Lecture 6. Design (3)

CENG 412-Human Factors in Engineering
May 28 2009
Outline

Prototyping techniques:
- Paper prototype
- Computer prototype
- Wizard of Oz

• Reading:
  – Wickens pp. 50-57
What is prototyping?

• producing cheaper, less accurate renditions of your target interface
• Prototyping is essential in the early iterations of the design process
• Useful in later iterations too
Why prototype?

• Get feedback earlier and cheaper
  – ‘Which is harder to change: a program with 1000 lines of code or a 1000 square foot slab of concrete?’ Alan Cooper, The perils of prototyping

• Experiment with alternatives

• Easier to change and throw away

from Dr. Miller’s Lecture notes on UI Design and Implementation, MIT 2005.
A paper mock-up for urban development planning

http://www.vam.ac.uk/images/image/10026-large.jpg
Prototype Fidelity

• Fidelity measures how similar the prototype is to the finished interface
• Low fidelity: cheaper materials, omit details
• High fidelity: look more like finished product

from Dr. Miller’s Lecture notes on UI Design and Implementation, MIT 2005.
Dimensions of fidelity (1)

• **Breadth:** % of features covered by the prototype
  – Low-fi in breadth: Only enough features for a certain task

• **Depth:** degree of functionality of the prototype
  – Low-fi in depth: limited choices, canned responses, no error handling

from Dr. Miller’s Lecture notes on UI Design and Implementation, MIT 2005.
Dimensions of fidelity (2)

• Look: appearance, graphic design
  – Sketchy, hand-drawn

• Feel: input method
  – Physical methods by which the user interacts with the prototype
  • E.g.: Pointing and writing feels very different from mouse and keyboard
Paper Prototype

• Elements on separate pieces of paper
• Can be sketched and/or screen shots
Paper prototype

- **Interactive** paper mockup
  - Sketches of screen appearance
  - Paper pieces show windows, menus, dialog boxes
- **Interaction is natural**
  - Pointing with a finger=mouse click
  - Writing=typing
- **A person simulates the computer’s operation**
  - Putting down and picking up pieces
  - Writing responses on the screen
  - Describing effects that are hard to show on paper
- **Dimensions of fidelity...**

from Dr. Miller’s Lecture notes on UI Design and Implementation, MIT 2005.
Why paper prototyping?

• Faster to build
  – Sketching is faster than programming

• Easier to change
  – Easy to make changes between user tests, or even during a user test
  – No code investment – everything will be thrown away (except the design)

• Focuses attention on the big picture
  – Designer doesn’t waste time on details
  – Customer makes more creative suggestions, not nitpicking

• Nonprogrammers (e.g. graphic designers, usability specialists) can help: no special skills required

from Dr. Miller’s Lecture notes on UI Design and Implementation, MIT 2005.
Tips for good paper prototyping

• Make it larger than life
  – Fingers are bigger than a mouse pointer
  – The prototype lies on a table and you are testing users with it.
• Make it monochrome
• Replace tricky visual feedback with audible descriptions
  – “A progress bar pops up: 20%, 50%, 75%, done”
• Keep pieces organized: binders, two-pocket folders etc.
Materials

- Poster board or sturdy paper for background
- Plain paper
  - Maybe graph paper if characters/pixels count
- Unlined index cards
- Colored markers
- Transparency, pens
- Highlighter
- Scissors
- Correction fluid
- Transparent tape
- Removable tape*
- Restickable glue*
Paper prototyping video demonstration
How to test a paper prototype

• Roles for the design team
  – ‘Computer’
    • manipulates prototype
    • Doesn’t give any feedback that the computer wouldn’t
  – Facilitator
    • Explains purpose and process of the user study
    • Obtains the user’s informed consent
    • Presents the tasks one by one
    • Encourages user to think aloud while the user is working on a task
  – Observers
    • Just for watching, not helping the user
    • Takes notes
From Mark Rettig, Prototyping with tiny fingers
Interaction with a PP

• It’s important for users to see how the interface reflects their actions

• Use realistic data whenever possible

From Carolyn Snyder, Paper prototyping
Reflect User Actions

From Carolyn Snyder, Paper prototyping
What you can learn from a paper prototype

• Conceptual model
  – Do users understand it?
• Functionality
  – Does it do what is needed?
• Navigation and task flow
  – Can users find their way around?
  – Are task preconditions met?
• Terminology
  – Do users understand labels?
• Screen contents
  – What needs to go on the screen?

