

Controls and Their Behaviors

Jeff Offutt

<http://www.cs.gmu.edu/~offutt/>

SWE 632

User Interface Design and Development

Cooper, Ch 20-24

Ch 20 : Window Behaviors (history)

- Most GUI concepts were developed in the late-60s and early 70s
 - Various academic researchers
 - 1963 MIT PhD student Edward Sutherland – Sketchpad introduced the idea of navigating an entire window
 - IBM TJ Watson (theory only)
 - Xerox's Palo Alto Research Center (PARC)
- Invented : mouse, WYSIWYG, desktop metaphor, windows, scrollbar, button, drop down menus, ...
 - NOT OO programming, as the book mistakenly says
- Xerox built Alto, then Star
 - Expensive, VERY slow, not ready for prime time

GUI Concept to Commercialization

- Apple took ideas from a Xerox demo
 - And hired several Xerox engineers
 - Sold the first commercially successful GUI computer
- Then MS (less than 10 people at the time) convinced Apple to let them develop applications for Apple's computer
 - Apple's Lisa was successful, but a little expensive
- MS then reverse-engineered the code from the Apple to create Windows
 - But very badly – their engineers were not as good
 - Ten years later, MS was convicted of theft, and paid a small fine

4/18/2011

© Jeff Offutt, 2004-2011

3

Mis-statements in Book's History

- Cooper's summary of the history is a little bit disingenuous
 - Biased toward Microsoft
 - Apple's Lisa was not a failure, but the price (NOT \$9995!) was too high for widespread home adoption
 - Multiplan was an unusable failure, and the forerunner of Excel was bought by Microsoft
 - OO started with Dave Parnas in early '70s, and was developed by language designers, software engineers, and graphics programmers at universities through the '70s and '80s
 - OO was adopted by the Mac in the late '80s

4/18/2011

© Jeff Offutt, 2004-2011

4

GUI Window Principles

A number of principles were developed by researchers at universities and Xerox that are still in use

- Use of visual metaphors
- Program state should not change the behavior of the UI
 - Drag-n-dropping files is move on the same drive, but copy on different drives
 - State-based behavior is okay IF the user knows the current state
- Overlapping windows is okay, tiled windows is not
 - “Proving” this was a PhD dissertation around 1984
 - Full-screen applications are okay when users only do one thing
- Multi-panes (eg, tabs) is a good compromise
 - Introduced in X toolkit applications in mid-80s

4/18/2011

© Jeff Offutt, 2004-2011

5

Ch 20 Summary : Window Behaviors

Most concepts and principles in GUIs have been around for more than 20 years ...

It took awhile for hardware designers and marketing to catch up to the research

4/18/2011

© Jeff Offutt, 2004-2011

6

Ch 21 : Using Controls

- Controls are objects on the screen that communicate between users and software
- Four types of controls
 1. Imperative controls : used to initiate a function
 2. Selection controls : used to select options or data
 3. Entry controls : used to enter data
 4. Display controls : used to visually manipulate the program
- We see a limited set of controls
 - Windows and Mac use only a fraction of the invented controls
 - HTML implements very few GUI controls

4/18/2011

© Jeff Offutt, 2004-2011

7

GUI Controls, or Widgets

- Some of the following is taken from Eberts, “*User Interface Design*,” Prentice-Hall, Inc
- Controls were traditionally called “widgets”
 - From graphics programming
 - “Control” probably matches the users mental model
 - “Widget” matches the implementation model

Widget : A building block for an interface

4/18/2011

© Jeff Offutt, 2004-2011

8

Widget History (2)

- The X windows library (late '80s)
 - X window functions stored in a standard collection of libraries for use on many computers
 - Public domain (this is one of the precursors to open software)
 - Very low level
- Widget Sets : Higher level functions built on top of low level libraries (such as X)
 - Motif (X)
 - DEC windows (X)
 - HyperCard (Macs)
 - Microsoft windows (PCs)
 - *Curses* – the original, Unix, non-graphics, lower level than motif

