GUI Tips

From GUI Bloopers, Jeff Johnson

GUI Blooper

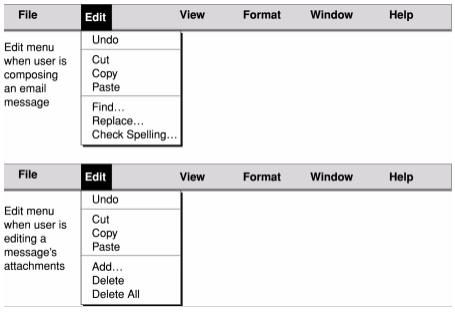
- List of common mistakes that software developers frequently make when designing Graphical User Interfaces (GUI)
- Categories we will discuss:
  - GUI Components
  - Layout and Appearance
  - Textual
  - Interaction
  - Responsiveness

Review of Design Principles

- Focus on the users and tasks, not technology
- Consider function first, presentation later
- Conform to the user's view of the task
- Don't complicate the user's task
- Be consistent
- Deliver information, not just data
- Design for responsiveness
- Try it out on users, then fix it!

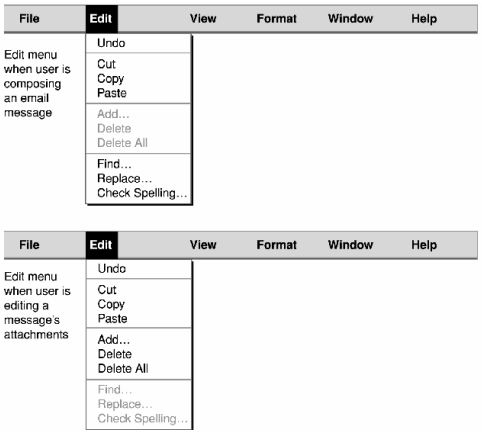
GUI Component Blooper: Dynamic Menus

- Menu item that changes depending upon the context
  - Might seem to help; removes commands one shouldn't be able to execute at that time
  - But users end up wondering where commands went



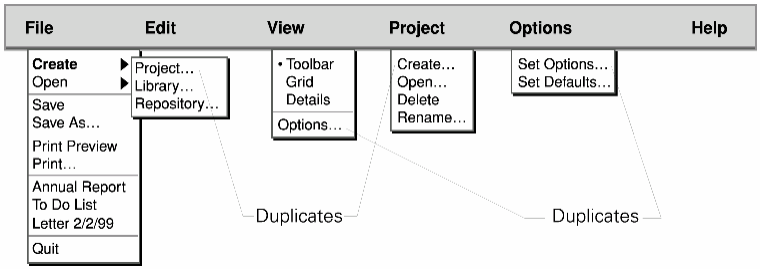
Dynamic Menus: Better

- Gray out or add entirely new menu that appears/disappears



GUI Component Blooper: Duplicate Menu Items

- Where to put an item?
- Better chance of finding if it's in two places?

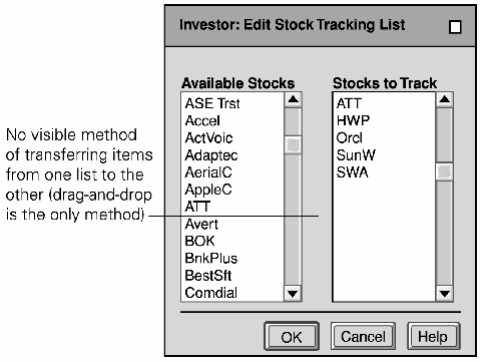


But users wonder if they are different or assume that they are

Even worse: multiple menu items with different names

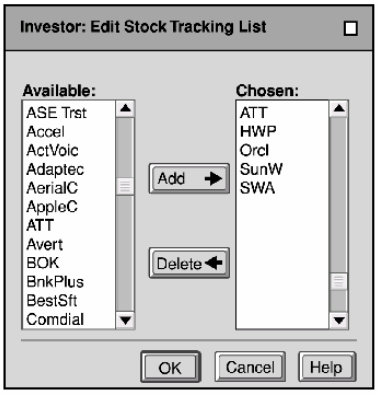
GUI Component Blooper: Hidden Functions

- Design might allow users to drag/drop between scrolling lists, use control key to delete



