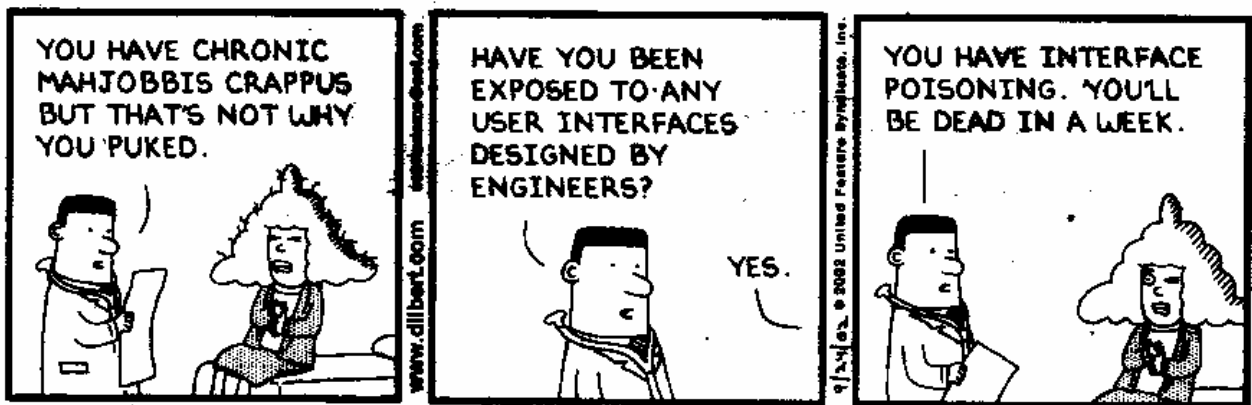
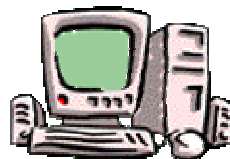


DESAIN



System-Centered Design

- Focus is on the technology
 - What can be built easily using the available tools on this particular platform?
 - What is interesting to me, as the developer, to build?



User-Centered Design

- Design is based on user's
 - Tasks
 - Abilities
 - Needs
 - Context
- Mantra: Know the user!



Design Process (1)

- How do we come up with new (good) designs for interactive systems?
- Briefly revisit Norman...

Good Design

- Invites person to use it properly
- *Visual affordance*
 - The perceived and actual fundamental properties of an object that help convey how it should be used -- (D. Norman)

Guidelines for Design

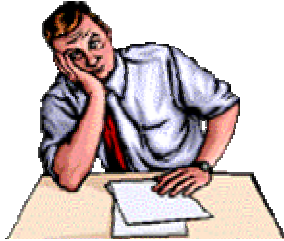
- 1. Provide a good conceptual model
 - User has mental model of how things work
 - Build design that allows user to predict effects of actions
- 2. Make things visible
 - Visible affordances, mappings, constraints
 - Remind person of what can be done and how to do it

Design Process (2)

Why is Design Difficult?

- 1. Increasing complexity/pressure
 - Number of things to control has risen dramatically
 - Display is increasingly symbolic/artificial
 - Feedback is more complex and subtle
 - Errors are increasingly serious/costly
- 2. Marketplace pressures
 - Time is money
 - Adding functionality (complexity) is now easy and cheap
 - Adding controls/feedback is expensive
 - Design usually requires several iterations before success
- 3. People often consider cost and appearance over human factors design
 - Style over substance
 - Bad design may not be visible

- 4. Creativity is challenging
 - Can't just make a copy
 - Want creativity, but want pragmatism



Idea Creation

How do we create and develop new interface ideas and designs?

- Ideas come from
 - Imagination
 - Analogy
 - Observation of current practice
 - Observation of current systems
- Borrow from other fields
 - Animation
 - Theatre
 - Information displays
 - Architecture
 - ...

Interface Metaphors

- *Metaphor* - Application of name or descriptive term to another object which is not literally applicable
 - Use: Natural transfer - apply existing knowledge to new, abstract tasks
 - Problem: May introduce incorrect mental model

Metaphor Creation (1)

- Prepare
 - What functions are needed?
 - What are users' problems?

