Graphical User Interfaces (GUI)

James Baldo Jr.

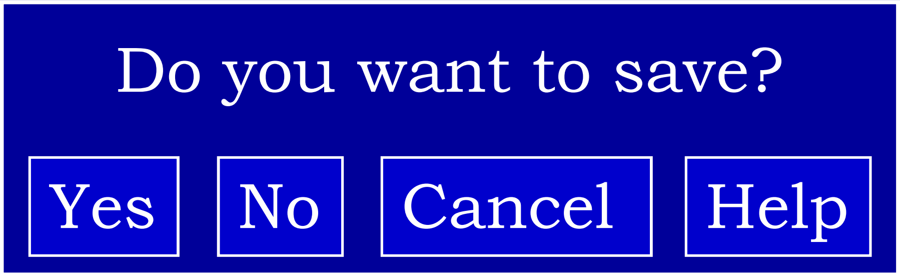
SWE 432

Design and Implementation of Software for the Web

GUI Flow

- **Flow**: The next thing the interface wants to do is exactly what the user expects.
  - Follow users' mental model
  - Let the user direct the software
  - Don't talk with the user
  - Keep all related tools available
  - Modeless feedback : The user should not have respond
- Interfaces should be invisible, not cool
- It's easy to make things complicated, it's hard to make things simple

Flow Example



Do you want to save?

Of course!!! Dont ask me.

Yes, GUIs are different from CLs by being easier, but slower …
But, that is not an excuse to design them to be slow!

Overhead and Excise tasks

- Overhead relates to solving problems:
  1. Revenue Tasks: Sub-tasks that work to solve the problem directly
     - designing
     - requirements
  2. Excise Tasks: Sub-tasks that must be done but that are not really part of the problem
     - compiling
     - debugging
- Excise tasks often satisfy the needs of the tools, not the users

Excise Tasks

- Excise tasks are trivial, unless we have a lot of them
  - Eliminate them if possible
  - Automate them as much as possible

- Excise for users with comp-semantic knowledge is often perceived as revenue for users without

Memory – Auto-customization

- Remember what the user did the last time
- Avoid unnecessary questions
- Imagine a secretary that asked you every time whether you wanted copies on front and back!
- Dialog boxes ask questions, buttons offer choices

Auto-customization Examples

- MS Word : I always put my files in C:\offutt

  But MS Word always thinks I'm going to open a file in C:\Program Files\ …

  (took me 2 years to find the customization!)

- PPT : I often print "Handouts", "2", "Pure black and white"

  If I print several PPT files in a row, I have to click all three boxes every time!

- ATM : I always withdraw $150.

  Why does the ATM always use $40 and $60 as defaults?

Selection : The Mouse

- “Democracy is a terrible system of organizing a society, but it's better than any of the others”
  - Winston Churchill
- So is a mouse a terrible selection device, but it’s better than any other
- The worst aspect of a mouse is that it forces us to take our hands off the keyboard
- MS Word 2010

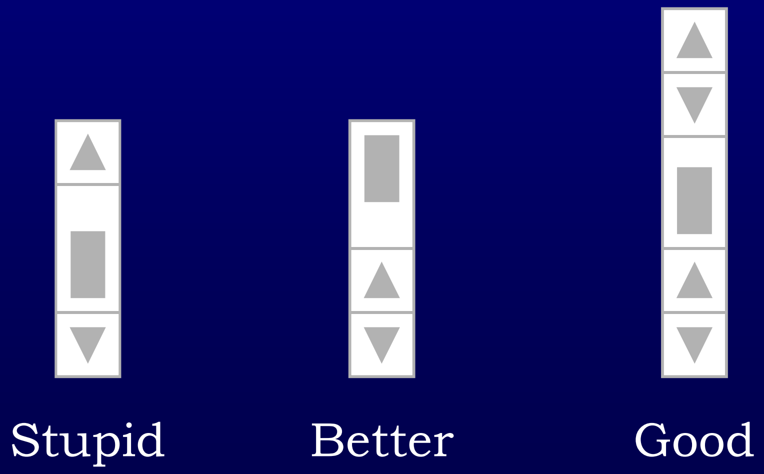
  We type text

  We issue commands with voice

  CLs are natural – typing commands is not

Selecting Events

- Alan Cooper expects a pen input to be used as a pointer, but it is much harder to pick up a pen than grasp a mouse.
  - This is why pens are no longer widely used
- Keep events close together!

