CS 349
17 Interaction Guidelines

Byron Weber Becker
Spring 2009

Slides mostly by Michael Terry

University of Waterloo

13-July-09 Announcements

- Matt is gone this week and won't have office hours.
- Handing back A3 is delayed due to Matt's conference.
- Wednesday's lecture is cancelled (we had it May 8).
- Agenda:
  - UI Video: Side Views
  - User Interface Guidelines
  - Models of Interaction

University of Waterloo

UI Guidelines/Principles

- Set of “rules of thumb” to guide the design of the user interface
  - Don’t tell you *what* you should build
  - But provide guidance when designing application after you know what should be built
- Run spectrum from explicit rules to higher-level principles for overall design

University of Waterloo

Example Guidelines

- GNOME Human Interface Guidelines (HIG)
  - http://library.gnome.org/devel/hig-book/stable/
  - “This document tells you how to create applications that look right, behave properly, and fit into the GNOME user interface as a whole.”
- Examples:
  - “A primary window usually presents a view of the user's data… A primary window is always shown on the panel window list. A primary application window normally has a border, a menubar and a statusbar, and may also contain one or more toolbars.”
  - “Use *Filename* as the window title for document-based applications. Do not use the full pathname, as the filename alone is easier to distinguish amongst other open window titles, for example on the window list… Do not place version numbers, company names, or other information that is of no immediate use to the user in the window title.”
  - Label all buttons with imperative verbs, using header capitalization. For example, **Save**, **Sort** or **Update Now**.

Example Guidelines

- Apple Human Interface Guidelines
  - http://developer.apple.com/documentation/UserExperience/Conceptual/AppleHIGuidelines/XHIGIntro/XHIGIntro.html
- Examples:
  - "The Clear key has the same effect as the Delete command in the Edit menu: It removes the selection without putting it on the Clipboard. Not all keyboards have a Clear key, so don't require its use in your application."
  - Every document and application window and panel should have, at a minimum:
    - A title bar. Even if a window does not have an actual title (a tools panel, for example), it should have a title bar so that users can move the window.
    - A close button, so that users have a consistent way to dismiss the window.

What Guidelines Provide

- Explicit, well-defined rules to follow
  - Acceptable font sizes and styles
  - Minimum spacings required between controls, borders of panels
  - Color schemes to avoid for sake of color blind users
  - *In many cases, these rules can be automatically checked by compliance software*
- Higher-level principles
  - Consistency
  - Use of metaphors
  - WYSIWYG…
  - These principles typically *cannot* be automatically checked

University of Waterloo

Original Mac Design Guidelines

- Use metaphors
- Direct manipulation
- “See and point”
- Consistency
- WYSIWYG
- User in control
- Feedback
- Forgiveness
- Aesthetic integrity
- Modelessness…

University of Waterloo

Questioning the Guidelines

- "The Anti-Mac Interface," Gentner and Nielson (*Communications of the ACM*, 1996)
- Don Gentner
  - Was HCI engineer at Sun Microsystems
- Jakob Nielsen
  - Usability guru
  - Writes regular column on usability: http://www.useit.com

University of Waterloo

Anti-Mac Interface Article

- Paper is now 13 years old
  - But still relevant today…
- "At recent user interface conferences, several speakers have lamented that the human interface is stuck"
- "Many recent interfaces have tried to overcome the limitations of the desktop metaphor by extending it to some other room or building metaphor… These 3D designs try to emulate virtual reality on a flat screen but often seem to introduce a level of clunky indirectness in achieving common user goals. They are navigationally cumbersome…"

Why Question the Guidelines?

- Users' *needs* are what should drive design of application and its functionality, not guidelines
- Need to recognize how guidelines and tool set color our perceptions of what types of interfaces are possible
- Need to understand full design space

Macintosh Guidelines	Anti-Mac Guidelines
Metaphors	Reality
Direct Manipulation	Delegation
See and point	Describe and command
Consistency	Diversity
WYSIWYG	Represent meaning
User is in control	Shared control
Feedback	System handles details
Forgiveness	Model user actions (learn)
Aesthetic integrity	Graphic variety
Modelessness	Richer cues

Metaphors

- Guideline: Rely on real-world phenomena to suggest how computational artifact is used
- Three issues with metaphors:
  1. Target domain can have features not in source domain
  2. Source domain can have features not in target domain
  3. Some features in both domains, but qualitatively different
- Examples
  - Typewriter vs. word processor
  - Trash can
  - Phelps tractor steering
- Do metaphors change? How many present have used a typewriter? When do (did?) we start saying "a typewriter is like a word processor except…"?