4/18/2011

© Jeff Offutt, 2004-2011

9

Widgets – Form

- A parent of other widgets
- Use when a set of widgets needs to be aggregated
- No callbacks
- Don't confuse with data input form – completely different

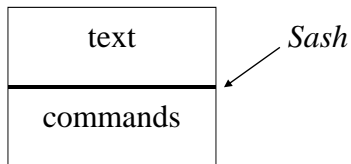
4/18/2011

© Jeff Offutt, 2004-2011

10

Widgets – Paned Windows

- Window with two areas, separated by a sash
- Used to offer the user flexibility
- Most often used between text and command or menu windows
- Outline view in PPT



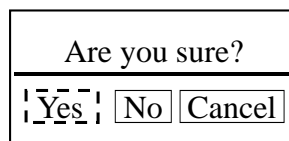
4/18/2011

© Jeff Offutt, 2004-2011

11

Widgets – Dialog Boxes

- Make sure that labels are clear and unambiguous
 - *Yes / No* is sometimes not as clear as *Yes / Cancel*
- The user must know what will happen when a choice is made
- Built-in dialog boxes are often convenient, but sometimes not exactly what is needed



4/18/2011

© Jeff Offutt, 2004-2011

12

Widgets – Pull-down Menus

- List should be short (7 ± 2)
- Can be longer if there is a clear ordering, or searching is easy
(states, courses, ...)
- Can be longer if list is fixed and syntax must be perfect
 - That is, we can trade off searching time for errors on data entry

4/18/2011

© Jeff Offutt, 2004-2011

13

Types of Control Widgets

Imperative

1. Push Button
2. Butcons / toolbar

Selection

3. Checkbox
4. Flip-flop
5. Radio box
6. Combutcons
7. List
8. Combo box
9. Tree control

Entry

10. Bound Value
11. Spinner
12. Text

Display

13. Label
14. Scroll bar
15. Sliders
16. Thumbwheels
17. Splitters
18. Drawers

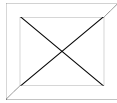
4/18/2011

© Jeff Offutt, 2004-2011

14

Imperative Controls: 1. Push Button

- Used to activate a particular action
- This is usually unrecoverable ... do not use in dangerous situations without a hesitation
- Usually offers no feedback



Label
Parent
Callback

4/18/2011

© Jeff Offutt, 2004-2011

15

Imperative Controls: 2. Butcons / Toolbar

- Used to collect a group of push buttons together
- Can be thought of as a “menu of buttons”
- Buttons are usually
 - Square
 - Use an icon (no text)
- Icons are sometimes confusing, hence tooltips were added to explain what the butcons do

4/18/2011

© Jeff Offutt, 2004-2011

16

Selection Controls: 3. Checkbox

- Allows users to make single binary choices
- Simple, visual, elegant
- “Latching butcons” combine butcons (an imperative control) with checkboxes
 - Selecting bold, italics, underline in MS Office products

4/18/2011

© Jeff Offutt, 2004-2011

17

Selection Controls: 4. Toggle Switch or Flip-flop

- Can be used to inform user or to allow user to make choices
 - Two states: on or off
 - Indicates state by color, icon, text, ...
- Two square buttons with a double-line border. The left button contains a solid black upward-pointing arrow. The right button contains a solid black downward-pointing arrow.
- Problem is the ambiguity ...
 - Do these buttons tell us the current state or indicate the next state if the button is pressed?

4/18/2011

© Jeff Offutt, 2004-2011

18

Selection Controls: 5. Radio Box

- Selects one from of a set of mutually exclusive options
- Actually a specialized menu (single-selection)
 - Collection of checkboxes
- Uses :
 - set some state in system
 - set options for customization
- 3 – 8 options
- Originally diamonds, MS changed to circles
- Very fast, reduces errors, easy to learn but uses a lot of screen space
 - Cooper is negative, but forgets about speed

4/18/2011

© Jeff Offutt, 2004-2011

19

Selection Controls: 5. Radio Box (2)

