Rapid Prototyping

Prof. James A. Landay
University of Washington
Autumn 2006

November 16, 2006

Hall of Fame or Shame?

Interesting Survey Results...

In 1997, Harris Research found that employees who use PCs can lose up to 3 weeks/year tackling technology problems. Why?

• Learning/installing new systems
  – each employee lost 100 minutes a week in the 1st month a new system was introduced
• Dealing with PC/software malfunction or misuse

http://www.cs.washington.edu/education/courses/cse490f/CurrentQtr/readings_files/harris-research-survey.htm

Who Adapts to Who?

“It is far better to adapt the technology to the user than to force the user to adapt to the technology.”

Larry Marine, Intuitive Design

Rapid Prototyping

Prof. James A. Landay
University of Washington
Autumn 2006

November 16, 2006
Outline

• Review lo-fi prototyping
• Informal prototyping tools
• Why build hi-fi prototypes?
• Hi-fi prototyping tools
• Widgets
• What prototyping tools lack

Lo-fi Testing Review

• Low-fi testing allows us to quickly iterate. The advantage of this is?
  – get feedback from users & change right away
• What are the other advantages of only requiring the skills of “tiny fingers”?
  – non-programmers can fully participate

Problems with Low-fi Prototypes

• “Computer” inherently buggy
• Slow compared to real app
  – timings not accurate
• Hard to implement some functionality
  – pulldowns, feedback, drag, viz …
• Won’t look like final product
  – sometimes hard to recognize widgets
• End-users can’t use by themselves
  – not in context of user’s work environment

Problems with Low-fi Prototypes?

• Doesn’t map well to what will actual fit on the screen
• Couldn’t hold in your hand – different ergonomics from target device
• Timing in real-time hard to do (sloooow computer)
• Some things could not be simulated (highlighting)
• Writing on paper not the same as writing on target device
• Appearance unrealistic
• Dynamic widgets hard to simulate (pop-ups)
• Some items had to be static!
• Dragging hard to simulate

Problems with Low-fi Prototypes?

• Couldn’t measure realistic I/O
  – mouse (can’t sketch the same way)
  – slow response
• Lack of interactive feedback
  – button highlights
• “Computer” has to keep track of a lot of paper
• Hard to draw well (recognition of elements)
• Users wouldn’t criticize UI
• Can’t get accurate time measurement
• Couldn’t give proper affordance that something wasn’t selectable
Informal UI Prototyping Tools

- Support advantages of low-fi paper prototypes
  - brainstorming
  - consider different ideas rapidly
  - do not require specification of details
  - incomplete designs
  - need not cover all cases, just illustrate important examples

- Add advantages of electronic tools
  - evolve easily
  - support for “design memory”
  - transition to other electronic tools
  - allow end-user interaction
  - work with real devices

Designers’ Outpost:
A Tangible Interface for Designing Information Architectures

- Combines physical & virtual
  - physical post-its, virtual feedback

- Supports existing practice
  - affordances of paper
  - collaboration
  - large, persistent representation

- Adds advantages of e-media
  - editing, reuse, distribution
  - hand-off later to other tools

DENIM:
Designing Web Sites by Sketching

- Early-phase navigation & interaction design
- Integrates multiple views
  - site map – storyboard – page sketch

DENIM
Finding a Tighter Fit Between Tools and Practice for Web Site Design
James Lin, Mark W. Newman, Jason I. Hong, James A. Landay
Group for User Interface Research
University of California, Berkeley
June 22, 2000
Low-fi Prototyping & Testing

Travelshare

A Visual Language for Sketching Large and Complex Interactive Designs

James Lin
Michael Thomsen
James A. Landay

May 11, 2001

Denim Demo

SUEDE:
Informal Prototyping for Speech-based UIs

• Support design practice
  – example scripts
  – Wizard of Oz (WoZ)
  – built-in iterative design
    • design – test – analysis
• Fast & fluid design
  – no speech recognition or synthesis
  – need not be programmer

Administrivia
Midterm Survey Results

• Thank you for active participation
• Appreciate positive comments on what you value
  – Useful for future work & applying what you learn
  – Good examples, screen shots, interactive lectures
  – Teams & picking own projects
• Things to work on (for now & future)
  – Respect your time (start & stop on time)
  – Updating examples
  – Getting grades back quicker (more TA support next year)
    • Talk #1 grades back today
    • Kate will email low-fi grades/comments on Wed.
    • Midterm returned next Tue.
  – Feedback on work
    • Industry mentors & more in class work next week, OHs
    • More data behind conclusions in slides
**Administrivia**

- **Course schedule update**
  - pushed some things out & removed a lecture to add in class work
  - Tue., 11/28 work on project in class (bring laptops)
  - Thur. 11/30 Interactive Prototype #6 due
  - Tue. 12/5 in class work on HE summary
- **Final presentations & project fair**
  - Thur. 12/7 10:30 AM – 1 PM, location TBA
  - Must let us know if you have to leave at noon (so we schedule your team early)
  - Short poster session afterwards
  - Industry representatives will be there and I’ll supply snacks

