PROTOTYPING: 
LOW TO HIGH FIDELITY PROTOTYPING

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

WEEK 8 – CLASS 15

© Joanna McGrenere and Leila Aflatoony
Includes slides from Karon MacLean and Jessica Dawson
TODAY

- Conceptual design vs. interface design
- Prototyping
  - Low fidelity
  - Medium fidelity to high fidelity
- In class activity
- Discussion of readings
- Quick overview of MSIV
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.
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 transfer, idioms and stereotypes

• 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?

prototypes take many forms:

- cardboard, foam, software, video, clay, paper, hidden people, website, sketches, scripts, index cards etc.

the point: make ideas real

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

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
WHEN TO PROTOTYPE?

to get out of a rut, focus discussion, reach agreement
when you have questions and you can’t proceed:

• functionality:
  • structure, sequencing, flow
  • clarity & completeness of information

• appearance
  • branding, clarity, aesthetics, color, shape, etc.

• specifications
  • “design by prototyping” (evolutionary approach)

when you need to communicate ideas

• design team, managers, users etc.
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?
OR ... YOU COULD TRY “DESIGN BY PROTOTYPING”

rather than using prototype variations to answer specific questions, could take an

- *evolutionary* approach to system development
- uses the prototype itself as the specification

pros: complete model at end, [maybe] fast to market.

cons: haphazard, feature crammed application, no overall performance strategy, may have to start again.
BEFORE YOU CAN PROTOTYPE

before you build, identify:

• users and tasks to build your prototype around
• 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

Usability testing and redesign
Limited field testing
Alpha/Beta tests

Low fidelity prototypes

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
- many kinds
- ethnography
- interviews, questionnaires
- task analysis

**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

**Make use of:**
- testable medium-fidelity prototypes

**G O A L S**

**MATERIALS / METHODS**

**PRODUCTS**

**PRE DESIGN**

**EARLY DESIGN**

**MID DESIGN**

**LATE DESIGN**

**low fidelity prototyping methods**

**med/high fidelity zprototyping methods**

**K MACLEAN - DERIVED FROM VERSION BY SAUL GREENBERG (U CALGARY)**
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
HANDHELD “UNIVERSAL REMOTE CONTROL”
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.,
    - whiteboards
      • magnetic tape
      • transparencies

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

• Good video example here: http://www.parisinnovationreview.com/articles-en/the-power-of-prototyping
haptok prototype (diane tam): the power of magnetic tape!
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
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

possible cons?
WIZARD OF OZ

A method of testing a system that does not exist

- the voice editor, by IBM (1984)

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
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](image)
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

<table>
<thead>
<tr>
<th>Control panel for pump 2</th>
<th>Control panel for pump 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>coolant flow 45 %</td>
<td>coolant flow 0 %</td>
</tr>
<tr>
<td>retardant 20%</td>
<td>retardant 20%</td>
</tr>
<tr>
<td>speed 100%</td>
<td>speed 100%</td>
</tr>
<tr>
<td>Shut Down</td>
<td>Shut Down</td>
</tr>
</tbody>
</table>

moving towards medi-fi elements can be active – but still only narrow functionality

DANGER!

next drawing
(on mouse press over button)
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**

- user and task descriptions
- design requirements

- throw-away prototypes
- design direction
- risk analysis

- testable medium-fidelity prototypes
- alpha/beta systems or complete specification

**Examining 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

**Evaluate w/:**
- graphical design
- interface guidelines
- style guides
- real & virtualized users

**Make use of:**
- low fidelity prototyping methods
- med/ high fidelity prototyping methods

**Field testing**

**Releases!**
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

- 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”? 

it has become confusing …  
*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 prototyping platforms can be low or medium! (*depends on how you use it.*)

but, FEW are all three (low to 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.

3) 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


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
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 MEDIUM AND HIGH-FIDELITY PROTOTYPING?

fidelity is partly a matter of completeness:

as you get more hi-fi: may become more restricted in platform (you’re trying to do more at once).

e.g., html could be good for:

• approximate look plus interactivity for a real, 3-D interface like a home alarm system controller
• hi-fi prototype of a web interface here, functionality and looks are compatible.
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
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 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
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* is available for free on all UBC-owned computers. X360 computers are outfitted with Photoshop, InDesign, Illustrator, Premiere (+ Media Encoder), Dreamweaver, and the new UX tool called Experience Design.

Microsoft Office suite (PPT can be useful).


Note that many tools have 30-day free trials
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
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 REQUIREMENT READINGS [20 MIN]

• What surprised you? or
• What you disagreed with?
• Others?
PROJECT – MS IV
ON DECK...

Next class (TUESDAY) …

• Cognitive Walkthroughs
• Guest lecturer