PROTOTYPING: LOW TO HIGH FIDELITY PROTOTYPING

CPSC 544 FUNDAMENTALS IN DESIGNING INTERACTIVE COMPUTATION TECHNOLOGY FOR PEOPLE (HUMAN COMPUTER INTERACTION)

WEEK 8 – CLASS 15
TODAY

- Admin [5 min]
- Prototyping [30 min]
  - Low fidelity
  - Medium fidelity to high fidelity
- In class activity [20 min] -> interactive session [10 min]
- Discussion of readings [10 min]
- Working time [10 min]
ADMIN

• Informal midterm evals
• Ideate status – Sally and I meeting after class, hope to release feedback by end of day
• Briefly step through Prototype milestone
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
RECALL: 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.
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
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
NOT JUST FOR SOFTWARE: HANDHELD "UNIVERSAL REMOTE CONTROL"
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:
  • 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?
WHEN TO USE DIFFERENT TYPES OF PROTOTYPES?

- Choose a representation
- Rough out interface style
- Task walkthrough & redesign
- Fine tune interface, screen design
- Heuristic evaluation and redesign
- Usability testing and redesign
- Limited field testing
- Alpha/Beta tests

early design

- Low fidelity prototypes

late design

- Medium fidelity prototypes
- High fidelity prototypes
- Working systems
User Interface Design Process: Evolving Iterations

**Understand USERS:**
- who they are
- their key tasks

**Understand DESIGN:**
- design space and risks
- choose design approach

**REFINE Design:**
- by element
- considering task
- varied contexts

**CONFIRM & debug:**
- performance in real use

**Examine existing:**
- user tasks & objectives
- contexts
- interfaces

**Make use of:**
- requirements
- task analysis
- real & virtualized users
- technology options
- company IP

**Evaluate w/:**
- observation
- interview/quest
- participatory interaction
- task walk-throughs

**Evaluate w/:**
- usability testing – controlled, uncontrolled
- heuristic evaluation

**Make use of:**
- graphical design
- interface guidelines
- style guides
- real & virtualized users

**Evaluate w/:**
- observation – many kinds
- ethnography
- interviews, questionnaires
- task analysis

**Evaluate w/:**
- usability testing – controlled, uncontrolled
- heuristic evaluation

**Make use of:**
- low fidelity prototyping methods

**Make use of:**
- medium/high fidelity prototyping methods

**PRODUCTS**
- user and task descriptions
- design requirements
- throw-away prototypes
- design direction
- risk analysis
- testable medium-fidelity prototypes

**MATERIALS / METHODS**
- requirements
- task analysis
- real & virtualized users
- technology options
- company IP

**GOALS**
- who they are
- their key tasks

**EXAMINE existing:**
- user tasks & objectives
- contexts
- interfaces

**Evaluate w/:**
- observation
- interview/quest
- participatory interaction
- task walk-throughs

**EVALUATE w/:**
- usability testing – controlled, uncontrolled
- heuristic evaluation

**MAKE use of:**
- graphical design
- interface guidelines
- style guides
- real & virtualized users

**CONFIRM & debug:**
- performance in real use

**Field testing**

**Release!**
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

Fully prototyped system

most relevant for low- and med-fi prototypes (when scope is limited)
APPROACHES TO ‘SCOPING’ PROTOTYPE FUNCTIONALITY

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

Horizontal prototype
APPROACHES TO ‘SCOPING’ PROTOTYPE FUNCTIONALITY

Vertical prototype

most relevant for low- and med-fi prototypes (when scope is limited)
APPROACHES TO ‘SCOPING’ PROTOTYPE FUNCTIONALITY

Common goal to support a scenario…

Horizontal prototype

Vertical prototype

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

Low-fi prototyping does not need to be paper only…

haptok prototype (diane tam): the power of magnetic tape!
PAPER PROTOTYPING VS. SKETCHING

Difference of intention…

<table>
<thead>
<tr>
<th>Sketch</th>
<th>Prototype</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evocative</td>
<td>Didactic</td>
</tr>
<tr>
<td>Suggest</td>
<td>Describe</td>
</tr>
<tr>
<td>Explore</td>
<td>Refine</td>
</tr>
<tr>
<td>Question</td>
<td>Answer</td>
</tr>
<tr>
<td>Propose</td>
<td>Test</td>
</tr>
<tr>
<td>provoke</td>
<td>Resolve</td>
</tr>
<tr>
<td>Tentative</td>
<td>Specific</td>
</tr>
<tr>
<td>Noncommittal</td>
<td>Depiction</td>
</tr>
</tbody>
</table>