- A variant is the “radio butcon”
- Small group of mutually exclusive choices
- If crowded, use a drop-down list (slower)
- Same as single selection lists



4/18/2011

© Jeff Offutt, 2004-2011

20

Pull-down Menus Vs. Radio Buttons

- They both accomplish the same task
- Radio buttons are fixed on the screen, Pull-down menus show up on-demand
- Radio buttons are faster to learn, faster to use, less error prone
- Use radio buttons except when :
 - More than 5 choices
 - Screen is crowded
 - Choices available depend on other selections (the choices change dynamically)

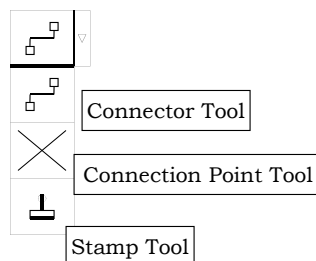
4/18/2011

© Jeff Offutt, 2004-2011

21

Selection Controls: 6. Combutcons

- A button that opens a menu of latching butcons
- A variant on drop-down menus
- Visio's connector tool :



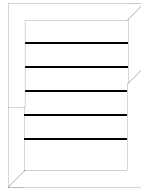
4/18/2011

© Jeff Offutt, 2004-2011

22

Selection Controls: 7. List

- Use as a menu
- Sometimes called *picklists*, *list boxes* and *listviews*
- All the menu guidelines should apply here
- Sometimes text only, sometimes text + icons



4/18/2011

© Jeff Offutt, 2004-2011

23

Selection Controls: 7b. List Choices

- Earmarking: Checkboxes are added to a long list to allow users to choose multiple items
 - Usually involves scrolling, which increases memory burden
- Dragging and dropping from lists: Two lists, and elements can be moved from one to the other
 - *sftp* allows this to copy from one computer to another
- Ordering lists: Moving list items around
 - *PPT* uses this for slide animation
 - *Mozilla* allows this to rearrange bookmark menus
- Horizontal scrolling: Avoid, especially with text

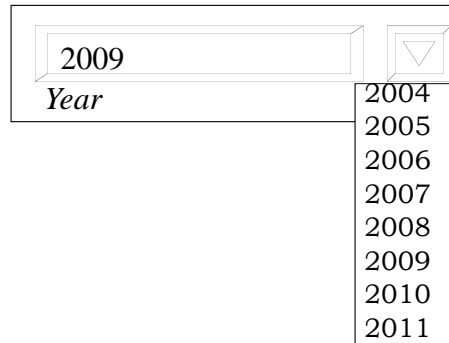
4/18/2011

© Jeff Offutt, 2004-2011

24

Selection Controls: 8. Combobox

- Combines a list box and a *text edit* field
- Users can type or select
- Optimizes for speed, flexibility, accuracy, and fast learning
- Text box can allow arbitrary (*invalid*) inputs or only inputs within the range



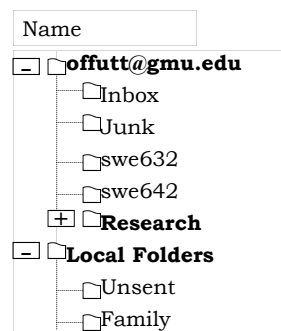
4/18/2011

© Jeff Offutt, 2004-2011

25

Selection Controls: 9. Tree Control

- Lists of hierarchical data
- Usually shown sideways and often with icons
- Useful when the data is “naturally” hierarchical – and programmers think more hierarchically than most users
- Thunderbird mail folders:



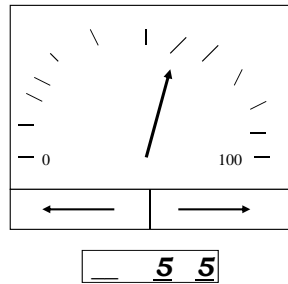
4/18/2011

© Jeff Offutt, 2004-2011

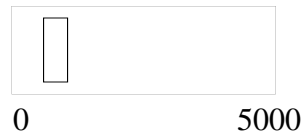
26

Entry Controls: 10. Bound Value – Scale

- Use when user needs to select a value from a large range
- Often combined with text selection for flexibility
- Only allows valid inputs
 - Applies data immunity



