strategic prototyping

and a focus on

low fidelity prototyping
Announcements

Please fill out the mid-semester survey

This is a great chance to send us anonymous feedback and directly impact how the course is taught. We’ll read every single response.

You’re also welcome to give us direct feedback in person, and/or talk to the TAs and ask them to shoot it up the ladder.
learning goals

describe the relationship between conceptual design and interface design, and why the distinction is important.

for a given situation, assess and explain why, when, what, and how you may want to prototype

explain how prototyping fidelity relates to design progression; and give examples of what prototypes at each stage might involve

describe different approaches to creating low fidelity prototypes (paper prototypes in particular)

explain the benefits and drawbacks of low fidelity prototyping (paper prototyping in particular)
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

“wrong” vs “simplified”? 
prototyping

when, what and how to prototype,
intro to prototype fidelity
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.
before you can prototype

before you build, identify:

• **users** and **tasks** to build your prototype around
• sketch of **requirements** you need to address
• **questions** your prototype(s) need to answer
types of prototypes

think of prototyping techniques as tools in your bag of tricks

• have lots so that you have appropriate one
• just like evaluation methods
• should be fast, effective and targeted to the issues
  ➔ don’t waste time implementing something that won’t teach you anything!

fidelity ranges from low to high
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
- Usability testing and redesign
- Limited field testing
- Alpha/Beta tests
- Low fidelity prototypes /
- Medium fidelity prototypes
- High fidelity prototypes / partially-working systems
- Working systems

late design
User Interface Design Process: Evolving Iterations

Understand USERS:
- who they are
- their key tasks

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

Evaluate w/:
- observation
- ethnography
- interviews, questionnaires
- task analysis

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

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

Low fidelity prototyping methods

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

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

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

Medium/High fidelity prototyping methods

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

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

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

Testable medium-fidelity prototypes

CONFIRM & debug:
- performance in real use

Field testing

Release!

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

Source: IDEO Case Study
http://www.ideo.com/case_studies/gyrus/
1. handheld “universal remote control”

Conceptual Prototypes
2. emulating human attention-getting practices with wearable haptics
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
approaches to ‘scoping’ prototype functionality

**vertical** prototyping
- includes *in-depth functionality for only a few selected features*
- key design ideas can be tested in depth

**horizontal** prototyping
- *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 corner of it.

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

<table>
<thead>
<tr>
<th>Low Fidelity</th>
<th>High Fidelity</th>
</tr>
</thead>
<tbody>
<tr>
<td>cheap</td>
<td>complete functionality</td>
</tr>
<tr>
<td>easy to build lots</td>
<td>interactive</td>
</tr>
<tr>
<td>facilitate communication</td>
<td>user-driven</td>
</tr>
<tr>
<td>very early layout design</td>
<td>exploration and testing</td>
</tr>
<tr>
<td>market requirements</td>
<td>look and feel of final product</td>
</tr>
<tr>
<td>proof-of-concept</td>
<td>provides specification</td>
</tr>
<tr>
<td>limited error checking</td>
<td>marketing and sales tool</td>
</tr>
<tr>
<td>limited coding</td>
<td>expensive</td>
</tr>
<tr>
<td>limited functionality</td>
<td>time consuming</td>
</tr>
<tr>
<td>can be hard to throw away</td>
<td>inefficient proof-of-concept</td>
</tr>
<tr>
<td></td>
<td>poor for requirements gathering</td>
</tr>
<tr>
<td></td>
<td>can be hard to throw away</td>
</tr>
</tbody>
</table>
Part 2: focus on low fidelity prototyping
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
demo

no activity today

instead, go through demo paper prototypes
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
  - *which interactions* to show is another question
testing
paper prototypes

conduct tests with users with a ‘Computer’

– one person pretends to be a computer
– has flow chart/script of possible actions
– responds to user interactions by manipulating inputs/system status/feedback etc. on prototype

... very simple form of wizard of oz prototyping

cognitive walkthroughs (W09)

– walkthrough and evaluate screens with experts
Wizard of Oz

human simulates system’s intelligence & interacts w/ user

uses real or mock interface
  • “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**
  • has computer/screen behave in appropriate manner

good for:
  • adding simulated and complex vertical functionality
  • testing futuristic ideas

possible 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
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

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

DANGER!

coolant flow 0%
retardant 20%
speed 100%

Shut Down

next drawing
(for shut down condition)
example:
balsamiq mockups linked together in Prezi or PPT

• can move between screens, but no real interactivity
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

Control panel for pump 2

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

next drawing
(on mouse press over button)

Control panel for pump 2

DANGER!

<table>
<thead>
<tr>
<th>Control panel for pump 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>coolant flow 0 %</td>
</tr>
<tr>
<td>retardant 20%</td>
</tr>
<tr>
<td>speed 100%</td>
</tr>
<tr>
<td>Shut Down</td>
</tr>
</tbody>
</table>
more low fidelity prototypes: tutorials and manuals

write them in advance of the system:

• tutorial for step-by-step description of an interaction
  – an interface “walk-through” with directions
• manual for reference of key concepts
  – in-depth technical description

if highly visual: storyboard is set within text explanations

a manual is a kind of prototype!
summary

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 prototyping methods
• vertical, horizontal prototyping
• paper
• sketching
• storyboarding
• scripted simulations
• Wizard of Oz