Human Computer Interface Design
Chapter 7 – User Interface Elements Design and Guidelines

Objective

UI Guidelines provides information on the theory behind the UI Elements "look and feel" and the practice of using UI components. They include many examples of good design and explain why one implementation is superior to another. This lecture describes:

- Roles played by guidelines in design
- Limitations of guidelines
- Guidelines as general design and evaluation principles
- Use of guidelines in specific contexts, e.g., screen, design, use of color, etc.

Outlines

1- Design of WIMP-Based UI (WIMP= Windows, Icons, Menus, Pointers)
2- Bad Design Examples for GUI Elements from Interface Hall of Shame Web Site
3- Design Guidelines: The Windows Interface Guidelines for Software Design, Macintosh Human Interface Guidelines, Java Look and Feel Guidelines, OSF Motif Standard Interface
4- RealThings Design Guide: An Alternative Approach to GUI Design

Graphical User Interface (GUI) Elements Taxonomy

GUI = WIMP User Interface = Windows, Icons, Menus, Pointers User Interface

- Windows, Panel and Frames
  1- Primary windows, secondary windows, Utility windows (a window whose contents affect an active primary window), and Plain windows (no title bar or window controls) provide the top-level containers for your application.
  2- A panel is a container for organizing the contents of a window or dialog box: panel, scroll panel, tabbed panel (tab cards), toolbar, group boxes and headings
  3- An internal frame is a container used in MDI (Multi-Document Interface) applications to create windows that users cannot drag outside of the desktop panel

- Icons: graphic representations of objects such as documents, storage media, folders, applications, and the Trash. Icons look like their real-world counterparts whenever possible.

- Controls: Command Button, Toggle Button (buttons with two States e.g. On/Off), Checkbox, Radio Button, Combo Box, Menu, Pop-Up Menus (Contextual Menu), Toolbar

Graphical User Interface Elements Taxonomy (Components)

- Dialog Boxes
- Text Components
  1- Uneditable information displays: Label, Progress Bar, Lists
  2- Editable displays: password field, text window, colors selector, files explorer

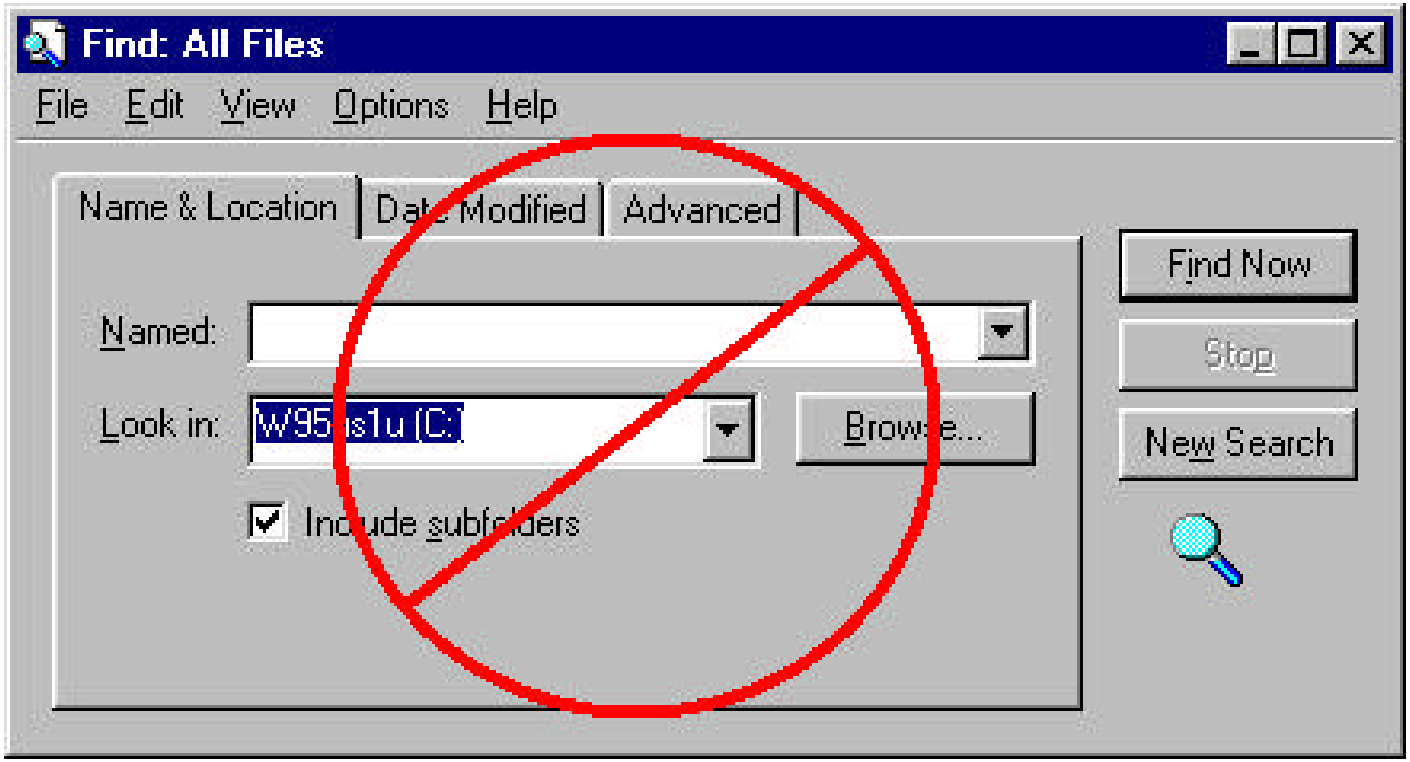
- Behaviors: mouse operations, drag-and-drop operations, Keyboard operations including the use of mnemonics, keyboard shortcuts
- Color
- Language including conventions for names in buttons, labels for checkboxes and radio buttons, messages in dialog boxes, online help systems, and manuals