Scale



4/18/2011

© Jeff Offutt, 2004-2011

27

Entry Controls: 11. Spinner

- Use when user needs to select a value from a range
- The value has to be precise, but the range is fairly large
 - Age
 - Weight
 - Day of year
- Can be combined with text selection for flexibility
- Edit window can allow arbitrary (*invalid*) inputs or only inputs within the range



4/18/2011

© Jeff Offutt, 2004-2011

28

Entry Controls: 12. Text Box

- Appropriate size (vertically and horizontally)
- Do not use when it is possible to select
- This is the most flexible but slowest and most error-prone selection method
 - Also can present security vulnerabilities
- Operations: *insert, delete, copy, cut, paste, select*
- Text boxes must be validated
 - Active: Invalid keystrokes are ignored
 - Passive: String is checked after user enters data

4/18/2011

© Jeff Offutt, 2004-2011

29

Display Controls: 13. Label (Text)

- Use for simple feedback
- Usually combined with other widgets
- Simple text
- No events

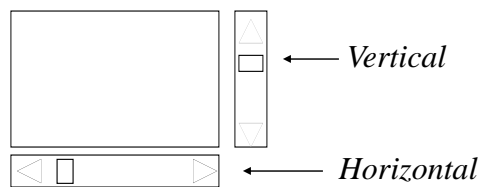
4/18/2011

© Jeff Offutt, 2004-2011

30

Display Controls: 14. Scroll Bar

- Use when list or text is too long to fit in the window
- Better to expand the window, if possible
- Horizontal scrolling is very hard
- Scrollbars require mixing fine motor control (holding a button on a tiny icon) with large motor control (moving your arm)



4/18/2011

© Jeff Offutt, 2004-2011

31

HTML Widgets – Form Elements

1. Textboxes
 1. Password boxes
 2. Text areas
2. Radio buttons
3. Checkboxes
4. Menus
5. Buttons
 1. Submit
 2. Reset
6. Tab order
7. Keyboard shortcuts

4/18/2011

© Jeff Offutt, 2004-2011

32

Ch 21 Summary : Using Controls

Don't use the first control that comes to mind.

Think about the user's needs and design the right solution

4/18/2011

© Jeff Offutt, 2004-2011

33

Ch 22 : Menus: The Pedagogic Vector

- Standard menus
 - File ... Edit ... Help
 - Pretty poor design, but inertia is strong
 - Based on an implementation model, not user goals
- This chapter is about how to make menus better ...
- Without totally redesigning them

4/18/2011

© Jeff Offutt, 2004-2011

34

Notes on Typical Menus

- **General notes**
 - Accelerator : keystrokes that activate functions – Ctrl-P
 - Mnemonic : keystrokes that activate with a menu – File-Close
 - Separator : Solid line that separates groups of commands
- File menu : Most recently use list is a shortcut
 - But it is “verb-object” – PPT then the file
- Edit menu : Don’t use as an “everything else” menu
 - Mozilla (1.4) put “preferences” on Edit ... confusing
- Windows menu : For multiple windows of the program
- Help menu : Needs notes on shortcuts

4/18/2011

© Jeff Offutt, 2004-2011

35

Notes on Optional Menus

- View menu : Options to let users look at data differently
 - Also, options for controls
- Insert menu : Ways to add data to document
 - Combine with Edit menu if there are not many choices
- Settings menu : Allows user to change settings in program
 - Should be comprehensive and include all available settings
- Format menu : Change properties of visual objects
 - Users always expect the properties to be with the objects, not separate
- Tools menu : Advanced and particularly powerful functions

4/18/2011

© Jeff Offutt, 2004-2011

36

Difficult Menu Design Choices

- Cascading menus : This is a hierarchical menu, which is confusing to many users
 - Don't hide frequently used functions in a hierarchy
- Expanding menus : Hiding items that are not used often, then showing if the user waits
 - Very annoying
 - Experimental research found that it slowed users down
 - How to turn it off in PPT?
- Bang menus : Putting command functions on the top level menu
 - Cooper says don't use them