Metaphor Creation (2)

- Generate
 - Use metaphor that matches users' conceptual tasks
 - Given choice, choose metaphor closest to way system really works
 - Ensure emotional tone is appropriate to users

Metaphor Creation (3)

- Evaluate
- Evolve

Idea Creation

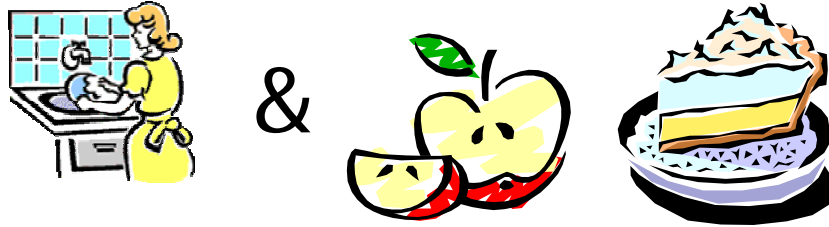
- Other methods for creating and developing interface ideas
 - ?

Idea Creation Methods

- 1. Consider new use for object
- 2. Adapt object to be like something else
- 3. Modify object for a new purpose
- 4. Magnify - add to object
- 5. Minimize - subtract from object
- 6. Substitute something similar
- 7. Rearrange aspects of object
- 8. Change the point of view
- 9. Combine data into an ensemble

Design Guidelines/Principles

- General guidelines (advice) to help create more usable systems



- Can be subtle, even contradictory

Design Principles

- 1. Use simple and natural dialog in user's language
 - Match user's task in a natural way
 - Avoid jargon, techno-speak

Insufficient funds to
withdraw \$100

VS.

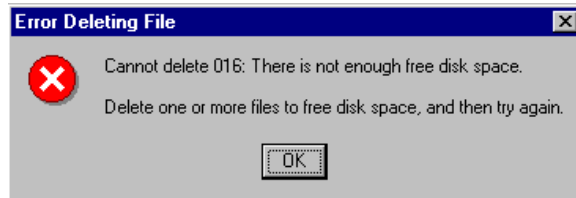
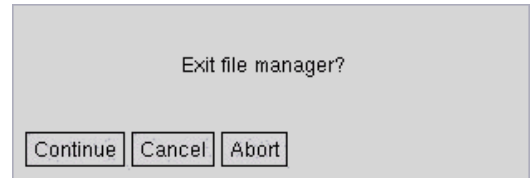
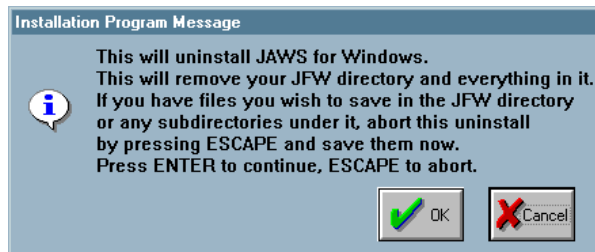
X.25 connection discarded due
to network congestion. Local
limits now in effect

- Present exactly info that user needs

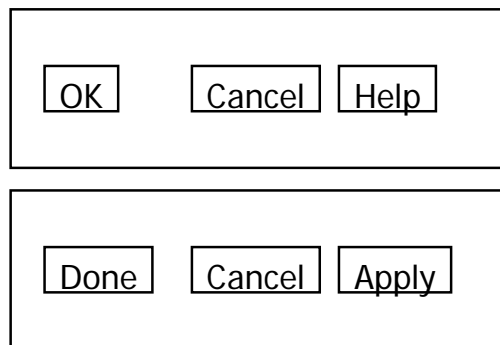
Less is more!

Fewer unnecessary windows, prompts, dialogs

Fun Examples



- 2. Strive for consistency
 - Sequences, actions, commands, layout, terminology
 - Makes more predictable



- 3. Provide informative feedback
 - Continuously inform user about what is occurring
 - Most important on frequent, substantive actions
 - How to deal with delays?
- 4. Minimize user's memory load
 - Recognition is better than recall
 - Describe required input format, include example and default
 - Date __ - ___ - __ (DD-Mmm-YY, e.g., 02-Aug-08)
 - Use small # of generally applicable commands

- 5. Permit easy reversal of actions
 - Undo!
 - Reduces anxiety, encourages experimentation
- 6. Provide clearly marked exits
 - Don't want the user to feel trapped
 - Examples

Cancel button on dialogs
 Interrupt/resume on lengthy operations (modeless)
 Quit - can exit anytime
 Reset/defaults - restore on a property sheet

- 7. Provide shortcuts
 - Enable frequent users to perform often-used operations quickly

Keyboard & mouse

- abbreviations
- menu shortcuts
- function keys
- command completion
- double click vs. menu selection

Navigation between windows/forms

Reuse

Provide history system

- 8. Support internal locus of control
 - Put user in charge, not computer
 - Can be major source of anxiety

Enter next command
 vs.
 Ready for next command

- 9. Handle errors smoothly and positively
- 10. Provide useful help and documentation