Note 5: Software Posture

Yan Shi
Lecture Notes for SE 3330
UW-Platteville

Based on About Face 3: Chapter 9
Posture

- A product’s behavioral stance.
- The way a product presents itself to users.

- The look and behavior of a product should
  - reflect how it is used
  - result from a specific, goal-directed reason
What are the predominant features of these applications’ behavior stances?

What are the differences?

What are the predominant features of these applications’ behavior stances?

What are the differences?
Sovereign Posture

- Monopolize users’ attention for long periods of time.
  - Offer a large set of related functions
  - Occupying the full screen
  - Run continuously

- Example: Word, PowerPoint, Visual Studio
Sovereign Applications: Users

- Users are typically intermediates.
- Learning time is relatively short compared to eventual using time.
- Don’t sacrifice speed and power for ease of learning!
- Occasional Users cannot be ignored.

— WordStar vs. Microsoft Word
Be generous with screen real estate.

Optimize sovereign applications for full-screen use.
- Default to maximized presentation
- Need to be fully resizable
- Toolbars and their controls can be smaller than normal.
- Screen-splitters, rulers and scrollbars can be smaller and more closely spaced.
- *Do not* assume high resolutions
- Multiple toolbars/ribbons are fine
Sovereign Applications: Color

- Use minimal visual style.

- Users will stare at it for a long time, so
  - Keep the colors few and conservative.
Sovereign Applications: Feedback

- Great platforms for rich visual feedback.
- Status bar, title bar, status of data, etc.
- First time users won’t even notice them.
- But later they will be curious.
- Be careful: do not create a hopelessly cluttered interface.
Sovereign Applications: Input

- Great platforms for rich inputs.
- Direct manipulation, dialogs, shortcuts
- Assume fine controls:
  - users can be precise
- Put most frequently used items in the center.
- Use “dusty corners” for less-frequently used or dislocating items
Sovereign Applications: Document-Centric

- Maximize document views within sovereign applications.
- Texts, Slides, Spreadsheets, Code

How about scanner?

A product with a transient posture comes and goes:

- Present a single function
- With limited controls
- Often to support some sovereign applications

Usage model:

- Invoked when needed
- Performs its job
- Then quickly leaves

Example: calculator, volume control, math input panel, snipping tool
Transient Applications Should Be

- Simple!
- Clear!
- To the point!
Bright and Clear

- Colorful and Larger control
  - Bolder graphics help users to orient themselves more quickly.
  - Less likely to bother users because of the short appearance.

- Built-in instructions
  - A button “Set up Margins” is better than “Setup”.
  - Never use abbreviations.
Keep it Simple

- Transient applications should be limited to a single window and view.
- No tiny scrollbars and fussy mouse interactions.
- All important functions should be visible on the interface.
- Keyboard shortcuts are necessary, but should be simple.
- A title bar for dragging: movable
Remember user choices

- A transient application should launch to its previous position and configuration.

- Previous configuration is almost always more apt than any default setting.
Daemonic Posture

- Programs that *do not* normally interact with users.
  - Serve quietly and invisibly in the background
  - Perform possibly vital tasks
  - Without the need of human intervention
  - Usually manage processes.

- Example: printer driver, network connection.
Interaction with Daemonic Applications

- They must be adjusted occasionally.
- The interaction is transient in nature.

- Windows: represent daemons with onscreen application icon in the system tray.
  - Only for daemons with continuous, useful status information!
- Mac OS and Windows: control panels
  - Efficient approach to configure daemons.
Different Postures

- Same applications may have different postures, depending on users’ goal

- Example: calculator, PDF viewer

- First Step of Design: What posture is appropriate?
Web Postures

▪ Information oriented sites
  — must balance information density and ease of use for infrequent users
  — cannot assume screen resolution or size
  — between sovereign and transient

▪ Transactional sites and applications
  — online stores and financial services websites.
  — heavy informational aspect plus functionalities
    □ shopping cart, check out, user profile, ...
  — requires navigational clarity

▪ Web Portals
  — similar as desktop applications

What is Google’s posture?
Mobile Apps’ Posture

- Limited resources (screen, memory, power)
- Mostly transient posture
- Avoid using modes
- Balance navigation with display density
- Minimize input complexity
  - One hand or two hands?
- Integrate UI design with hardware design

Special Consideration for touch screen
Summary

- Software posture:
  - sovereign: run for long periods
  - transient: pop-ups
  - daemonic: background

- Web posture

- Mobile apps’ posture