

## Rapid Prototyping

Dimensions and terminology

Non-computer methods

Computer methods

Poster session preview

Exam recap

## Design Artifacts

•How do we express early design ideas?

- No software coding at this stage

•Key notions

- Make it fast!!!
- Allow lots of flexibility for radically different designs
- Make it cheap
- Promote valuable feedback

**\*\*\* Facilitate iterative design and evaluation \*\*\***

## Dilemma

You can't evaluate design until it's built

**but...**

After building, changes to the design are difficult

**The solution ...**

Simulate the design, in low-cost manner

## Prototyping Dimensions

### 1. Representation

- How is the design depicted or represented?
- Can be just textual description or can be visuals and diagrams

### 2. Scope

- Is it just the interface (mock-up) or does it include some computational component?


### 3. Executability

- Can the prototype be "run"?
- If coding, there will be periods when it can't


### 4. Maturation

- What are the stages of the product as it comes along?
  - Revolutionary - Throw out old one
  - Evolutionary - Keep changing previous design

## Rapid Prototyping Methods

Non-computer (Typically earlier in process) → 

VS

computer-based (Typically later in process) → 

### Non-Computer Methods

- Goal: Want to express design ideas and get quick & cheap opinions on system
- Methods?

## Design Description

- Can simply have a textual description of a system design
  - Obvious weakness is that it's so far from eventual system
  - Doesn't do a good job representing visual aspects of interface

### Sketches, Mock-ups

- Paper-based "drawings" of interfaces
- Good for brainstorming
- Focuses people on high-level design notions
- Not so good for illustrating flow and the details
- Quick and cheap -> helpful feedback

## Storyboarding

- Pencil and paper simulation or walkthrough of system look and functionality
  - Use sequence of diagrams/drawings
  - Show key snap shots
  - Quick & easy



- Example



## Scenarios

- Hypothetical or fictional situations of use
  - Typically involving some person, event, situation and environment
  - Provide context of operation
  - Often in narrative form, but can also be sketches or even videos

### Scenario Utility

- Engaging and interesting
- Allows designer to look at problem from another person's point of view
- Facilitates feedback and opinions
- Can be very futuristic and creative

## Other Techniques

- Tutorials & Manuals
  - Maybe write them out ahead of time to flesh out functionality
  - Forces designer to be explicit about decisions
  - Putting it on paper is valuable

## Computer Methods

- Simulate more of system functionality
  - Usually just some features or aspects
  - Can focus on more of details
  - Typically engaging
  - Danger: Users are more reluctant to suggest changes once they see more realistic prototype

## Terminology

- **Horizontal prototype**
  - Very broad, does or shows much of the interface, but does this in a shallow manner
- **Vertical prototype**
  - Fewer features or aspects of the interface simulated, but done in great detail
- Early prototyping
- Late prototyping
- Low-fidelity prototype
- High-fidelity prototype

## Prototyping Tools

1. Draw/Paint programs
  - Ex. Photoshop, CorelDraw
2. Scripted simulations/slide shows
  - Ex. PowerPoint, Hypercard, Macromedia Director, HTML
3. Interface Builders
  - Ex. Visual Basic, Delphi, UIMX

### Prototyping Tools

#### 1. Draw/Paint programs

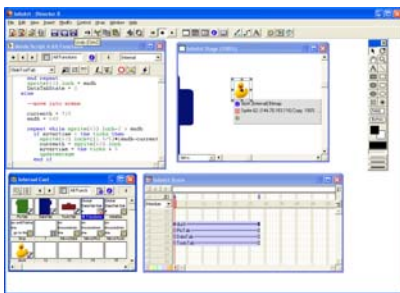
- Draw each screen, good for look
- Thin, Horizontal Prototype
- Adobe Photoshop



Prototyping Tools

## 2. Scripted Simulations / Slide show

- Put storyboard-like views down with (animated) transitions between them
- Can give user very specific script to follow
- Often called *chauffeured prototyping*
- **Macromedia Director**



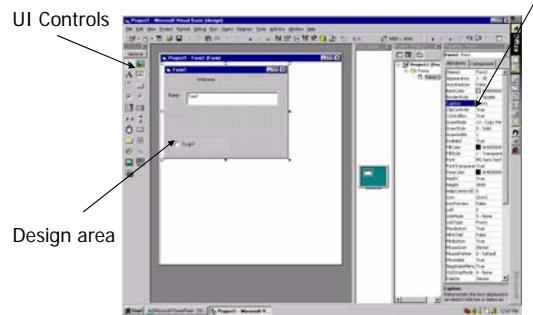
Rapid Prototyping

13/17

Prototyping Tools

## 3. Interface Builders

- Tools for laying out windows, controls, etc. of interface
  - Have build and test modes that are good for exhibiting look and feel
  - Generate code to which back-end functionality can be added through programming



Rapid Prototyping

14/17

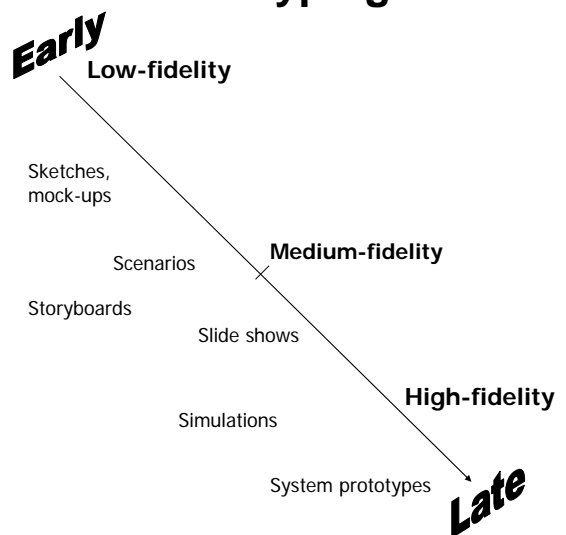
## Prototyping Tools

- Good features
  - Easy to develop & modify screens
  - Supports type of interface you are developing
  - Supports variety I/O devices
  - Easy to link screens and modify links
  - Allows calling external procedures & program
  - Allows importing text, graphics, other media
  - Easy to learn and use
  - Good support from vendor

Rapid Prototyping

15/17

## Prototyping



Rapid Prototyping

16/17

## Prototyping Technique

- **Wizard of Oz** - Person simulates and controls system from “behind the scenes”
  - Use mock interface and interact with users
  - Good for simulating system that would be difficult to build



Can be either computer-based or not