Prototyping

Prof. Narges Mahyar
UMass Amherst

nmahyar@cs.umass.edu
Courses, projects, papers, and more:
http://groups.cs.umass.edu/nmahyar/

© Mahyar with acknowledgements to Joanna McGrenere and Leila Aflatoony
Today

- Discussion of readings [10 min]
- Prototyping [35 min]
  - Low fidelity
  - Medium fidelity to high fidelity
- In class activity [20 min]
- Feedback [10 min]
Discussion on requirement readings [10 min]

• A randomly assigned team will summarize and discuss readings:
  • What you learned?
  • What surprised you?
  • How can you use this knowledge in your project?
Learning Goals

• Understand different types of prototyping, purpose and characteristics of each.
• List dimensions of prototyping fidelity and explain how these dimensions may vary;
• Explain how these dimensions might differ in low to med to high fidelity prototypes, and give examples of when/why you may use each type
• Make strategic choices about prototyping tools given your goals and constraints; be able to justify your choice.
Conceptual models & conceptual design:

- Conceptual model = the foundation of the interface. Different user interfaces could be built upon it.

- *Interface design* translates the *conceptual model* into things we can see and interact with. It involves design choices, but must stay faithful to the concepts and terminology of the conceptual model.
Conceptual design is:

• Designing systems so users can understand them
• Assisting the user to build useful mental models

**Interface design is:**

• Representing the CM to the user
From conceptual models to interface design

• **Interface design goal** is to communicate your conceptual model

• **Problem:**
  • *Designer’s conceptual model* is communicated via *system image*:
    Interface, appearance, instructions, system behavior through interaction

  • If system image does not make model clear and consistent:
    → User’s mental model will be inconsistent with conceptual model
How to get started on interface design?

• Prototyping!
What is a prototype?

Representation of conceptual design for users (and designers, and other stakeholders) to interact with

• Prototypes take many forms:
• Cardboard, foam, software, video, clay, paper, hidden people, website, sketches, scripts, index cards etc.
Handheld “universal remote control”

Conceptual Prototypes
Google Rapid Prototyping video
Why prototype?

- **Communication**: discuss ideas with stakeholders
  - “Where’s the ON button?”

- Develop **requirements** and/or specifications
  - “Uh-oh, here’s something we forgot.”

- **Learning** and problem solving
  - “Hey, that will work!”

- **Evaluate** interface effectiveness for communicating conceptual model
  - “Whoops, users didn’t understand that.”

- Further develop **conceptual and physical design**
  - “that’s way too heavy”

- **Save time and money**
  - Don’t waste time coding/building the wrong thing

many different kinds of goals and questions possible
Questions that *might* need prototyping to answer:

• For example:
  - Screen too crowded? Actions clear, or lost in clutter?
  - Knob versus slider for controlling volume
    - Much more involved for innovative physical interface
    - … Imagine the prototyping for the first iPhone!
  - Navigation: e.g.
    - Transparent menu versus solid menu
    - How many files to show in file selection box

What stage of design would you want to establish this sort of question?
Before you can prototype

Before you build, identify:
• Users and tasks to build your prototype around
• Requirements
• Goals: questions your prototype(s) need to answer
When to use different types of prototypes?

early design

• Choose a representation
  • Rough out interface style
    • Task walkthrough & redesign
• Fine tune interface, screen design
  • Heuristic evaluation and redesign

late design

• Low fidelity prototypes
• Medium fidelity prototypes
• High fidelity prototypes
 • Working systems

Usability testing and redesign
• Limited field testing
  • Alpha/beta tests
Low fidelity prototypes

Meant to be rough, quick to build, easy to throw away

Purposes

• Proof of concept(s)
• Rough (but flexible) interface design
• Facilitate communication with users early on
  • Can be useful for generating and narrowing requirements
Benefits of low fidelity prototypes

Cheap/easy to make
  • Try out and explore multiple conceptual models

Lack of polish less intimidating to users
  this is surprisingly important
  • More willingness to criticize
  • Inspires more creative feedback
  • Avoids nitpicky feedback

Reduces effort invested by design team
  • So easier to make changes, start over
IDEO surgical tool prototype
Approaches to prototype/product integration

• **Throw-away**
  • Prototype only serves to elicit user reaction
  • Creating prototype must be **rapid**, otherwise too expensive

• **Incremental**
  • Product built as separate components (modules)
  • Each component prototyped and tested, then added to the final system

• **Evolutionary**
  • Prototype altered to incorporate design changes
  • Eventually becomes the final product
Approaches to ‘scoping’ prototype functionality

• **Vertical** prototype
  - Includes *in-depth functionality* for only a few selected features
  - Key design ideas can be tested in depth

• **Horizontal** prototype
  - *Surface layers only*: includes the entire user interface *with no underlying functionality*
  - A simulation; no real work can be performed

• **Prototype scenario**
  - *Scripts* of particular fixed uses of the system; no deviation supported
  - See whole thing (fake)
  - *Use* implemented small part of it.
Paper prototyping

Common low fidelity technique

Popular in industry . . .

