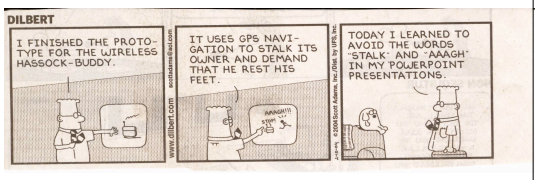
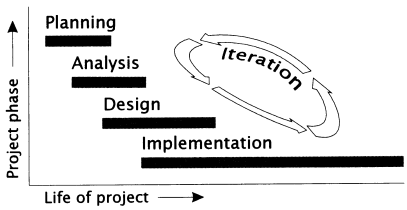
Lecture 8

Prototyping
Rosson & Carroll Chapter 6



Iterative Model of Development

- Describe steps in method



Compare to the Waterfall Model

Steps in method (note: iterative!)

- Planning
  - Scope of project, investigate user population (document analysis, interviews, surveys, observation) & related systems
- Analysis (R&C Requirements Analysis)
  - Task analysis, problem scenario development, requirements for usefulness and usability
- Design (R&C Activity, Information & Interaction Design)
  - Specifications (yes!) for the human-computer interaction (UI?)
- Implementation (R&C Prototyping)
  - Storyboards, mock-ups, software prototypes
- Usability Evaluation (R&C Evaluation)
  - Evaluation without users: cognitive walkthrough, guidelines, GOMS, Keystroke Level Model (KLM)
  - Evaluation with users (usability testing, interviews, questionnaires)

UI Prototypes

- Definition: A concrete but partial implementation of a system design built to explore usability issues.
- Why prototype?
  - Support creativity
    - Exploring the design space: generating alternatives
    - Contracting the design space: selecting alternatives
  - Encourage communication
  - Permit early evaluation of design
  - Cheap!

Dimensions of Prototypes

- Representation
  - Paper or computer product
- Precision
  - Level of detail (rough or highly polished)
- Interactivity
  - Can user actually interact with the prototype?
- Expected life cycle
  - Rapid (throw-away) vs. evolutionary

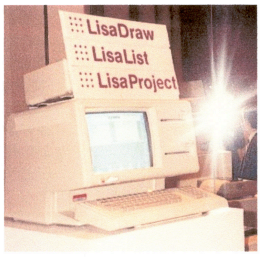
Types of UI Prototypes

- Rapid Prototypes
  - Non-computer
    - Paper sketch
    - Paper mockup
    - Storyboard
    - Video animation
  - Computer
    - Electronic mockup
    - Computer animation
    - Wizard of Oz
- Evolutionary software prototypes
  - Similar to Extreme Programming (Kent Beck)
  - UI Toolkits
  - UI Builders
  - UI Development Environments (UIDE)

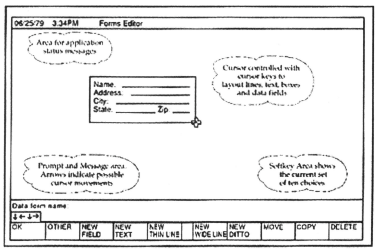
Types of UI Prototypes

- Rapid Prototypes
  - Computer
    - Electronic mockup (non-interactive)
      - Photoshop
    - Computer animation(non-interactive)
      - Macromedia Director
        - » http://www.macromedia.com/
    - Computer animation(interactive)
      - Macromedia Director
      - Scripting languages: Tcl/Tk (also for Python)
        - » www.scriptics.com
      - MacProto (Awe & Jessen)
    - Wizard of Oz
      - Workstation connected to invisible human assistant who simulates input, output or processing functionality not yet available

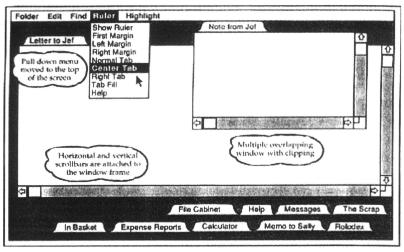
Apple's Lisa (1979-1983) First GUI personal computer



Lisa Display Sketch (June 1979)



Lisa Display Mockup (Standards document 1980)



Tool Command Language
TCL

- Scripting language for developing & using GUIs
- Allows generic programming
  - variables, loops, procedures
- Embeddable into an application
- Extensible
- Interpreter written in C called Wish
  - Advantages? Disadvantages?

Toolkit for Tcl

TK

- Cross-Platform UI Widgets
  - X Window, Microsoft Windows, Mac
- Can program widgets with Tcl scripts
- Written in C
- Extensible
  - new UI widgets
  - new geometry managers

Tcl/Tk Example

```
button.dialogbox.ok -text OK -command
   {destroy.dialogbox}
```

- Creates a button, called ".dialogbox.ok" with the label "OK". It deletes its parent window ".dialogbox" when the button is pressed.
- Traditional language would take 5 to 20 lines to create same button.

Tcl/Tk

Benefits

- Rapid development
  - interpreter *wish* (windowing shell)
  - higher level language than C, C++ or Motif Tk
    - 1/10 less time to code
    - easier to learn
- Can call Java or C programs
- Can "glue" together many library packages
- Convenience
  - cross-platform
  - shareware, freeware

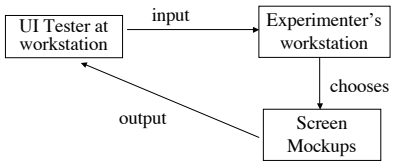
Tcl/Tk

Disadvantages

- Interpreter creates slow code
  - 8.0 has compiler
- Replace with Java?
  - probably not: Tcl/Tk is much faster to learn and code
- Text oriented
  - GuiBuilders available: SpecTcl (see /local/apps/tcltk/SpecTcl-1.1 directory and Visual Tcl

MacProto
(Awe & Jessen)

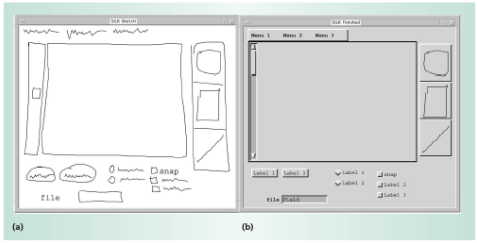
Wizard of Oz



Types of UI Prototypes

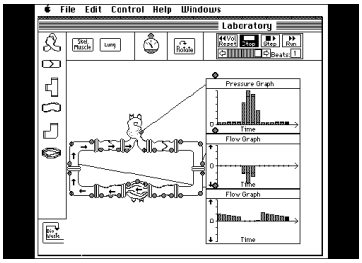
- Evolutionary software prototypes
  - Similar to Extreme Programming (Kent Beck)
  - UI Toolkits
    - Java Swing
    - Tk (under Tcl)
  - UI Builders
    - Direct manipulation environment with widget objects
      - Visual C++ Studio
      - Code Warrior
      - Visual Tcl/Tk
    - Visual Basic
  - UI Development Environments (UIDE)
    - SILK by James Landay

Landay's SILK from sketch to working MOTIF UI



Tcl/Tk and Visual Tcl/Tk Demo

Cardio-Vascular Construction Kit (CVCK) Demo



CVCK Prototype Examples Videotapes

- Paper mockup
- Rapid prototype in LISP
- Final usability testing on Mac in C