Better: Hidden Functions

- Hidden functions OK as long as it's not the only way

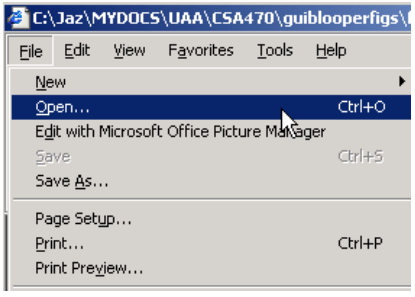


Hidden Functions

- (Weak) Reasons
  - Design software to resemble software they already know (e.g. vi)
  - Lack of space
  - Belief that they are faster ways than visible interfaces

GUI Component Blooper: No Keyboard Accelerators

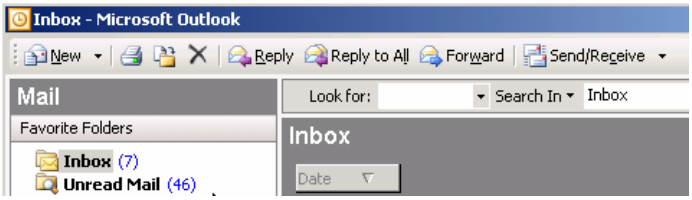
- Keyboard shortcuts in case:
  - Blind, Mouse inoperable, Save time



  - Avoid same shortcut key that does different things in different contexts (Ctrl-F / Ctrl-S)

GUI Component Blooper: Commands only on toolbar buttons

- What do the icons mean?
- Toolbar should be shortcuts for menu items



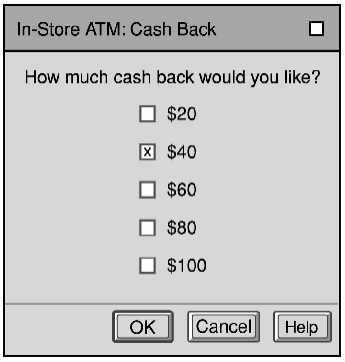
- Opposite problem:
  - ALL commands on the toolbar

Toolbars

- Primary windows should always include a menu bar
  - All top level commands organized into categories
  - Provide keyboard shortcut to invoke
- Toolbar optional
  - Should contain commands users will use frequently
  - Ideally customizable by the user

GUI Component Blooper: Confusing checkboxes and radio buttons

- Radio buttons when only one is selectable
- Checkboxes when many selectable



Word: Change font, effects, subscript and superscript

GUI Component Blooper: No Initial Values

- Always provide initial values when possible or logical

Some programmers do this intentionally as a way to allow users to not specify a value, but it violates the expectations about how a radio button operates

Instead give a “No Opinion” or “No answer” option

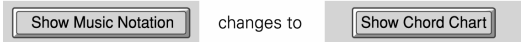
GUI Component Blooper: Checkboxes for non-ON/OFF Setting

- Checkboxes should be used for on/off not for a selection of items.

- Instead use radio buttons.

GUI Component Blooper: Command Buttons as Toggles

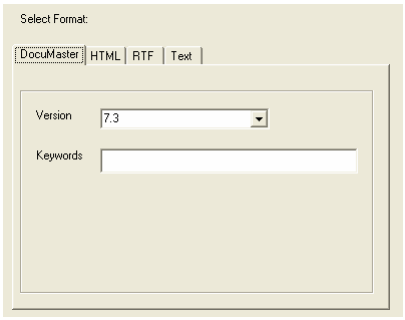
- Saves space on the screen but toggling meaning of a button can be missed by the user



- Use two buttons and disable the inactive one, or use a toggle switch style control

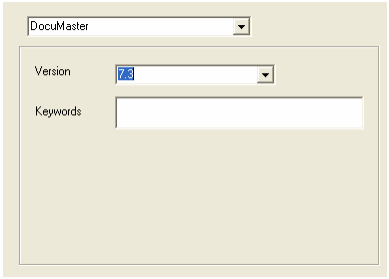
GUI Component Blooper: Using tabs as radio buttons

