attention to the interface
  - Create a design that is easy to learn and remember
  - Design the interface to improve user efficiency and productivity
  - Write commands, actions, and system responses that are consistent and predictable

Designing the User Interface

- Create an Interface that Is *Easy to Learn and Use*
  - Clearly label all controls, buttons, and icons
  - Select only those images that users can understand easily, and provide on-screen instructions that are logical, concise, and clear
  - Show all commands in a list of menu items
  - Make it easy to navigate

Designing the User Interface

- *Enhance User Productivity*
  - Organize tasks, commands, and functions in groups that resemble actual business operations
  - Create alphabetical menu lists
  - Provide shortcuts so experienced users can avoid multiple menu levels
  - Use default values if the majority of values in a field are the same

Designing the User Interface

- Make It *Easy for Users to Obtain Help or Correct Errors*
  - Ensure that Help is always available
  - Provide user-selected Help and context-sensitive Help
Designing the User Interface

- **Minimize Input Data Problems**
  - Create input masks
  - Display event-driven messages and reminders
  - Establish a list of predefined values that users can click to select
  - Build in rules that enforce data integrity

Designing the User Interface

- Provide **Feedback to Users**
  - Display messages at a logical place on the screen
  - Alert users to lengthy processing times or delays
  - Allow messages to remain on the screen long enough for users to read them
  - Let the user know whether the task or operation was successful or not

Designing the User Interface

- Create an **Attractive Layout and Design**
  - Use appropriate colors to highlight different areas of the screen
  - Use special effects sparingly
  - Use hyperlinks that allow users to jump to related topics
  - Group related objects and information

Designing the User Interface

- Use **Familiar Terms and Images**
  - Remember that users are accustomed to a pattern of red=stop, yellow=caution, and green=go
  - Provide a keystroke alternative for each menu command
  - Use familiar commands if possible
  - Provide a Windows look and feel in your interface design if users are familiar with Windows-based applications
Designing the User Interface

- Add Control Features
  - Menu bar
  - Toolbar
  - Command button
  - Dialog box
  - Text box
  - Toggle button

- Add Control Features
  - List box – scroll bar
  - Drop-down list box
  - Option button, or radio button
  - Check box
  - Calendar control
  - Switchboard

Designing Form Interfaces

- Forms have several general areas in common:
  - Header information
  - Sequence and time-related information
  - Instruction or formatting information
  - Body or data details
  - Totals or data summary
  - Authorization or signatures
  - Comments

Form Layout Design

- Building Forms
  - Form layout
  - Heading zone
  - Control zone
  - Instruction zone
  - Body zone
  - Totals zone
  - Authorization zone
Structuring Data Entry

<table>
<thead>
<tr>
<th>Entry</th>
<th>Never require data that are already online or that can be computed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defaults</td>
<td>Always provide default values when appropriate</td>
</tr>
<tr>
<td>Units</td>
<td>Make clear the type of data units requested for entry</td>
</tr>
<tr>
<td>Replacement</td>
<td>Use character replacement when appropriate</td>
</tr>
<tr>
<td>Captioning</td>
<td>Always place a caption adjacent to fields</td>
</tr>
<tr>
<td>Format</td>
<td>Provide formatting examples</td>
</tr>
<tr>
<td>Justify</td>
<td>Automatically justify data entries</td>
</tr>
<tr>
<td>Help</td>
<td>Provide context-sensitive help when appropriate</td>
</tr>
</tbody>
</table>

Controlling Data Input

- **Objective**: Reduce data entry errors
- **Common sources data entry errors in a field**:
  -Appending: adding additional characters
  -Truncating: losing characters
  -Transcripting: entering invalid data
  -Transposing: reversing sequence of characters

Providing Feedback

- **Three types of system feedback**:
  - **Status information**: keep user informed of what’s going on, helpful when user has to wait for response
  - **Prompting cues**: tell user when input is needed, and how to provide the input
  - **Error or warning messages**: inform user that something is wrong, either with data entry or system operation

Designing Dialogues

- **Dialogues**: the sequence of interaction between a user and a system
- **Dialogue design involves**:
  - Designing a dialogue sequence.
  - Building a prototype.
  - Assessing usability.
Designing the Dialogue Sequence

- Typical dialogue between user and Customer Information System:
  - Request to view individual customer information.
  - Specify the customer of interest.
  - Select the year-to-date transaction summary display.
  - Review the customer information.
  - Leave system.

Designing the Dialogue Sequence (Cont.)

- Dialogue diagramming: a formal method for designing and representing human-computer dialogues using box and line diagrams

Designing the Dialogue Sequence (Cont.)

- Three sections of the box are used as:
  - Top contains a unique display reference number used by other displays for referencing it.
  - Middle contains the name or description of the display.
  - Bottom contains display reference numbers that can be accessed from the current display.

FIGURE 11-17
Sections of a dialogue diagramming box
Designing the Dialogue Sequence (Cont.)

- Dialogue diagrams depict the sequence, conditional branching, and repetition of dialogues.

Output Design

- Before designing output, ask yourself several questions:
  - What is the purpose of the output?
  - Who wants the information, why is it needed, and how will it be used?
  - What specific information will be included?
  - Will the output be printed, viewed on-screen, or both? What type of device will the output go to?

- Before designing output, ask yourself several questions:
  - When will the information be provided, and how often must it be updated?
  - Do security or confidentiality issues exist?
- Your answers will affect your output design strategies
Output Design (details in next lecture)

- Overview of Report Design
  - Few firms have been able to eliminate printed output totally
  - Turnaround documents
  - Reports must be easy to read and well organized
  - Database programs include a variety of report design tools
  - Character-based reports

Summary

- In this chapter you learned how to:
  - Explain the process of designing interfaces and dialogues and the deliverables for their creation.
  - Contrast and apply several methods for interacting with a system.
  - List and describe various input devices and discuss usability issues for each in relation to performing different tasks.
  - Describe and apply the general guidelines for designing interfaces and specific guidelines for layout design, structuring data entry fields, providing feedback, and system help.

Summary (Cont.)

- Design human-computer dialogues and understand how dialogue diagramming can be used to design dialogues.
- Design graphical user interfaces.
- Discuss guidelines for the design of interfaces and dialogues for Internet-based electronic commerce systems.