

CSC420

Topic 2: Information Architecture and Application Structure

Information

- Lists of Objects (nouns)
- Lists of Actions (verbs)
- Lists of Categories (containers)
- Lists of Tools (separate programs)

Physical Structure

- Multiple windows
- Single swap-out window
- Tiled panes

Two-Panel Selector

- Two (or more) panels in the same window
- Click in one, show in the other
- Use when you want to show things (objects, categories, etc) that have a title and associated content
- Good because:
 - Reduces physical effort (everything is on screen, everything is close)
 - Reduces cognitive load (the screen contents are mostly stable)
 - Reduces memory burden (it's all on screen, just look)
- Examples

Canvas plus Palette

- Palette with icon-buttons next to an empty canvas
- Click on a button, then put on the canvas
- Good for graphical editors or, in general, anything that lets users create and arrange visual stuff
- Good because:
 - Is similar to the real world: take a tool from the toolbox, apply it on the empty canvas
 - Is familiar to users (long tradition of visual tools, reused icons, etc)
- Examples

One-Window Drilldown

- Every page with selections in a single window; when a selection is made, swap out the contents
- Good for applications with complex info structures and:
 - You have limited screen space
 - Or you have newbie users who cannot cope with multiple windows (and should not be distracted)
- Alternative to the higher-density patterns
- Examples

Alternative Views

- Allow users the option to switch to different *structural* views of the same content (as opposed to skins, which are just chrome)
- Good for formatted content, which will be seen by diverse people
- Good because:
 - Different goals (browse vs print)
 - User preferences
 - A need to triangulate (when dealing with creative users)
- Examples

Wizard

- Lead a user by the hand through a complex sequence
- Good for:
 - Long tasks that can be divided into steps
 - Downstream options depend on upstream choices
 - Newbie users dealing with unfamiliar tasks
 - Newbie users willing to surrender control
- Good because it reduces cognitive load (by assuming that users don't need to know the overall structure)
- Examples

Extras on Demand

- Show the most important content, and allow quick access to all the rest
- Good for cases when there is too much information to be shown, but some of that can be skipped
- Good because:
 - Lets users adjust the interface to their needs (simple for newbies, complex for experts)
 - Saves a lot of screen space
- Try to achieve the 80/20 rule: make 80% of the use-cases easy, and the rest at least possible
- Examples

Multi-Level Help

- Use multiple help components of various complexity
- Good for: complex applications with diverse users
- Good because:
 - People are different. Some want it quick and fast, others want to know and plan ahead
 - People are different. Some need an essay, others just a reference
 - Newbies vs experts
- Examples