- Misuse of tabs is to use them as if they are radio buttons to present choices that affect what the application will do rather than just which controls are displayed
- Some users will not realize the last tab selected is the one that is used



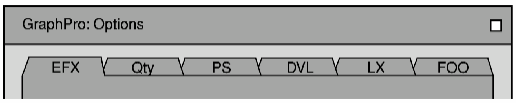
Tabs as Radio Buttons

- Better design:



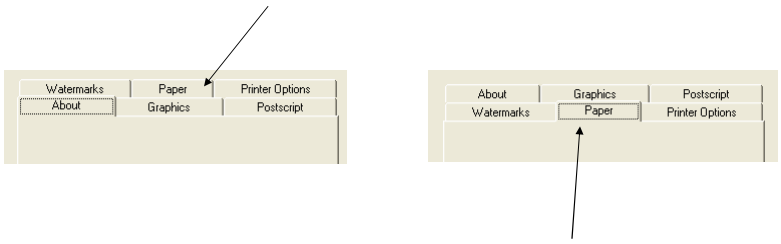
GUI Component Blooper: Too Many Tabs

- Intended to save space but too many uses more space – usually doesn't scale beyond a handful

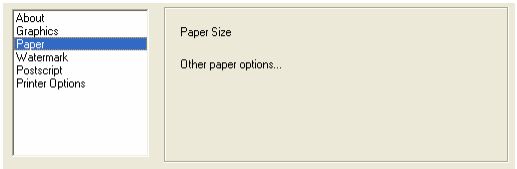


- Never use dancing tabs; change position based upon which tab is selected
  - Unavoidable with multi-rows of tabs

Multi-Row Tabs

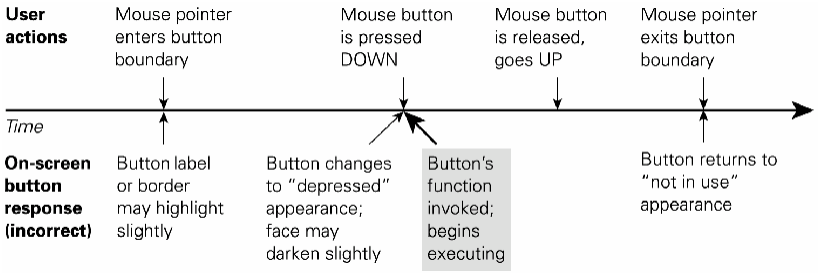


Solutions: Widen panel, make tabs narrower, or use another control instead of tabs



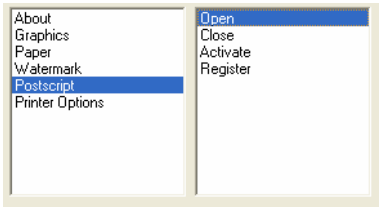
GUI Component Blooper: Buttons Triggered on Mouse Down

- Button should not trigger until mouse-up to give the user an option to abort



GUI Component Blooper: Ambiguous Selections

- If your app allows the selection of multiple items in different controls, make it clear which one is "the" selection



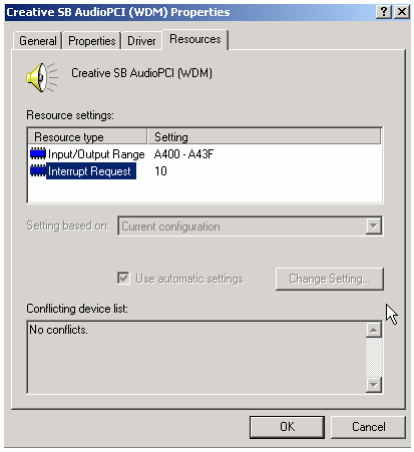
If press delete, which selection does it apply to?