Advanced Controls, Dialog Boxes, etc.

- Tables and Grids, Trees, Graphs, OLE/ActiveX Controls
- Objects Explorer (Files, Color, Settings, etc.)
- Animation: Progress and Delay Indication Bars, System Status Animation, Slides (to let the user enter a numeric value bounded by a minimum and maximum value)

GUI Elements Bad Design: Find Applet

From The Interface Hall of Shame Web Site (http://www.iarchitect.com/mshame.htm)



The Find Applet is a truly bizarre mix of interface elements: a dialog box with a menu bar. This is typically regarded as a common mistake for beginning UI programmers

- The Find dialog exists as both a dialog box and a document-centric application. Confused?

- Why an Edit menu? What is there to cut, copy or paste? Since this dialog box has no Cancel button, the first menu the user is likely to use is the File menu, since it contains the Close command.

- The Find window is not really a dialog box. Well, it is - sort of - until you perform a search. The menus become relevant only after you have hit the Find Now button (as opposed to Find Later?), and they are only relevant if the search was successful. As soon as the user presses Find Now, the dialog box becomes taller to display the results of the search. Of course, if you've moved the window to the lower portion of your screen, you might never know that your search was successful.

Solution

- The basic problem is that the Find window attempts to display together content that really should be displayed separately. There should be an initial Find dialog for the user to enter search criteria, and a separate results window.

- This would allow the design of a window optimized for its distinct purpose, without confusing the user with extra irrelevant controls.

Solution Justification

- Rather than having to rely on the rationale "I just think it would be better", they can have the weight, research, and opinion of others with them through guidelines.

- There has always been a rather simple rule to follow when designing a window: dialog boxes don't have menus. Menus indicate a document-centric application (e.g. Word).

Roles of User Interface Design Guidelines

1- Promote Awareness of Concepts
   Cascading menus: A cascading or hierarchical menu is a sub-menu attached to the right side of a menu item…

2- Sound basis for decisions regarding users
   Locating a command line for type in: Place the command line near the bottom of the screen unless it is clear that the user's gaze will be elsewhere…

3- Offer Strategies for Problem Solving
   Scrolling through large amounts of data: When your application organizes data logically into pages, provide page-oriented scroll bars…

4- Support Evaluation
   Evaluating a menu design according the guidelines

Macintosh Human Interface Guidelines

The Macintosh Human Interface Guidelines defines the elements and behaviors of the Macintosh standard UI:

- Menus
- Windows
- Dialog Boxes
- Controls
- Icons
- Color
- Behaviors
- Language

This guideline presents examples of the right and wrong ways to use interface elements and behaviors for Macintosh applications. It also shows how to combine the pieces of the interface with behaviors, aesthetics, and language to create a superior product.

Dialog Box Design from The Macintosh Human Interface Guidelines

Four types of dialog boxes:

1- Modeless dialog boxes, which are useful for getting user input and for making changes to a document. Once open, they are available until the user closes them.

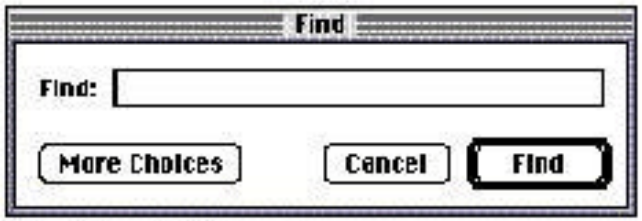
2- Movable modal dialog boxes, which are useful for requesting user input and for making changes to a document while allowing the user to switch to another application. Also useful for allowing the user to see parts of a document that might be obscured by a modal dialog box.

3- Modal dialog boxes, which are useful for forcing the user to provide necessary information before carrying out the current operation.