University of Waterloo

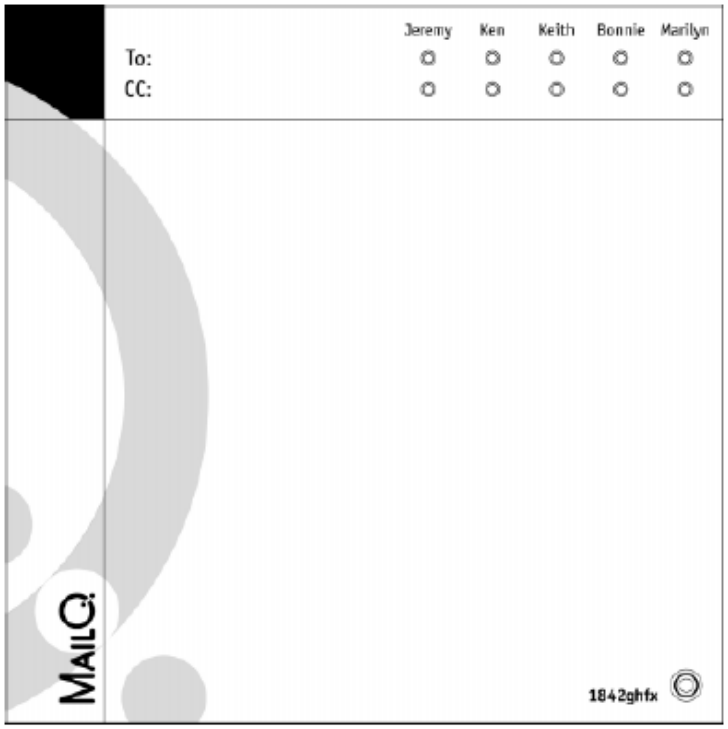
Rethinking Metaphors

- "Anti-Metaphor": Consider "reality"
- Amplify strengths of digital, physical media so they work in concert with one another
  - Don't blindly create high fidelity facsimile of real-world in software interface
  - Don't dismiss strengths of physical media
- Example:
  - Paper PDA

University of Waterloo

Paper PDA

Heiner, Hudson, Tanaka (UIST, 1999)



Direct Manipulation/See and Point

- *By Definition*: User composes commands by directly interacting with visible objects of interest
- *By Example*:
  - dragging a document to the trash/recycle bin
  - changing the size of a triangle by interacting with a visual representation
  - inserting characters in a word processor
- *By Contrast*: one of several interaction styles
  - menu selection
  - form fill-in
  - command language
  - natural language
  - direct manipulation

University of Waterloo

Direct Manipulation/See and Point

- *By characteristics*: (from *User Interface Design & Evaluation*, p. 213-214)
  - There is a visible and continuous representation of the task objects and their actions. Consequently, there is little syntax to remember.
  - The task objects are manipulated by physical actions, such as clicking or dragging, rather than by entering complex syntax.
  - Operations are rapid, incremental, and reversible: their effects on task objects are immediately visible.
- *By benefits*:
  - While interacting with DM interfaces, users feel as if they are interacting with the domain rather than with the interface, so they focus on the task rather than on the technology. There is a feeling of direct involvement with a world of task objects rather than communication with an intermediary.