Multiple selection in the same control is fine; operation applies to all selected items

GUI Component Blooper: Abusing Text Fields

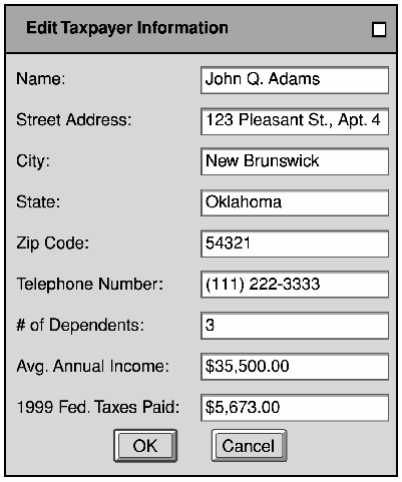
- Using text fields for read-only data
- Use label instead

- Worse:
  - Read-only but selectable, reader tries to change it



GUI Component Blooper: Overusing Text Fields

- Common when converting a command line or paper form



Take advantage of the GUI

Instead of:

Birthdate: 4/21/52

Telephone Number: (415) 555-1212

Use:

Birthdate: 4 / 21 / 52

Telephone Number: (415) 555 - 1212

Instead of:

Appt. Time: 11:30 am

Use:

Appt. Time: 11 : 30 am pm

or

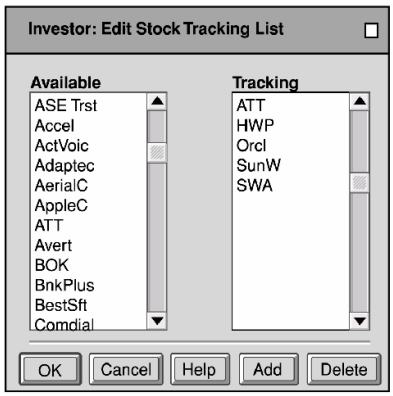
Appt. Time: 11 : 30 am pm

State: Oklahoma

State: Oklahoma

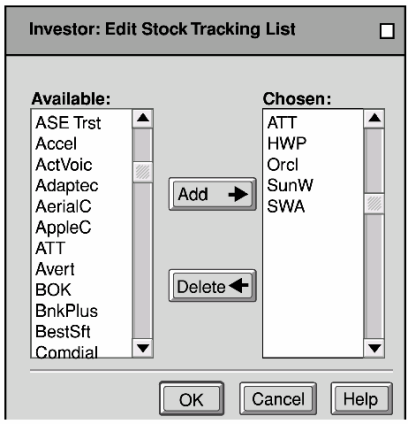
Layout Blooper: Mixing dialog box buttons with content control buttons

- Last time we talked about hidden controls
- Everything OK here?