Despite prevalence of ‘mockup’ software tools

Because: easy to
• Build
• Alter on the fly
• Show
• Stick on wall & compare
• Discuss

Paper prototyping materials

• Interface elements/screens created on paper
  • Or other ‘easy to throw away or modify’ materials, e.g.,
    - Whiteboard, magnetic taps, transparencies

Can incorporate other things that people interact with in completing their task, e.g:

• Other people
• Hardware
Simulating interactions in paper prototyping

Can simulate relatively sophisticated interactions
  - Complex/subtle interactions won’t be perfect
  - Requires some imagination on users part
  - *Forces you to stay in “early design” mode*

With some creativity, can mockup almost any kind of widget or interaction
Rapid Prototyping 1 of 3: Sketching & Paper Prototyping

https://www.youtube.com/watch?v=JMjozqJS44M
Technique: digital storyboards

- Draw each storyboard scene on computer
  - Use wire framing/mockup software (e.g., Balsamiq)
  - Or painting/drawing packages (e.g., Photoshop)
- A very thin horizontal prototype
- Does not capture the interaction “feel”
Technique: scripted simulations & slide shows

• Encode the storyboard on the computer
  • Scene transition activated by simple user inputs (i.e. Clickable regions)
  • A simple horizontal and/or vertical prototype
  • Supports ‘limited’ branching

• User given a very tight script/task to follow
  • Appears to behave as a real system
  • But script deviations blow the simulation

moving towards med-fi elements can be active – but still only narrow functionality
Wizard of Oz

• A method of testing a system that does not exist
  • the voice editor, by IBM (1984)

Wizard of Oz ("WOZ")

• Human simulates system’s intelligence & interacts with user
  • “Pay no attention to the man behind the curtain!”

• User uses computer as expected

• “Wizard” (sometimes hidden):
  • Interprets subject’s input according to a preset algorithm
  • Makes computer/screen behave in appropriate manner

• Good for:
  • Adding simulated and complex vertical functionality
  • Testing futuristic ideas
Wizard of Oz examples

• IBM: an imperfect listening typewriter using continuous speech recognition
  • Secretary (i.e., Wizard) trained to:
    • Understand key words as “commands”
    • Type responses on screen as the system would
    • Manipulate graphic images through gesture and speech

• Intelligent agents / programming by demonstration
  • Person trained to mimic “learning agent”
    • User provides examples of task they are trying to do
    • Computer learns from them
  • Shows how people specify their tasks
Summary of Lo-Fi

• Prototyping
  • Speeds up design and lowers overall cost
  • Allows users to react to the design and suggest changes
  • Prototypes and scenarios are used throughout design
  • Low-fi best for brainstorming and choosing a conceptual model
  • Med/hi-fi prototypes best for fine-tuning and detailed design

• Low-fi prototyping methods
  • Scope: vertical, horizontal prototyping
  • Paper
  • Sketching
  • Storyboarding
  • Scripted simulations
  • Wizard of oz
Medium-fidelity prototypes

- Prototyping with a computer
- Engaging for end users
- Simulate some but not all features of the interface (interactive)
- Can test more subtle design issues

Pitfalls
- Blinds people to major representational flaws because of a tendency to focus on more minor details
- Users reluctant to change/challenge designer
- Management may think its real!
What’s the difference between “low” and “medium”?

- Used to be obvious! Paper vs. Nearly anything else.

- In last ~10 years: many powerful tools that:
  1. Make it very easy (a low-fi trait) to generate mockups
  2. Look real and are at least somewhat interactive (usually a “medium fidelity” trait)
- E.G.: Balsamiq, axure – low or medium; usually not high
Many dimensions of “fidelity”

• What are ways a prototype can be ‘true to life’?
  • **Visual realism**: how real it *looks*. Polish, graphic imagery
  • **Physical realism**: shape and form for 3D objects; feel
  • **Scope**: how many functions included; horizontal vs vertical
  • **Functionality**: what actually works? E.G. Web app: links live?
  • **Data**: operates on real vs faked data
  • **Autonomy**: operates alone vs requires “supervision”
  • **Platform**: interim vs final implementation
Important lessons:

1) It is COMPLICATED (slow, expensive) to prototype multiple dimensions at once.
   → So don’t. Instead: *modularity of prototyping*.

2) Each prototyping tool has strengths and weaknesses
   - May be *better* (more efficient and capable) for some of these prototyping dimensions than others.
   → You may need multiple tools throughout your design’s life cycle.
Matching game: what medium makes most sense for each dimension?