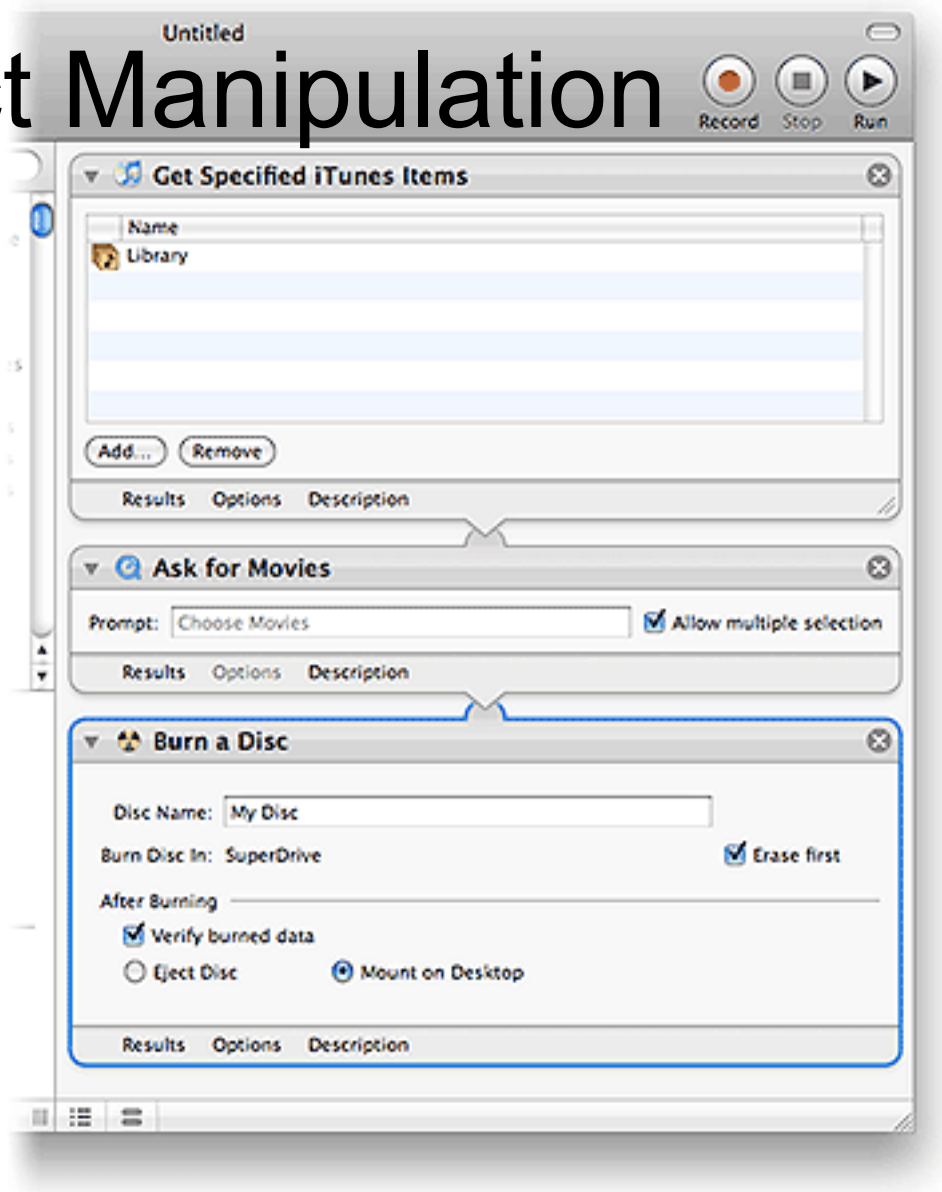
Direct Manipulation Issues

- Always working at an atomic level
- Tedious when working with many objects
- Cannot easily filter, select objects matching certain criteria
- Not amenable to scripting, automation
- Can lack precision in some tasks (e.g, drawing applications)
- Lacks expressiveness of language
- Can’t manipulate objects not present (e.g. set up future actions)

University of Waterloo

Rethinking Direct Manipulation

- "Anti-Direct Manipulation": Provide "levels" of interactive possibilities
  - Novices can focus on direct manipulation
  - Experts can harness power of language to script, automate interface
- Examples
  - Command-line interfaces
  - Scriptable applications
  - OS X's Automator: Bridges GUI and language