*PPT 2003: Tools –
Customize – Options –
Always show full
menus
(Not Tools – Options!?)*

4/18/2011

© Jeff Offutt, 2004-2011

37

Menu Conventions

- Disable menu items if they are not currently relevant
 - Keep them visible, but “grayed out”
 - Don't mock users with a “made you look” action
- Checkmarks can show when items are turned off or on
 - Firefox “Go” shows current page and Thunderbird shows “View Display Attachments Inline”
- Flip-flop menu items show what is currently not available
 - Confuses users who think it shows state, not potential action
- Icons on menu items help users find items quickly
 - Pictures are faster to recognize, but text is more precise

4/18/2011

© Jeff Offutt, 2004-2011

38

Keyboard Accelerators

- Help users invoke a function from the keyboard
 - All tools in a “collection” should have the same style and same accelerators when possible
 - Users must be able to find them
- Conventions from ancient history (early '80s)
 - Ctrl-C, Ctrl-V
 - Ctrl-P
 - Ctrl-S
 - Ctrl-Z, Ctrl-Y
- Which commands should have accelerators?
 - Let users define their own (power users!)
- PPT accelerators are a real mess ...

4/18/2011

© Jeff Offutt, 2004-2011

39

Ch 22 Summary : Using Menus

Getting menus right is simple ...

**Unfortunately,
it's even easier to get them wrong**

4/18/2011

© Jeff Offutt, 2004-2011

40

Ch 23 : Using Toolbars and ToolTips

- Toolbars provide fast access to commonly used functions for knowledgeable users
 - Not much help for new users – may detract from learning
 - They offer a kind of shortcut
- Use menus to teach and to access seldom-used functions
- Use toolbars for fast access
 - Unfortunately, new users tend to learn the toolbars and never go to the menus ...
- How do users know what toolbar buttons do?
 - The icons are good for reminding, but not for teaching

4/18/2011

© Jeff Offutt, 2004-2011

41

Tool Tips

- An old idea in the research literature in the '70s
- Apple tried a version that always popped up (balloon help)
 - It was a too intrusive for expert users
- Windows made them smaller and delayed activation
 - They use the events “hover” not “mouseover”
 - This is much more effective than balloon help

4/18/2011

© Jeff Offutt, 2004-2011

42

Tactics for Tool Bar Buttons

- 1) Use icons, not text
 - Text takes longer to read
 - Text + pictures use a lot of space
 - The little helper flags (tool tips) are great!
- 2) Buttons should not disappear
 - Inactive buttons should not “depress” when clicked
 - They should be grayed out, if possible
 - The worst error messages are those that say “you can’t do that now.”

4/18/2011

© Jeff Offutt, 2004-2011

43

Tactics for Tool Bar Buttons (2)

- 3) When tool bars are available, menus are used to teach
 - Hmm ... new users may never go to the menus ...
- 4) Types of buttons on tool bars
 - Momentary (traditional)
 - Push it and it activates something
 - Used to start a function
 - Latching
 - Button stays depressed
 - Used to change state
 - Pop-ups

4/18/2011

© Jeff Offutt, 2004-2011

44

Tactics for Tool Bar Buttons (3)

5) Tool bars should be customizable

- Movable
- Reshapable
- Allow users to add / remove buttons
- Allow users to change size of buttons
- Office has lots of customization options, but it's very hard to learn to use them
 - Sometimes PPT and Word will override users customizations

6) Magnetic buttons

4/18/2011

© Jeff Offutt, 2004-2011

45

Ch 23 Summary : Using Toolbars and ToolTips

The Toolbar is very useful technology and still evolving

4/18/2011

© Jeff Offutt, 2004-2011

46

Ch 24 : Using Dialogs

- Dialog boxes are the most inconveniently designed part of most GUIs (*according to Cooper*)
- We have already talked about:
 - Unnecessary dialog boxes
 - Placement
 - Confusing button labels
- Remember : You are designing a language
 - Unfortunately, you have been trained to expect languages to be poorly designed (C, C++, Z), but you can do better!
 - You have the advantages of a class in UI design as well as a degree in SWE (some of you)