Prototyping dimension:

• How real it looks (polish)
• Scope how many functions included; horizontal vs vertical
• Real vs faked functionality how much of it is faked?
• Operates on real vs faked data
• Operates alone vs requires “supervision”
• For 3D products: physical aspects, or just images?
• Interim vs final platform

Useful Links:

https://www.creativebloq.com/advice/the-8-best-prototyping-tools-for-2018
https://kfginternational.com/blog/top-prototype-ux-ui-tools/
http://www.nexgendesign.com/top-7-prototyping-mockup-tools

• Prototyping medium:
  • Paper
  • Balsamiq
  • Axure
  • Invision
  • Proto.io
  • Flinto
  • Powerpoint
  • Html (or dreamweaver)
  • Java/swing
  • Processing
  • Modeling foam & hot-melt glue
  • Flash
  • Visual basic
  • Photoshop
  • Arduino
  • Found objects
  • Tcl/tk
  • Python
  • Pop
A competitive analysis of prototyping tools

<table>
<thead>
<tr>
<th>Prototyping Tools Compared</th>
<th>Invision</th>
<th>Flinto</th>
<th>Principle</th>
<th>Proto.io</th>
<th>Origami Studio</th>
<th>Framer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sketch Import &amp; Sync</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Screen Transitions</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Micro Interactions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Collaboration Tools</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Preview on iOS</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Preview on Android</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Share on iOS</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Share on Android</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Low Learning Curve</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>User Testing Tools</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Device Integrations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

https://uiux.blog/quest-for-the-perfect-prototyping-tool-ef35f89bfb31
You can even make medium-fidelity mockups look low-fi

- this graphic is generated from code (*processing*).
  http://www.gicentre.org/handy/  [more examples]
Balsamiq: low to medium

- Quickly mock up images and hyperlinked interactivity.
- But - real functionality difficult.
Difference between med to high-fidelity prototypes

Increasing in completeness and detail:

- More aspects being prototyped at same time
- Higher degree of functionality
- Higher degree of polish
- Etc. . .

Fidelity is a spectrum

- Not always a firm line between low/med or med/hi
Medium and high-fidelity prototyping what can you use?

• Many things:
  • Drag-and-drop GUI toolkits for standard UI mockups
    • E.G. Axure, visual basic
  • Scripting languages & interface libraries
    • E.G. Python, tcl/tk, java script libraries (e.g., Jquery)
  • Graphical languages for visualization & novel interface creation
    • VB, java, flash; processing; D3
  • Special purpose tools and environments
    • E.G. Toolkits for integrating speech, haptics, I/O devices

→ A prototyping platform can be medium- or hi-fi; depends on how you use it.
The situation today for prototyping tools (vs. Developing on final platform)

• For simple prototyping.
  • Balsamiq, axure, html, powerpoint

• More advanced features in e.g. Supercard, director:
  • Text-to-speech, speech recognition, quicktime, filmstrips, graphic import and export, MP3 playback etc.

• Advanced uis still require (scripting) language + libraries
  • HTML + javascript
  • Tool command language/tool kit (TCL/TK)
  • Python
  • Processing (java based, but way more accessible; good for sketching, no good for larger code projects)
  • Still a need for C++, C#, objective C, java
Home alarm system

Flash:

- Product for the home
- Needed to gauge reactions to having it in ones house
- Imagery + graphic resolution critical
Flex:

needed to test how well the concept worked for actually taking notes in lecture

highly functional
detailed vertical
Sonic stage music synchronization tool

Flash w/ imported photoshop

observe scanned, hand-drawn sketches
How do you know when you have – or need – a high-fi prototype?

• Scope is complete (horizontal and vertical)

• Prototype can be tested in just about every way performance as well as subjective and cognitive analysis; more realistic scenarios; in field

• Feels like time to switch to final development platform

• Design is becoming rigid and finalized
On deck...

• Next class …
  • Tuesday oct 22: midterm
  • No class on Thursday oct 24
Activity [20 min]

Discuss the following questions for your own projects:

• What are the main challenges that your prototype means to solve?
• What fidelity seems right for your prototype?
• What dimensions you need to consider?
• What are some possible tools for your project? What are the tradeoffs?
Extra slides
summary
low fidelity vs. high fidelity

• cheap
• easy to build lots
• facilitate communication
• gross design (layout)
• market requirements
• proof-of-concept
• limited error checking
• hard to get to code
• facilitator driven
• limited functionality

• complete functionality
• interactive
• user-driven
• exploration and testing
• look and feel of final product
• provides specification
• marketing and sales tool
• expensive
• time consuming
• inefficient proof-of-concept
• poor for requirements gathering
• can be hard to throw away
Tools available to you

• *Balsamiq*

• *Axure*

• *The Adobe Suite,* Photoshop, InDesign, Illustrator, Premiere, and Dreamweaver

Microsoft Office suite (PPT can be useful).

• Note that many tools have 30-day free trials