Topic 5d: Designing Interfaces and Dialogues – Part 2

Lecture Objectives

At the completion of this lecture you should be able to understand:

- the importance of using mental models
- how metaphors can prove useful
- different graphical interface controls available and their effective use
- the process of defining the interface dialogue

Cooper’s Manifest Model

**manifest models**

- models reflect technology
- are chosen by the developer.
- The best ones reflect the user’s mental model not the implementation model

**mental models**

- reflect user’s vision

Mental Models

- humans simplify (the music doesn’t come straight from the tape)
- humans anthropomorphise (inanimate objects gain personality)

The Value of the Manifest Model

- “Understanding how software actually works will always help someone to use it, but this understanding usually comes at significant cost. The manifest model allows software creators to solve the problem by simplifying the apparent way the software works. The cost is entirely internal, and the user never has to know.”

- User interfaces that abandon implementation models to follow mental models are better.”

(Cooper 1995, p31)

Interface follows the Implementation Model
**Topic 5d: Designing Interfaces and Dialogues – Part 2**

**Interface follows the User's Mental Model**

- main screen
  - everyone
  - loans staff
  - admin
- find items
- check loans
- new items
- remove items
- make loans
- accept returns

**Metaphors**

Metaphors are used to draw an analogy between the way the dialogue works and some object the user is familiar with...

- originally a “trash can” on Macintosh. Note the difference in the metaphor!
- implies the ability to take work home
- a common storage & classification device

**Choose your metaphors carefully**

A metaphor should:
- make it easier for the user to understand the function and purpose of the software
- NOT limit the range of useful features offered
- enable the user to predict ALL the functions available
- NOT behave differently on computer from real life

**Graphical Interface Controls**

- **Control**
  - Text box: gadgets
  - List box: [gadgets, widgets]
  - Combo box: [gadgets, widgets, doodads]
  - Option button:
  - Check box:
  - Toggle button:
  - Command button:

- **Data entry mode**
  - Text box: free text
  - List box: select one item from a list
  - Combo box: select an item or enter text
  - Option button: choose one option only
  - Check box: choose one or more options
  - Toggle button: toggle choice on or off
  - Command button: perform a preset function

**Text Boxes for Display**

- A text box is used to display data. The form and text box are BOUND to a column in the data table
- Normally you set the box so that it is obvious this is OUTPUT. You can do this by making the box the same colour as the background and removing the box outline
- This field will be skipped in the Tab order. If the user selects the field they can copy the data but not change it (locked).

**Text Boxes for Input**

- A text box can be used to input many types of data. The text box will be UNBOUND for input only, or BOUND to a column if you want to display and alter values.
- Normally you set the box so that it is obvious this is INPUT. You can do this by making the box a different colour from the background and having a clear box outline (these are the default settings).
- This field has formatting prompts as well as automatic checks for numeric data.
List Boxes for Input

- A list box can be used to restrict input of data to values currently saved. The list box is BOUND to the relevant column in the data table in which the values are stored.
- You can arrange the size of the box to display as many values at once as you like. If more values are available a scroll bar will appear.
- List boxes like these prevent the user entering invalid data but do require the use of a mouse.

Properties

Control Source | Name | Row Source
--- | --- | ---
Table/Query | Select (Name) From [gizmos]; widgets

Combo Boxes for Input

- A combo box displays the list of current values but also allows the user to type in new values or to select current values from the keyboard.
- The combo box may be BOUND to a column in the data table from which the values will be displayed OR the possible values may be specified in the property sheet.

Properties

Control Source | Name | Row Source Type | Row Source
--- | --- | --- | ---
Name | gizmos; widgets; doodads | Value List

Other choices for Input

- Option buttons, check boxes and toggle buttons can be used to provide a choice from a range of alternatives.

Options:
- Option button
- Check box
- Toggle button

- Typically these controls will be placed in a group so that the range of choices will be clear.

Command Buttons

- Command buttons are used to carry out program steps such as saving a record, accessing another file, etc.

Properties

On click | [Event Procedure]
--- | ---
The event procedure may be generated using a wizard or may be coded as a macro or in Visual Basic.

Note the CD player metaphor in this example. Is it a good one?

Designing Dialogues

- The process of defining the manner in which humans and computers exchange information
- Analogous to a conversation between 2 people
- Interface and dialogue design is critical for successful information systems
- “...to the user the interface is the system”
- Should provide a uniform structure for finding, viewing, and invoking different components of an information system
Designing Dialogues

3 major steps:
1. Designing the dialogue sequence
   - based on how the user wishes to use the system
2. Build a prototype
   - use tools such as CASE tools, Visual Basic, Access
3. Assess Usability
   - use same approach as with forms and reports - time to learn, speed of performance, rate of errors, retention over time, subjective satisfaction

Interaction Methods

- Common interface methods are:
  - Command language interaction
  - Menu interaction
  - Form interaction
  - Icon/object-based interaction

Command Language Interaction

- Users enter explicit commands to invoke operations
  - users must therefore remember syntax and semantics
- MSDOS, UNIX commands
  - copy A:\myfile.doc to B:\myfile.doc
- Complex applications such as word processors and spreadsheets may have many commands
  - <CTRL-P> print
  - <CTRL-S> save
  - <CTRL-C> copy