---

**CSE490f Winter Quarter Overview**

- Continue where this quarter leaves off
  - pick top 3-4 projects from this course to continue
  - fill in holes in designs
  - high fidelity prototypes (w/ working code)
  - more user testing & iteration
- **Smaller course (10-15 students)**
- **Lecture/Studio hybrid**
  - post individual assignments on walls/PPT & do class critique
- **Explore advanced HCI methods & research**
  - e.g., video prototyping, etc.
- **Have add codes-- email me for one**

---

**TOPIARY:**

**Informal Prototyping for Location-enhanced UIs**

- Create location-based scenarios
  - place people, places, & things on map
- Use scenarios as conditions on storyboard transitions
- Iterative design
  - Wizard of Oz (WoZ)
  - Place Lab Wi-fi location sensor
- Fast & fluid design
  - no special hardware required
  - need not be programmer

---

**Activity Designer:**

**Informal Prototyping for Activity-based UIs**

- Create activity-based scenes
  - actions in a particular situation (e.g., running in the park at lunch)
  - visually create status properties & visual feedback
  - number of times someone ran
- Use scenes & properties as conditions on storyboard transitions
- Iterative design
  - Wizard of Oz (WoZ)
  - Test in field w/ actual devices
- Fast & fluid design
  - no special hardware required
  - need not be programmer
Augmented User Study/Design Tools

Visual Language for Property Computation

Testing Prototypes in Situ & with Wizard of Oz

DaPPL3: Tool for Designing & Prototyping Pervasive Computing Environments

- Create storyboards in designer’s language
- Extract & specify events & desired UI actions
- Build prototypes spanning lo-fi to medium-fi
- Test with Wizard of Oz or on real ubicomp technology
  - e.g., RFID readers, handheld devices, displays, etc.
Why Build Hi-fi Prototypes?

- Must test & observe ideas with customers
- Paper mock-ups don’t go far enough
  - how would you show a drag operation?
  - not realistic of how interface will be used
- Building final app. now is a mistake (?)
  - changes in the UI can cause huge code changes
    - need to convince programmer & hope they get it right
    - takes too much time
    - what did Cooper say it is harder than???
    - changing a 1000 square foot slab of concrete
  - drag & drop prototyping tool appropriate
  - but only after we have iterated on design
- Why is Cooper against prototyping?
  - sees prototyping as programming (above problems)
  - advocates paper (which I still consider prototyping)

Why Use Tools (rather than code)?

- Faster
- Easier to incorporate testing changes
- Multiple UIs for same application
- Consistent user interfaces
- Easier to involve variety of specialists
- Separation of UI code from app. code
  - easier to change and maintain
- More reliable
  - bugs found in the tool are fixed for all applications

Prototyping Tools (historical)

- HyperCard
  - for Macintosh – built by Bill Atkinson
  - metaphor: card transitions on button clicks
  - comes with widget set
  - drawing & animation limited
- Director
  - still commonly used by designers
  - intended for multimedia – originally lacked interface widgets or controls
  - good for non-widget UIs or the “insides” of app
  - Flash may replace Director for prototyping
- Both have “scripting” languages

HyperCard

- Tool palettes

Director Cast

- Basic objects used in interface
- Mainly multimedia in nature
  - images, audio, video, etc.
  - synchronize with cue points
Director Score

- Overview of events in time

Director Behavior Inspector

- Connect events to actions
- Drag & drop

UI Builders

- Visual Basic
  - lots of widgets (AKA controls)
  - simple language
  - slower than other UI builders
- MS Visual Studio .NET, JBuilder, PowerBuilder...
  - widgets sets
  - easily connect to code via "callbacks"
  - "commercial" strength languages

What's the Difference?

- Performance
  - prototyping tools produce slow programs
  - UI builders depend on underlying language
- Widgets
  - prototyping tools may not have complete set
  - UI builders have widget set common to platform
- Insides of application
  - UIBs do little, prototypers offer some support
- Final product
  - generally use UI builders, though occasionally see things created in a prototyping tool (multimedia)

Widgets

- Buttons (several types)
- Scroll bars and sliders
- Pulldown menus

Widgets (cont.)

- Palettes
- Dialog boxes
- Windows and many more...
What is Missing?

• Support for the “insides” of applications

Summary

• Informal prototyping tools bridge the gap between paper & high-fi tools
• High-fi UI tools good for testing more developed UI ideas
• Two styles of tools
  – “Prototyping” vs. UI builders
  – what is the difference?
• Both types generally ignore the “insides” of application → this is research

Further Reading

Prototyping

• Books
• Articles
• Web Sites
  – Group for User Interface Research, for DENIM & SUEDE downloads, http://guir.berkeley.edu

Next Time

• In class work on project
  – bring laptop
• Discussion Section
  – Visual Design
  • Read Kevin Mullet & Darrel Sano, Designing Visual Interfaces, Ch. 2 and 5 (online)