4/18/2011

© Jeff Offutt, 2004-2011

47

Dialog Boxes (2)

- Dialog boxes are interruptions and inherently intrusive
 - We cannot always afford to spend 20 minutes talking every time someone comes to our office
- Dialog boxes are excise tasks
- Use dialog boxes for :
 - Exceptional interaction (errors, printing, ...)
 - Dangerous interaction (requiring extra concentration)
- My favorite worst use of a dialog box:
 - Find in Netscape, MS Word and PPT
 - Adobe integrates find into the toolbar
 - Firefox uses keyboard accelerator: “/string”
 - This is the first use I’ve seen of using advanced vi-style navigation in a GUI ... why not more ?
 - Why doesn’t Chrome do that ?

4/18/2011

© Jeff Offutt, 2004-2011

48

Modal Dialog Boxes

- Modal : No other interaction is allowed until the dialog box is closed
- Error messages should almost NEVER be in modal dialog boxes
 - The message should disappear with the next interaction with the parent window
- Modal boxes are :
 - easy to program
 - easy to understand
 - annoying
 - too common

4/18/2011

© Jeff Offutt, 2004-2011

49

Modeless Dialog Boxes

- The “*owning*” program continues
- Usually have terminating commands (*close*)
- Much less intrusive than modal dialog boxes
- A little confusing – when do they go away?
 - people expect them to be modal
- Example : Find (^F) in MS Office

4/18/2011

© Jeff Offutt, 2004-2011

50

Four Kinds of Dialog Boxes

1. Property dialog box

User can change settings of an object

- font
- printing options

2. Function dialog box

User performs some function

- find
- print
- spell checking

3. Process dialog box

Tells user the system is busy

- Hour glass is not always sufficient
- Process box should:
 - Explain what's happening
 - Express that it is unusual
 - State how much longer ... this is hard!
 - Provide a *cancel*

4. Bulletin dialog box

Gives the user feedback

- error message
- confirmation messages

User does not request these!