http://support.apple.com/kb/HT2488

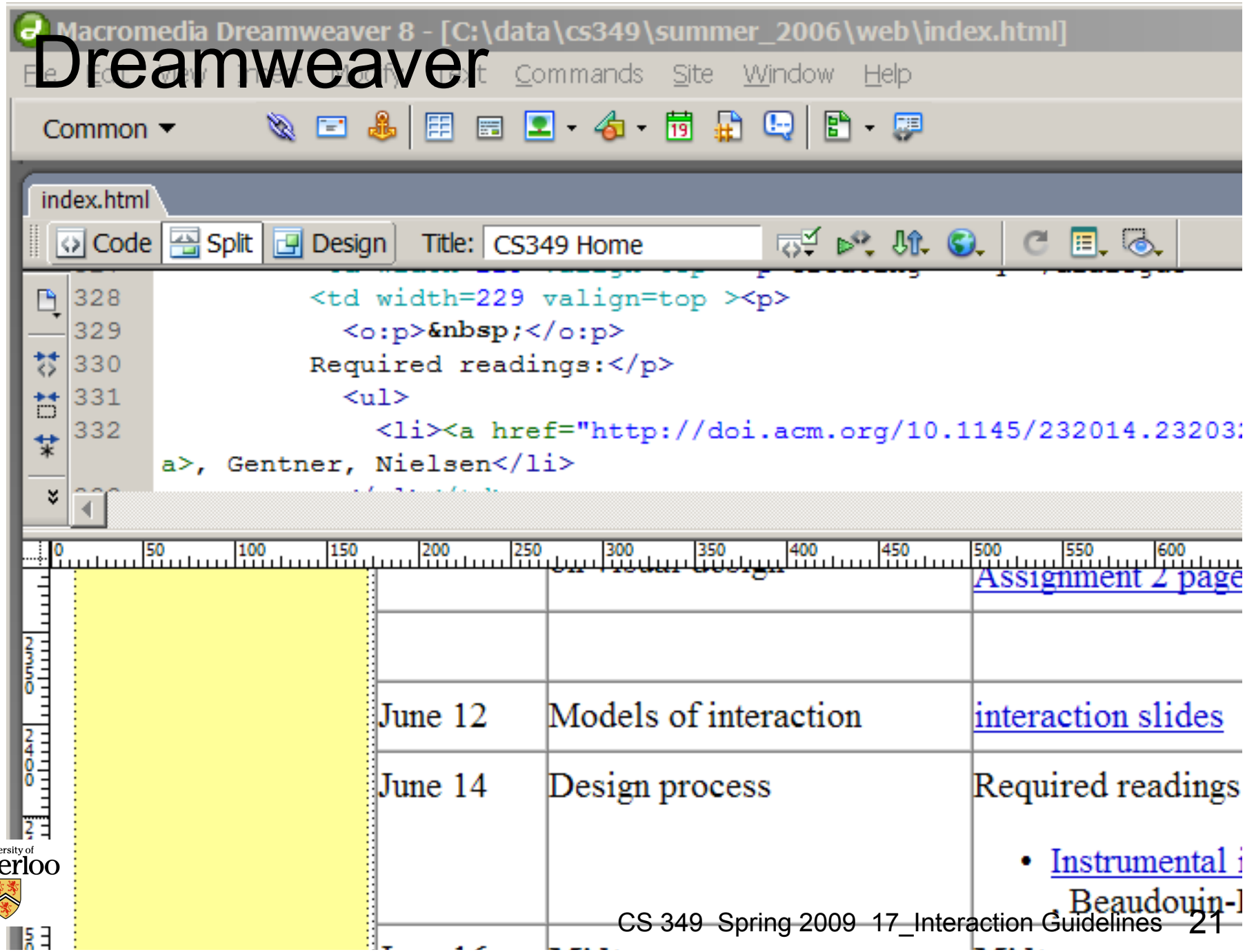
WYSIWYG

- Guideline: “What You See Is What You Get” to provide high-fidelity representation of document data, previews
- Issues:
  - Computational data is often multilayered, much richer than final representation
  - Often useful to be able to work with underlying representations rather than the “final result”

Rethinking WYSIWYG

- "Anti-WYSIWYG": Provide ability to interact with different representations of same data
- Examples
  - Dreamweaver
  - Inkscape
  - Side Views

Dreamweaver



Inkscape

- Demo…

Forgiveness

- Guideline: Ask for confirmation/permission before doing something radical
- Issues:
  - Can become tedious if continually ask user if it's OK to do something
- Resolution:
  - Maintain state information
  - Allow user to answer all future questions in same way
    - "Yes to "All"
    - "Don't ask me again"