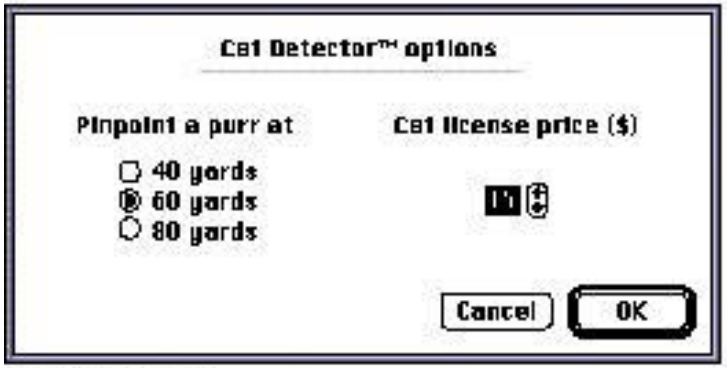
4- Alert boxes (a type of modal dialog box), which are useful for communicating error conditions or preventing any other activity until the user responds to the error conditions

Macintosh Human Interface Guidelines

Modeless dialog box



Movable modal dialog box



Modal dialog box

Alert box

Behaviors Design from the Macintosh Human Interface Guidelines

- Behaviors design guidelines cover the interaction between the user and the computer, detailing how the computer should respond.
- It contains information about the mouse and the pointer, the keyboard, and the behavior of different types of objects such as text, graphics, and arrays.
- It discusses the behaviors associated with the primary user input devices - the mouse and keyboard.
- It also describes responses generated by your application, such as selection behavior and keyboard navigation through lists.

Examples of Mouse Actions: double-clicking and clicking

- The most common use of double clicking is to provide a shortcut to other actions. For example, clicking an icon twice is a faster way to open it than clicking once to select it, then choosing Open from the File menu.
- Some uses of clicking are to select an object, to move an insertion point, to activate a button, and to turn on a control such as a checkbox. If the function of the click is to cause an action (such as clicking a button), the selection is made when the button is pressed, and the action takes place when the button is released.

RealThings Design: WIMP World Limits

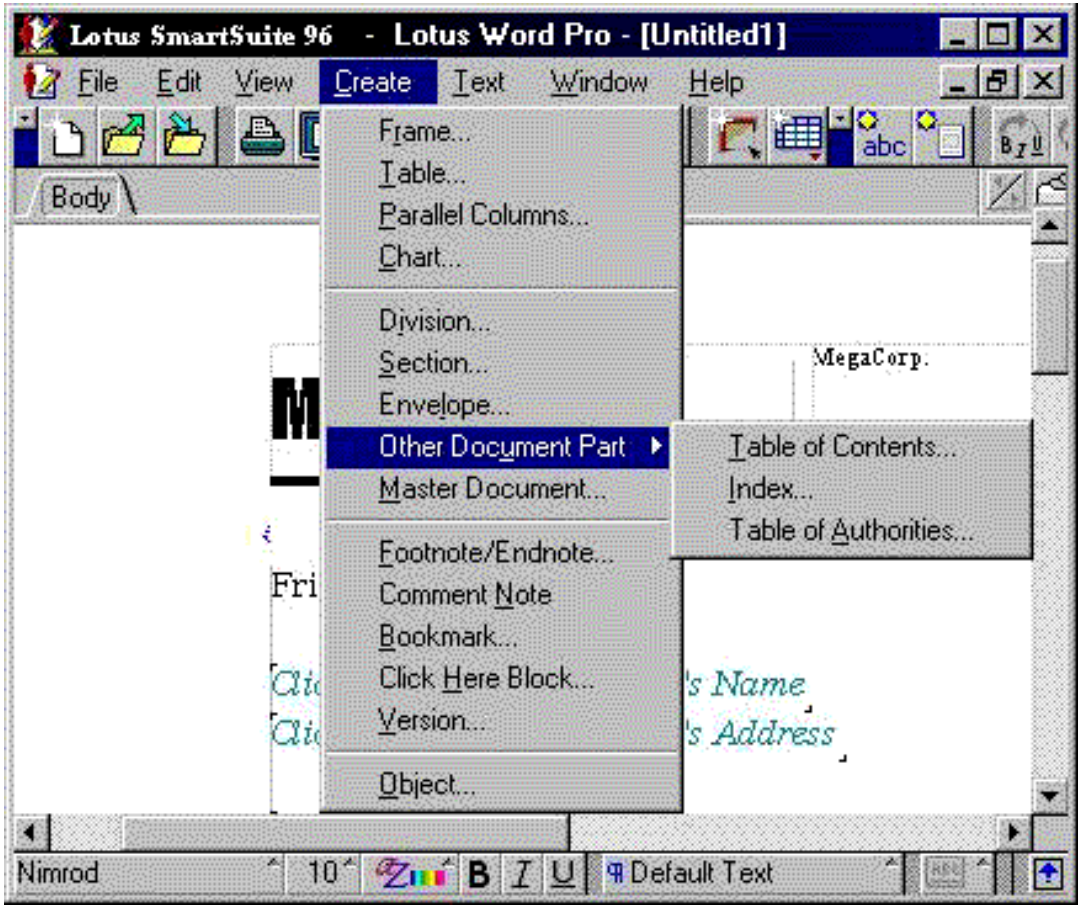
- UI elements (or more usually, applications) do not look real, or even bear more than an occasional passing resemblance to anything in our real world outside the computer

