PROTOTYPING:
LOW TO HIGH FIDELITY PROTOTYPING

690A- Advanced Methods in HCI

Prof. Narges Mahyar
TODAY

- Conceptual design vs. interface design [5 min]
- Prototyping [30 min]
  - Low fidelity
  - Medium fidelity to high fidelity
- In class activity [20 min]
- Discussion of readings [15 min]
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 CM into things we can see and interact with. It involves design choices, but must stay faithful to the concepts and terminology of the CM.
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

review
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.

4 designs: image-enhanced planner
HANDHELD “UNIVERSAL REMOTE CONTROL”
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

**low fidelity prototypes**

**medium fidelity prototypes**

**high fidelity prototypes**

**late design**
- Usability testing and redesign
- Limited field testing
- Alpha/Beta tests
- Working systems
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.

**most relevant for low- and med-fi prototypes (when scope is limited)**
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”

Control panel for pump 2

<table>
<thead>
<tr>
<th></th>
<th>coolant flow 45 %</th>
<th>retardant 20%</th>
<th>speed 100%</th>
<th>Shut Down</th>
</tr>
</thead>
</table>

Control panel for pump 2

<table>
<thead>
<tr>
<th></th>
<th>coolant flow 0 %</th>
<th>retardant 20%</th>
<th>speed 100%</th>
<th>Shut Down</th>
</tr>
</thead>
</table>

DANGER! (for shut down condition)

next drawing
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

Control panel for pump 2

- coolant flow 45%
- retardant 20%
- speed 100%

Shut Down

Control panel for pump 2

- coolant flow 0%
- retardant 20%
- speed 100%

DANGER!

next drawing (on mouse press over button)

Shut Down

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
## 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 for add’l flexibility
  • 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
E-READER & NOTE-TAKING TOOL

Hybrid View:
Split views for displaying two files simultaneously.

References: Can make hyperlink references between content

All controls are preserved

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
ACTIVITY

get into groups of 2-3

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?
DISCUSSION ON REQUIREMENT READINGS [20 MIN]

- What surprised you? or
- What you disagreed with?
- Others?
ON DECK...

Next class (TUESDAY) …

• Cognitive Walkthroughs
• Heuristic Evaluation

Readings (as posted)
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* hands out course licenses – let me know if you want me to ask for one.

*Axure* is installed on the X360 (HCI Studio) computers with a license that is a few years old, but still operable.

*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