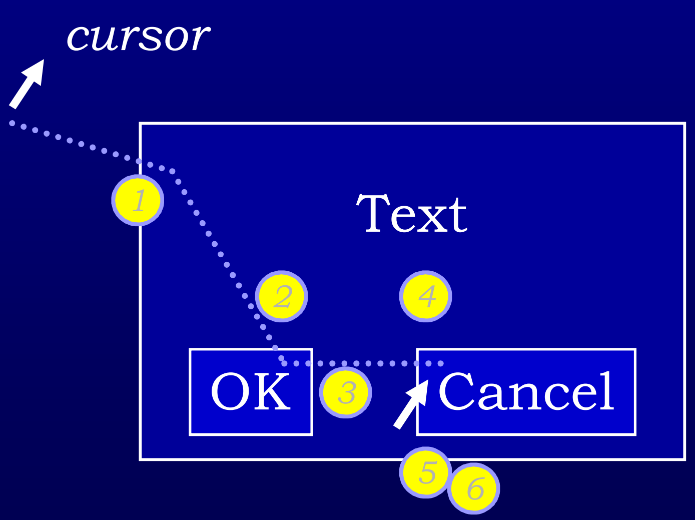


Direct Manipulation

- Direct Manipulation is about *visual feedback*, not just about *using a mouse*.
- Cooper : "*I've seen many instances of direct-manipulation idioms implemented without adequate visual feedback, and these idioms fail to satisfy the definition of effective direct manipulation*."
- That's what I mean by "*a menu system with a mouse*."
  - Like Patriot web

GUI Events

Events are state changes generated when the interface is manipulated (usually the cursor is moved or the mouse is clicked)



Mouse Events

1. Into text box
2. Into OK button
3. Out of OK
4. Into Cancel button
5. Left mouse button down
6. Left mouse button up

GUI Events (2)

Speeding up a GUI is primarily about reducing the number of events needed to accomplish a task.

Levels of events

1. User level
   - Selecting
   - Dragging
   - Resizing
2. GUI level
   - Crossing a widget boundary
   - Button down
   - Button up

GUI Events (3)

- These are languages with which we manipulate GUIs. The user level language is built on top of the GUI level.
- When we build a GUI, we are using the user level language to build an application-level language.

GUI tools provide a user level language, but also restrict the design!

User Level Language Hints

- Provide a way to escape from dragging
  - escape
  - any keystroke
  - chord-click-click right button to cancel

    MS API does not define chord-click

    Programmers must go to the GUI level
- Try to avoid mixing fine motor control (clicking) with gross motor (dragging) as much as possible
- Provide visual feedback when grasping (selecting)
- Apply the principle of locality

  Put widgets that users will use adjacently in time adjacent in space

Dialog Boxes

- Dialog boxes are the most inconveniently designed part of most GUIs. Dialog boxes are sometimes:
  - Unnecessary
  - In the wrong place
  - Labeled with confusing text
- Remember: You are designing a language
  - Unfortunately, you have been trained to expect languages to be poorly designed (C, C++, Windows, English), but you can do better!
  - You have the advantages of a class in HCI design ☺

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
  - Firefox uses keyboard accelerator: “/string”

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

Modeless Dialog Boxes

- The "*owning*" program continues
- Usually have terminating commands (*close*)
- Slightly unclear – when do they go away?
  – people expect them to be modal
- Example: Find (^F) in MS Office

Improving Modeless Dialog Boxes

1. Make modeless boxes look different
   - Color
   - Pattern
   - Border

   highlight
2. Use consistent terminating commands for modeless boxes
3. Make modeless boxes very different from modal boxes (for example, a floating tool bar)

   This example doesn't always work, but the concept might …

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

Four Kinds of Dialog Boxes (2)

3. Bulletin dialog box

   Gives the user some feedback

   - error message
   - confirmation messages

   Note that the user does not request these!

4. Process dialog box

   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*

GUI Menus: Tool Bars

A tool bar is a menu of buttons

- Permanent menu
- Designed for knowledgeable users
- Use to provide fast access to commonly used functions

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 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."

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

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

Summary

If you want to examine the impact that Alan Cooper has had on GUI design, compare MS Office 2000 products with previous MS Office versions.