- Menu bars and tool bars proliferate. Users spend much of their time 'mining' menus or hovering over buttons waiting for help

- Icons only show a gross level of information - they indicate the class of object, but rarely impart status information or make important properties apparent.

- As the user fumbles through the interface in search of particular functions, the clutter and visual pollution detracts from the content or task, which should be the user's main concern. The UI itself distracts and intimidates the user rather than helping

Example of GUI [Lotus SmartSuite]



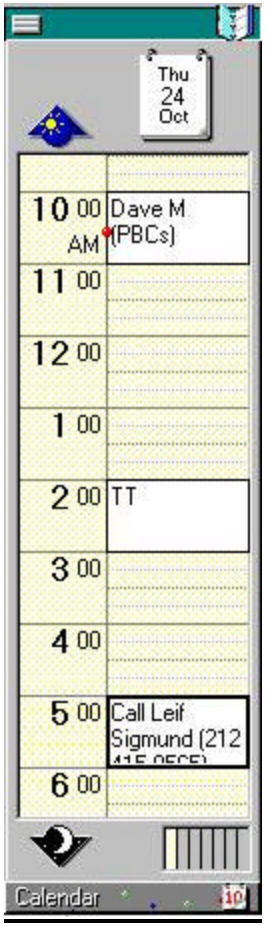
An Alternative Approach to GUI - A Clean Screen UI and RealThings

- In the Lotus Organizer UI, most operations can be performed in place without resorting to menus or tool bars

- Users are presented with a UI that resembles something they already understand - a book (A RealThing Object)

- The Smart Center drawers for Calendar (shown here) and Address are also a natural and very usable extension of this philosophy

- They are unobtrusive yet accessible - combining the value of pull-down menus with real objects that provides valuable, but not over-comprehensive, function for standard and frequent tasks to the user



RealThings: Focus on content, not control

- RealThings exhibit a new, real-world user interface style. They set a new direction in making user interfaces more approachable for novice and casual users. At its simplest, the RealThings philosophy is to make software constructs and applications appear as they would if they existed in the real world outside the computer. Objects and mechanisms are presented in context, and actions are surfaced in more natural ways.

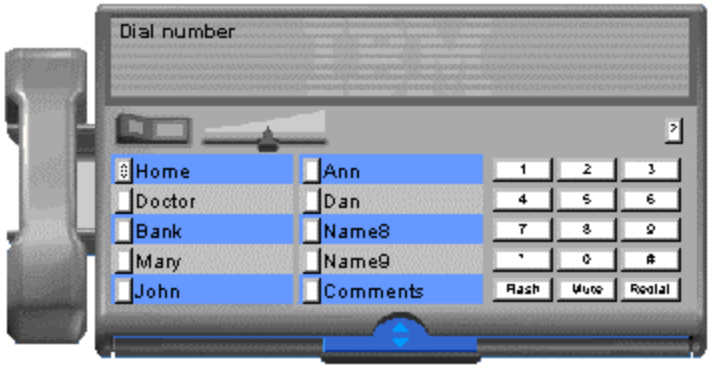
- The key to making RealThings work, as the next major stage in the development of ease of use is knowledge transfer from the real world. By making the behavior of computer objects more natural - more like the user expects from his or her experience of the world - we make it more accessible, less daunting, easier to comprehend and to use.

- Capitalizing on the user's knowledge of real things as well as a design that is well understood and tested by time.

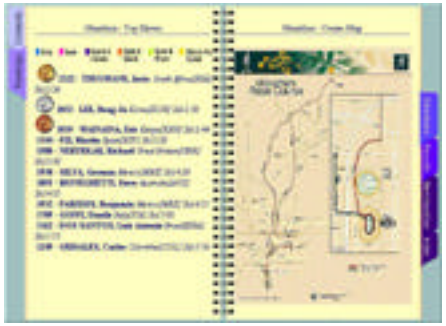
- But RealThings aren't just copying the real world with its inherent limitations, they are providing navigation techniques that enhance the real world, as well as providing mechanisms and interaction cues that are immediately recognizable and understood.

RealThings Examples [From IBM]

Real Phone



Real Book



Real CD