4/18/2011

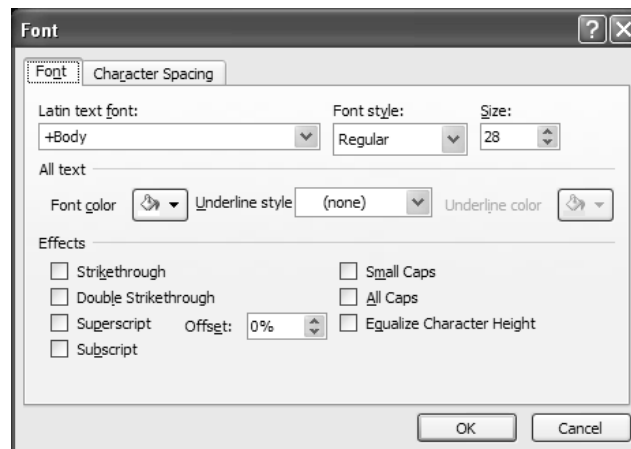
© Jeff Offutt, 2004-2011

51

(1) Property Dialog Boxes

User can change settings of an object

- font
- printing options



18-Apr-11

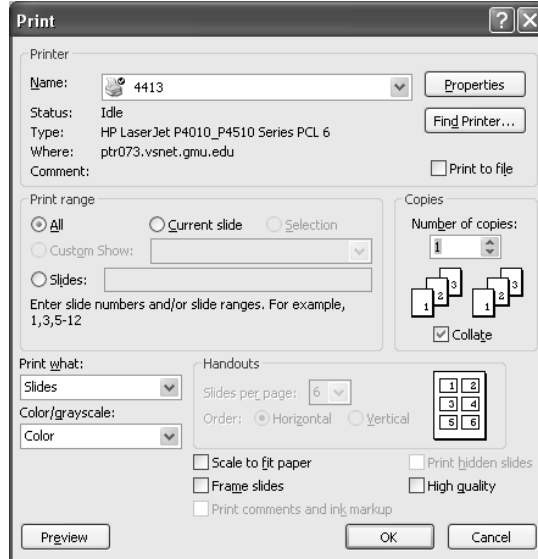
© Jeff Offutt, 2010-2011

52

(2) Function Dialog Boxes

User performs some function

- find
- print
- spell checking



18-Apr-11

© Jeff Offutt, 2010-2011

53

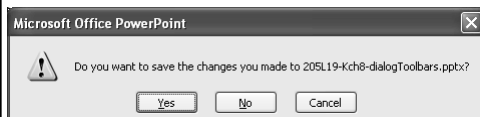
(3) Bulletin Dialog Boxes

Gives the user some feedback

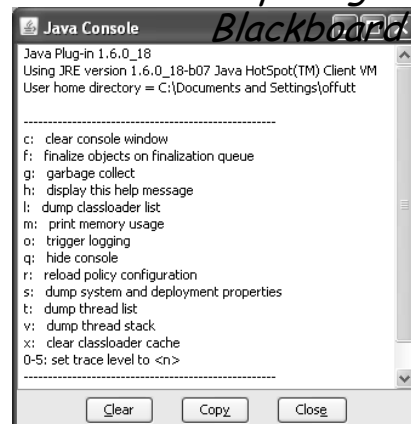
- error message
- confirmation messages

Note that the user does not request!

When opening



Our favorite



18-Apr-11

© Jeff Offutt, 2010-2011

54

(4) Process Dialog Boxes

Tells the user the system is busy

- Hour glass is not always sufficient
- Process box should:
 - Explain to the user what's happening
 - Express that it is unusual
 - State how much longer ... this is hard!
 - Provide a *cancel*

*I couldn't find a
good example*

Summary : Using Dialogs

**Dialog boxes are hard to get right and
often very annoying**

Dialog Etiquette

- A lot of UI design is about writing software that is polite to users – users simply want respect
 - Respect my goals
 - Respect my time
- Property and function dialogs are requested by users
- Bulletins and process dialogs are sent by the software

4/18/2011

© Jeff Offutt, 2004-2011

57

Dos and Don'ts about Dialog Boxes

- All dialog boxes should be movable
- Dialog title bars should indicate function (verb) and object
- Dialog boxes should behave as transient programs, but be as small as possible
- If a user moves a dialog box once, it should remember and return to its new home
- Don't use them unless necessary
- Always include OK and Cancel
- Semantics of a close button on a dialog box is unclear – don't use it
- "Help" is not equivalent to "cancel" and should not be adjacent

4/18/2011

© Jeff Offutt, 2004-2011

58

Variants of Dialogs

- Tabbed dialogs allow convenient organization
- Stacking tabs is confusing
- Expanding dialogs – hide rare options and make them accessible through a button
 - If I go to the “expert” pane once, it should reappear the next time
- Cascading dialogs – dialog boxes beget more dialog boxes ...
 - Like coming to office hours to ask me a question, then making me give the answer to someone else, who then brings in a third friend for clarification ...
- Dynamic dialogs – A selection in one pane changes the contents of another
 - Confusing but useful
 - PPT (2003) – More Buttons – Add or remove buttons – Customize
 - PPT (2007) – Quick access toolbar – More commands – Customize

4/18/2011

© Jeff Offutt, 2004-2011

59

Summary : Dialog Etiquette

**Dialog boxes are often poorly designed
because they are NOT designed**

**Programmers use default built-in boxes
without thinking**

4/18/2011

© Jeff Offutt, 2004-2011

60

Summary : Engineering

**Designers should design
Programmers should program**

Summary of Controls

If you want to examine the impact that Cooper has had on GUI design, compare MS Office 2003 & 2007 products with previous MS Office versions