Menu Interaction

- Users select from a list of system options and a specific operation is executed
- Menu interaction is still the most widely used interaction method
- Menus can differ greatly in their design and capability
  - Simple single menu
  - Hierarchy of menus
  - Drop-down menus
  - Pop-up menus
  - Hyper-text/hyperlink menus
  - Graphic menus

Simple Single Menu

```
TEXT-BASED SYSTEM
MENU OF CHOICES
1 BOOKINGS
2 CUSTOMERS
3 TOURS
4 MAINTENANCE
5 LOGOUT
```

Menu Hierarchy
Drop-Down Menu

Pop Up Menu

Hyper-text /Hyperlink Menu

Graphic menus: Switchboards
- switchboards can be useful as an overview of the full system
- switchboards are used by novice and intermediate users, and when the hardware/software restricts navigation techniques
- switchboards are the GUI equivalent to menu structures on text systems
- usually requires many levels of switchboard to cover all functions;
- users will not remember the structure of the set of menus

Graphic menus - Toolbars
- toolbars are an alternative to menus and provide more direct access to commonly used functions.
- toolbars now contain some conventions for “standard” features such as save, cut, copy, paste - try to stick to them
- toolbars group related items for ease of access and learning;
- where you need new tools, choose icons that are standard or fit the metaphor you are using
- what’s wrong with these toolbars?

Graphic menus - Tabbed Menus
- a common technique for selecting from a relatively large number of options.
- provides choice for a range of changes before returning to the original point.
- Note the keyboard options here after the behaviour of the Enter key etc

- toolbars are an alternative to menus and provide more direct access to commonly used functions.
- toolbars now contain some conventions for “standard” features such as save, cut, copy, paste - try to stick to them
- toolbars group related items for ease of access and learning;
- where you need new tools, choose icons that are standard or fit the metaphor you are using
- what’s wrong with these toolbars?
But don't go overboard!!!

Guidelines for Menu Design

- Wording
  - meaningful title, clear, unambiguous command verbs
- Organisation
  - consistent, eg. related options should be grouped together
- Length
  - use sub-menus to break up only very long menus
- Selection
  - methods should be clear and consistent
- Highlighting
  - should be minimised and used to convey selected options (marked) or unavailable options (dimmed)

Menus... Some suggestions

- every function must be available on a menu
- users don’t understand hierarchical menus - avoid them as far as possible - stick to single level menus
- don’t put bang (immediate action) menu items on a top-level menu

And a few more...

- put icons on menus so that users learn what they mean by familiarity
- disable menu items that are not available
- use toggle (flip-flop) menu items where appropriate.
- use short cuts when possible

Don’t Duplicate!!!

Form Based Interaction

- Users fill in ‘blanks’ on a form when they interact with the system
- Screen format is similar to paper-based forms
- Effective for both input and presentation of data
- Form interaction is common on many applications, the World Wide Web
- Form-based programming languages e.g. VB.NET

(See Topic 5e)
Icon/Object-Based Interaction

- Often based on metaphor
- Icons are graphic symbols that look like the processing option they represent
- Operations are selected by pointing at and clicking on the desired option
- Easily understood and take up little screen space
- Many icons appear in the form of buttons or controls

Sentences and Paragraphs - Closure Points

- In human speech, closure points enable the speaker to take a breath and the listener to respond. In writing, closure points are given by ends of paragraphs and chapters.
- In human-computer dialog, closure points allow the user to take a short mental break. Where delays in processing occur, these cause least interruption at closure points.

Closure Points and Data Validation

- Closure points are obvious stages at which to check for errors in the data presented so far.
- So ...
  - use closure points as logical places to allow the user to check and correct the data entered so far
  - adjust the level of data validation to the level of closure point
  - remember - error messages interrupt the flow of the dialog - attempt to minimise this interruption

Error Messages

- Error messages may place the user in a different mode.
- In this example, the user is required to change from one mode (the mode of the main process) to respond to the error (error handling mode).
- Modeless feedback is preferable.

Approaches to Errors

- There are several ways of handling errors:
- Restrict the user’s opportunities for making errors:
  - get users to select values rather than type them in (can use typing to select values though)
  - don’t accept invalid formats (go back to start of field and insist on correct format) - take care with this!
- Tell the user what is acceptable and validate the input
  - give patterns for data entry fields
  - show rules, examples next to the field
- Accept everything the user enters and deal with it somehow (Cooper’s preference)
  - save whatever the user enters
  - offer alternatives
Summary - Good user interface design is ...

Making the computer user literate

Summary - Bad user interface design is ...

Not making the user, computer literate

References

Material for this lecture prepared by Ms Helen Smith and Mr Ryan Fernandes.

Chapter 14, 15

Chapter 13, 14, 15

References

- Don Norman’s The Design of Everyday Things (Doubleday Currency 1986)
  - Very useful and general discussions of the problems of designing interfaces applied to doors, windows, watertaps and computer systems. A wonderful book.
- Alan Cooper’s About Face: The Essentials of User Interface Design (Programmer’s Press, 1995)
  - Practical advice combined with criticism of a number of standard aspects of interface design. Amusing, exciting, useful and frustrating.
  - The rules for designing consistent interfaces. Specific to Windows 95, so of no use at Monash but useful at home.