WIZARD OF OZ

A totally different 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

cons?
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
RECALL: 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”
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
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
**HCI prototyping**: kind of prototypes → what we do with them

Understand USERS:
- who they are
- their key tasks

Understand DESIGN:
- design space and risks
- choose design approach

REFINE Design:
- by element
- considering task
- varied contexts

CONFIRM & debug:
- performance in real use

GOALS

MATERIALS / METHODS

PRODUCTS

Examine existing:
- user tasks & objectives
- contexts
- interfaces

Evaluate w/
- observation – many kinds
- ethnography
- interviews, questionnaires
- task analysis

Make use of:
- requirements
- task analysis
- real & virtualized users
- technology options
- company IP

Evaluate w/
- observation
- interview/quest
- participatory interaction
- task walk-throughs

Make use of:
- graphical design
- interface guidelines
- style guides
- real & virtualized users

Evaluate w/
- usability testing – controlled, uncontrolled
- heuristic evaluation

- testable medium-fidelity prototypes

low fidelity prototyping methods

med/ high fidelity prototyping methods

Field testing

Release!

PRE DESIGN

EARLY DESIGN

MID DESIGN

LATE DESIGN

• user and task descriptions
• design requirements

• throw-away prototypes
• design direction
• risk analysis

• alpha/beta systems or complete specification

K MACLEAN - DERIVED FROM VERSION BY SAUL GREENBERG (U CALGARY)
MEDIUM-FIDELITY PROTOTYPES

• requires 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

• user’s reactions often “in the small”
• 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”?

less clear than it used to be.

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
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.
BEST PROTOTYPING TOOL?

No such thing. It depends.

Search on best prototyping tools…

11 Best Prototyping Tools For UI/UX Designers — How To Choose The Right One?

The 9 Best Go-To Prototyping Tools for Designers in 2019

etc.
## 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>

Copyright @ Shruthi Padala

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.
- [https://balsamiq.com/](https://balsamiq.com/) (good ~1min intro video)
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
UBC STUDENT AID – PAST CPSC344 PROJECT (HTML)

apply online

To apply for a BC Student online, please click "start"

Start

- Browser requirements
- Ongoing site maintenance
- Security

Browser requirements

This website is best viewed using Microsoft Internet Explorer 5.5 or higher or Netscape 6. x or higher. Earlier versions of these browsers may cause difficulty in navigating Studentaid BC Online or completing the application forms and are not recommended.

If you could not access Studentaid BC Online, check your browser version by following these links:

- Click 'Help' on your menu bar, then click 'About' to find your browser version.
- To upgrade your version of Microsoft Internet Explorer, click here.

To upgrade your version of Netscape, click here.

Ongoing site maintenance

html:

final platform didn’t need to be glitzy
easy to copy existing text, look and feel
then alter everything
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 with 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
CASE STUDY ACTIVITY: PART 1
INTRODUCTION + UIST SUBMISSION VIDEO PROTOTYPE

Siyan Zhao, Oliver Schneider, Roberta Klatzky, Jill Lehman, and Ali Israr. 2014. FeelCraft: crafting tactile experiences for media using a feel effect library. In UIST’14.
CASE STUDY ACTIVITY: PART 1
INTRODUCTION + UIST SUBMISSION VIDEO Prototype

one worksheet per person; work in pairs!

1) ~10 minutes to complete part 1
2) take up as a group

some high-level questions:

• what were the main challenges the prototype and video were meant to solve?
• what fidelity was the HTML prototype?
• did the authors choose the correct tools for the job? what were the tradeoffs?
CASE STUDY ACTIVITY: PART 2
ITERATION AND PROTOTYPE DEMO

1) ~10 minutes to complete part 2
2) take up as a group

some high-level questions:

• what were the main challenges the prototype and video were meant to solve?
• what fidelity was the demonstration prototype?
• did the authors choose the correct tools for the job? what were the tradeoffs?
• was the prototyping evolutionary? or modular?
ACTIVITY RECAP:

1. What *investigative challenges* do you need to answer?
2. What kind of *evaluation* should you do to answer that question?
3. What should a *prototype* to support that evaluation emphasize?
4. What *prototyping tool* might be a good choice?
DISCUSSION ON READINGS [20 MIN]

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

Next class (TUESDAY) …

• Cognitive Walkthroughs

• Heuristic Evaluation

Readings (as posted)