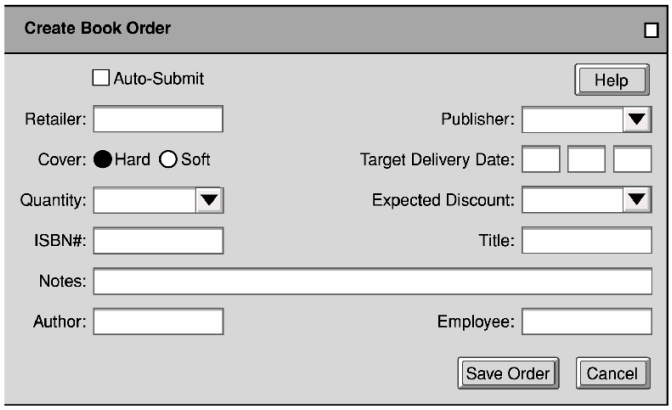
Align Buttons To Controls

- Make function clear



Layout Blooper: Un-Natural Order

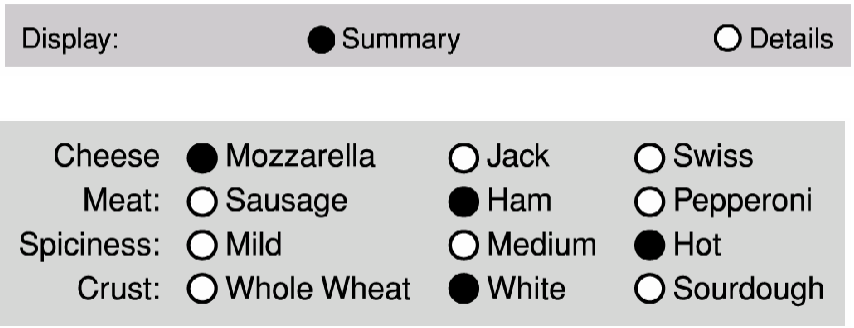
- Avoid the “random” layout



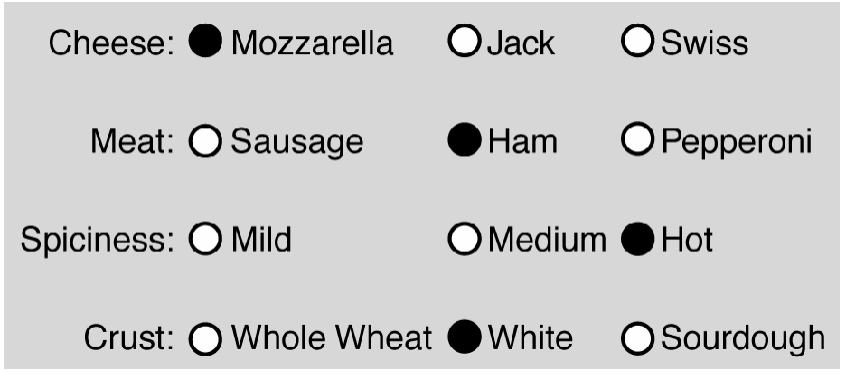
Add proper tab stops, but also reorganize layout

Layout Blooper: Radio Buttons too far apart

- Related radio buttons should be grouped closely together



Improved Spacing



Layout Blooper: Shoddy Labeling

- Alignment of labels and textboxes, use of :

Layout Blooper: Inconsistent or Tiny Fonts

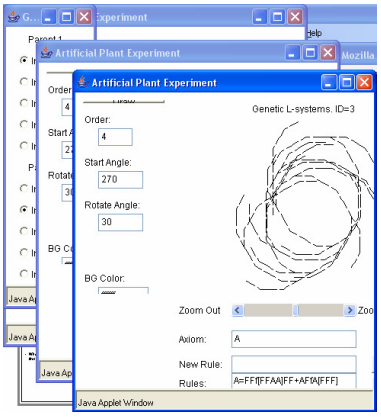
Excuses for Tiny Fonts

- I can read it. What's the problem?
- Hey, we gotta fit all this info in somehow.
- I just used the default font.
- It's not my fault, the text is in the image.
- It's big enough in low resolution.

- Minimum font size is 10, but 12 better
- Design for high resolution displays
- Let users adjust the font size

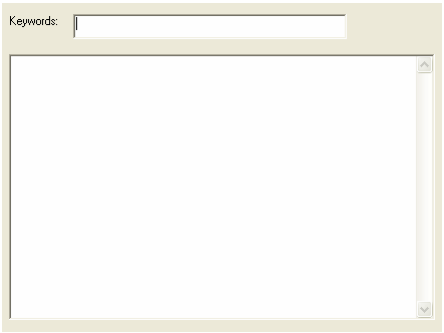
Layout Blooper : Poor Window Location

- Where should an application's windows first appear?
- Heuristics:
  - On-screen
  - Staggered
  - No occlusion



Layout Blooper: Unlabeled Scrolling Container Components

- It is common for GUI programmers to label small components (e.g. textboxes) but leave large scrolling components unlabeled
  - Perhaps the content explains its purpose? Not always the case



Text Blooper: Inconsistent Terminology

- Easy to be inconsistent in providing documentation regarding terms
  - Folder, Directory
  - Properties, Attributes, Parameters, Settings
  - Find, Search, Query
  - Arguments, Parameters
  - Login, Authenticate
  - Server, Service
  - ID, Key
  - Pounds, Kilograms

Text Blooper: Geek Speak

- Easy to allow programmer jargon to seep into the end product (assuming non-tech end user)
  - Error while checking mail

    TCP/IP Error 706; {37:1253}
  - Interface Hall of Shame:
    - A caller to Compuserve customer support said that even though he did what the software told him to do, it didn't work.