University of Waterloo

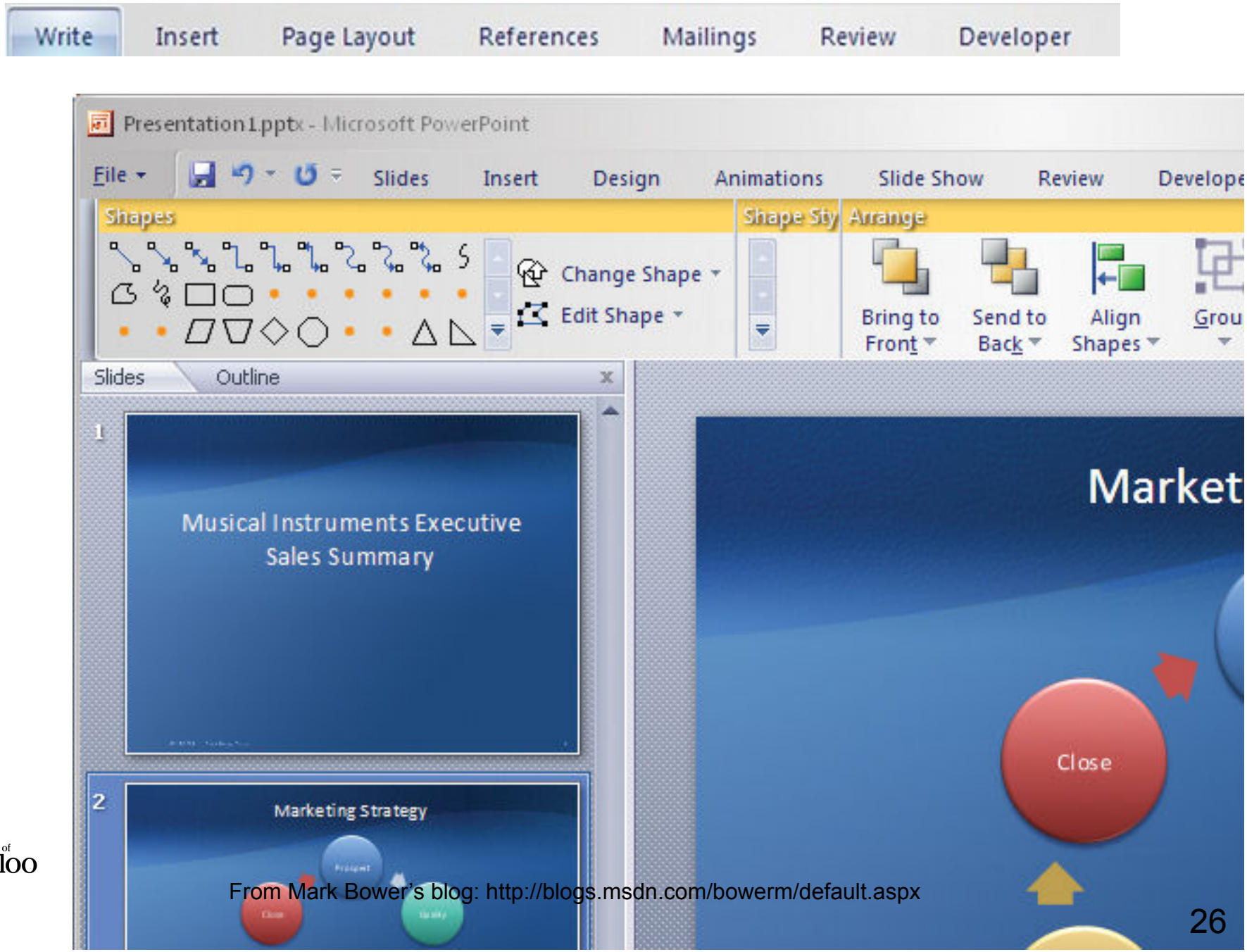
Modelessness

- Guideline: Don't impose modes on user, where possible
- Issues:
  - Modelessness can overwhelm users with choices
- Example:
  - Photoshop has 100s of commands available
  - All tools available, though only a handful applicable to task at hand
  - Why clutter the screen with them?

University of Waterloo

Rethinking Modelessness

- Moded interaction
  - Guide users through tasks by limiting what is visible, possible at any point in time
- Examples
  - “Wizards”
  - Microsoft’s “ribbons”



From Mark Bower's blog: http://blogs.msdn.com/bowerm/default.aspx

The Anti-Mac Interface

- The central role of language
- A richer internal representation of objects
- A more expressive interface
- Expert users
- Shared control

University of Waterloo

Summary

- Guidelines useful to structure interaction design
- But consider what they are trying to achieve
  - Understand implications and limitations
- Consider what user is trying to accomplish
  - Design to needs rather than to guidelines

University of Waterloo