from Dr. Miller’s Lecture notes on UI Design and Implementation, MIT 2005.
What you can’t learn

• Look: color, font, white space etc.
• Feel: interaction with paper prototype may be different than with final UI
• Response time (backend simulated by ‘human computer’)
• Are small changes noticed?
  – Even the tiniest change to a paper prototype is clearly visible to the user
• Exploration versus deliberation
  – Users are more deliberate with a paper prototype; they don’t explore as much as with the real product.

from Dr. Miller’s Lecture notes on UI Design and Implementation, MIT 2005.
Computer prototype

• Interactive software simulation
• High-fidelity in look & feel
• Low fidelity in depth
  – Paper prototype has a human simulating the backend; the computer prototype doesn’t
  – Computer prototype is typically horizontal: covers most features, but no backend.

from Dr. Miller’s Lecture notes on UI Design and Implementation, MIT 2005.
What you can learn from computer prototypes

• Screen layout
  – Is it clear, overwhelming, distracted, complicated?

• Can users find important elements?

• Colors, fonts, icons, other elements
  – Well-chosen?

• Interactive feedback
  – Do users notice and respond to status bar messages, cursor changes, other feedback?

from Dr. Miller’s Lecture notes on UI Design and Implementation, MIT 2005.
Computer prototyping techniques

• Storyboard
  – Sequence (graph) of painted screenshots connected by hyperlinks (“hotspots”)

• Form builder: a tool for drawing real, working interfaces by dragging widgets from a palette and positioning them on a window

• Wizard of Oz: hybrid between a computer prototype and a paper prototype

from Dr. Miller’s Lecture notes on UI Design and Implementation, MIT 2005.
Storyboarding tools

• HTML: each screen is an image map
• Flash:
  – Tool for constructing multimedia interfaces;
  – Particularly useful for prototyping interfaces with rich animated feedback
• PowerPoint: images + links + animation

from Dr. Miller’s Lecture notes on UI Design and Implementation, MIT 2005.
Pros and cons of storyboarding

• Pros
  – You can draw anything

• Cons: storyboarding is static
  – All you can do is click, not enter text
  – Widgets (scrollbars, list boxes, buttons) are static pictures
  – Testing with users: “Hunt for the hotspot”

from Dr. Miller’s Lecture notes on UI Design and Implementation, MIT 2005.
Form builders

• HTML pages and forms
• Natural if you’re building a web application: prototype uses static HTML pages for simulating dynamic responses of your web interface
• Visual Basic
• Java GUI builders: Sun NetBeans, Eclipse Visual Editor, Borland JBuilder
• Example: MATLAB Guide

from Dr. Miller’s Lecture notes on UI Design and Implementation, MIT 2005.
Pros and cons of Form Builders

• Pros
  – Use actual working widgets, not just static pictures.
  – High fidelity in look and feel

• Cons
  – Don’t separate design from development

from Dr. Miller’s Lecture notes on UI Design and Implementation, MIT 2005.
Computer prototyping: an example

- The Nokia TV guide: a dialogue system customized for TV program information to be used in the Nokia Media Terminal
- Iterative computer prototyping
  - first iteration: prototype able to respond to isolated questions (no context used when interpreting a request)
  - second iteration: enriched interaction, system more ’intelligent’

http://www.ida.liu.se/~nlplab/miina
Computer prototyping example: 2\textsuperscript{nd} iteration

http://www.ida.liu.se/~nlplab/miina
Computer prototyping example: 3\textsuperscript{rd} iteration

http://www.ida.liu.se/~nlplab/miina
Computer prototyping example: final version-video

Allows for spoken interaction.
http://www.ida.liu.se/~nlplab/miina/dialogue1.mov
Wizard of Oz Prototype

• Software simulation with a human in the loop to help
• “Wizard of Oz” = “man behind the curtain”
  – Wizard is usually but not always hidden
• Often used to simulate future technology (e.g. speech recognition, visual gesture recognition etc.)
• Issue:
  – Two UI to worry about (user’s and wizard’s)
Wizard of Oz example - video