Text Blooper: Poor Writing

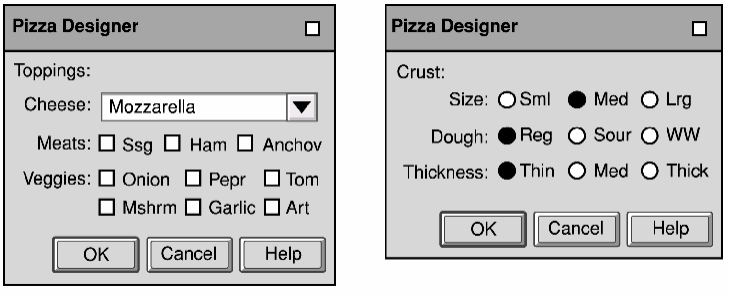
- Actual examples:
  - Link label: “Information before Getting Started”
  - “Connect is your contact to CompanyX technical support.”
  - “Can not connect with host server exitting.”

Text Blooper: Clueless Error Messages

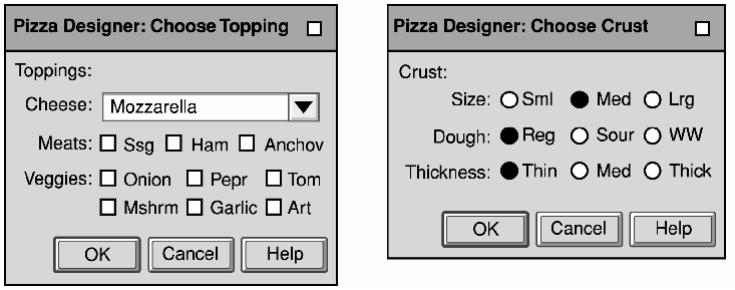
- Some examples:
  - “Nesting level too dip.”
    - Burned into ROM and shipped tens of thousands
  - “Error 500 HTTP Web Server”
  - “Excuse me, but Eudora could use some help.”
  - “File missing or you don’t have access.”
  - “Name contains invalid characters.”
  - “Value of field exceeds limit.”

Text Blooper: Misleading Window Titles

- No distinguishing name in MDI Form
- Can mislead users as to “where they are”, which function they are using

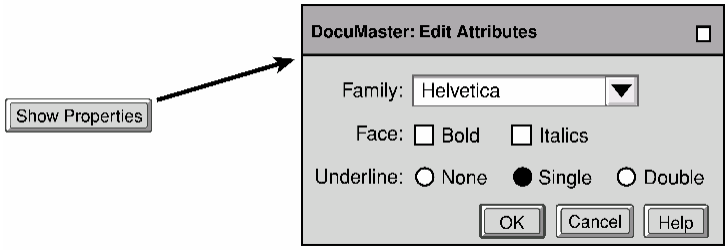


Unique Window Names



Text Blooper: Window Title Doesn't Match Invoking Command

- Mismatched titles can mislead user into thinking they selected the wrong command



Interaction Blooper: Exposing implementation to users

- Users should not be subjected to internal implementation details when they are contrary to their working model
- Examples:
  - Speed in a game a setting from 1 to 10
    - Expect 10 to be fast and 1 to be slow, but it was the opposite
    - Delay loop for the setting's number of times
  - Limits on data sizes to "weird" numbers
    - 16, 32, 64, 128, etc.
    - Most people would prefer 10, 100, 1000, etc.
- Design for the convenience of users, not developers

Interaction Blooper: TTY GUIs

- Conversion of old prompt-based UI to GUI:

Old prompt-based UI

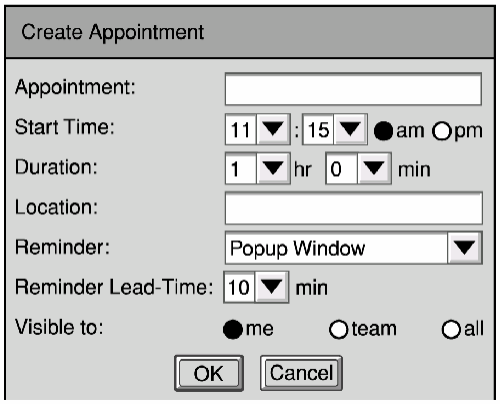
% Create Appointment	*User invokes function.*
Enter appt. name >	
Enter starting time >	*Prompts appear one by one after user types response to previous prompt and presses RETURN.*
Enter duration >	
Enter reminder >	
Enter reminder lead-time >	
Enter visible to >	
Appointment created.	*Function signals that it's done.*
%	

New TTY GUI

Avoid TTY-GUI

- Take advantage of GUI components to validate, constrain:

Users should also be able to create appointments by dragging and dropping email messages or attachments that describe appointments into their calendar.

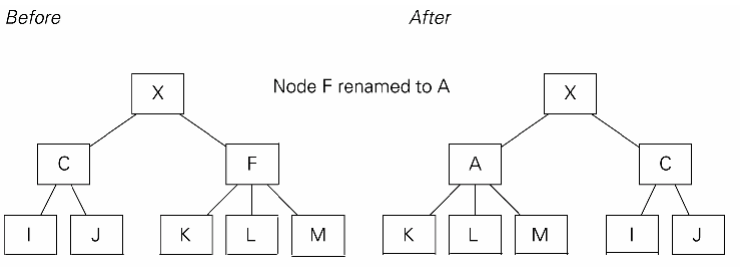


Interaction Bloopers

- Overwhelming users with decisions and details
  - Too many choices, many the user doesn't care about
  - Trying to fit too much into one screen can make the application more complex
  - Chunk choices into categories or different screens
  - http://holmes-iv.engr.uaa.alaska.edu/ncar/tdf14wizard.htm
- Easily missed information
  - Often status indicators, mode indicators, prompts for input, results, status messages, can be missed
  - May not be where the user is looking
  - Make these items stand out if important

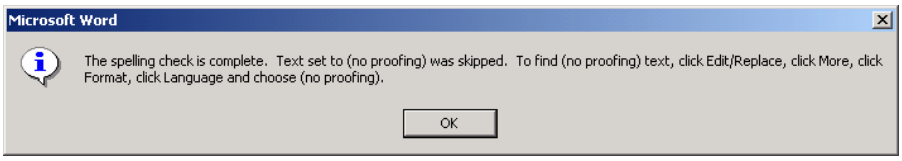
Interaction Blooper: Unexpected Rearrangement of Display

- What if the OS constantly rearranged your icons for you?
- Software sometimes rearranges when the user does not expect it



Interaction Blooper: Mission Impossible Instructions

- Instructions that go away too soon



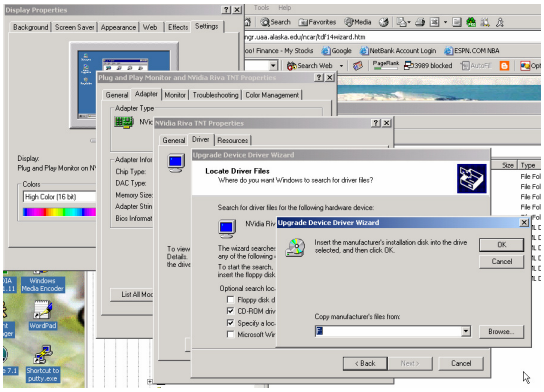
- Detailed instructions should remain on the screen while the user is carrying them out
  - Latest Office apps display help in right hand pane

Interaction Blooper: Installation Nightmares

- Many software products are a nightmare to install and set up, even if it is otherwise useful and usable
- gForge installation

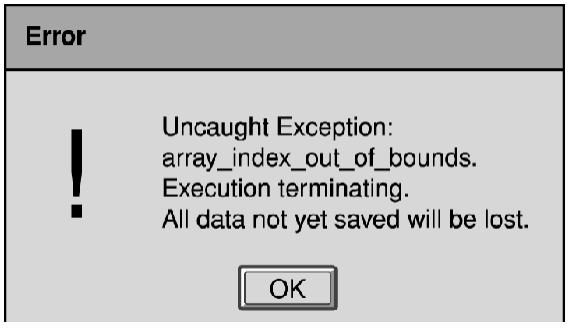
Interaction Blooper: Diabolical Dialog Boxes

- Too Many Levels
  - Deep hierarchies divert users from original goals, lose track of which OK, Apply, Cancel buttons are before them
  - Most people lost track more than a few levels down a hierarchy



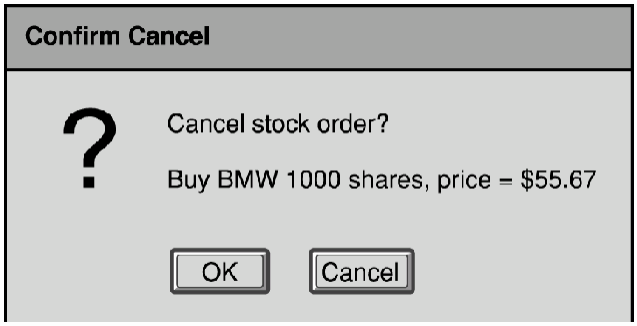
Interaction Blooper: Dialog Boxes Traps

- Dialog boxes that provide no way out other than the way the user doesn't want to go



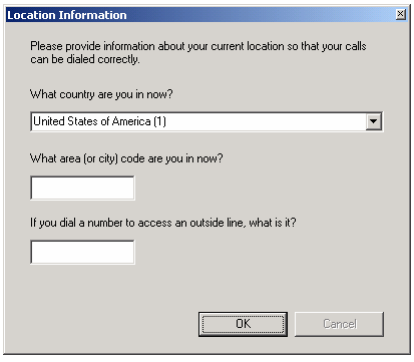
Dialog Box Traps

- Unclear meaning behind cancel



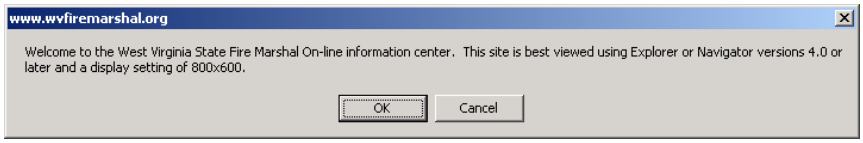
Dialog Box Traps

- Forced to enter data! Cancel disabled?



Interaction Blooper: Ok and Cancel do the same thing

- OK should mean “Yes do this” and Cancel should mean “No, I don’t want to do this”



Responsiveness Blooper: Waiting

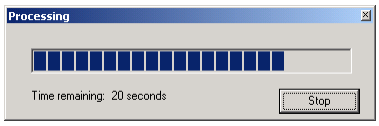
- Let the user know when the system is busy

- Without it the user may think the application is locked up, submit data multiple times, kill it

Responsiveness Blooper: Phony Progress Bars

- Progress bars should actually be accurate

- Better



- Beware of fancy animations that don't actually display progress
  - Might as well just give an hourglass

Responsiveness Tips

- UI designers rarely consider responsiveness during design
- Programmers equate responsiveness with performance
  - e.g. better algorithms or data structures
- A system can still be responsive but have "low" performance
  - Respond to user input quickly, but may process in the background
  - Respond to user input immediately but put job in queue for later

Real Time Interface

- 0.1 seconds
  - Limit for perception of cause-and-effect between events
  - Software that waits longer than 0.1 seconds to register a reaction to a user action appears “broken”
  - Limit for perception of smooth animation
- 1 second
  - Maximum comfortable gap in a conversation
  - If displaying information on the screen the user is unlikely to react until at least one second
- 10 seconds
  - Unit of time into which people break down their planning and execution of larger tasks
  - Every ten seconds user like to look up and reassess their task status, relax, etc.
  - Like to mark a task complete and move onto the next one
  - Amount of time a user is willing to spend to set up and operation and start it before losing patience (operation can take longer)

UI Testing

Even if you commit all of these bloopers, UI Testing could still save you.