HCI Bootcamp
6-10 January 2014

Wendy E. Mackay
mackay@lri.fr

Jérémie Garcia
garcia@lri.fr

web: http://insitu.lri.fr/People/HCIBootCamp
Generative Design

Discovery
  Who is the user?

Invention
  What is possible?

Design
  What should it be?

Evaluation:
  Does it work?
Invention

What is possible?
**Invention: What is possible?**

Collect or sample information
- Web search
- Oral Brainstorming
- Video Brainstorming

Analyze information
- Preference votes
- Design dimensions

Create resources for design
- Key ideas
- Design space
How do you find the design concept?

Based on your studies of users, choose a problem to solve specific to your audience.

Generate a variety of ideas that offer potential solutions.

Create a design space to embody the set of alternatives.

Choose a concept to explore not just functionality, but also interaction.
Inspirations

Oregon scientific projection clock projects time and weather information onto the ceiling or the wall.
Inspirations

Consider the context:
Who needs this?
Where can they use it?
When do they need it?
For what purpose?
Generate new ideas

Brainstorming:
Imagine different situations in which users might interact with technology in a new way that meets a need or helps them do something new

Focus on interaction in context
not just a list of functions
Express interaction:

Several levels of representation

Text: explain an idea in words
     (Standard brainstorming)

Sketch: draw to illustrate an idea
       (Standard brainstorming)

Mockups: create and interact with paper prototypes
       (Rapid prototyping)

Theater: Act out the idea
       (Rehearse video brainstorming)

Video: Capture the details of the interaction
       (Video brainstorming)
Oral brainstorming rules

Phase I
Generate the maximum quantity of ideas
Everyone participates
Record every idea
... and everyone contributes at least one stupid idea

Phase II
Reread all the ideas
Everyone has three votes: mark your favorite ideas
Rank the ideas according to the number of votes
Discuss these ideas with respect to your design concept
Don’t forget weird or unusual ideas
Exercise : Oral brainstorming

Each group should choose:
  Moderator: Ensures that everyone participates
  Stops discussions and critiques,
  Keeps the time
  Scribe: Writes every idea
  Reads the ideas at the end

Remember:
  Generate the maximum number of ideas
  without evaluating them
  Quantity is more important than quality
  Everyone must participate
  Everyone has to give at least one ‘stupid’ idea
<table>
<thead>
<tr>
<th>Brainstorming: What not to do</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discuss ideas</td>
</tr>
<tr>
<td>Criticize ideas</td>
</tr>
<tr>
<td>Argue why an idea is good/bad</td>
</tr>
<tr>
<td>Ignore each other’s ideas</td>
</tr>
<tr>
<td>Shift topics</td>
</tr>
<tr>
<td>Jump to abstractions</td>
</tr>
<tr>
<td>Get stuck</td>
</tr>
</tbody>
</table>
**Opposites Technique**

If you get stuck, push existing ideas in new directions

<table>
<thead>
<tr>
<th>Opposites:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>simple</td>
<td>complex</td>
</tr>
<tr>
<td>short</td>
<td>long</td>
</tr>
<tr>
<td>direct</td>
<td>indirect</td>
</tr>
<tr>
<td>good</td>
<td>bad</td>
</tr>
<tr>
<td>direct</td>
<td>indirect</td>
</tr>
<tr>
<td>text</td>
<td>graphic</td>
</tr>
<tr>
<td>funny</td>
<td>serious</td>
</tr>
<tr>
<td>process</td>
<td>objet</td>
</tr>
<tr>
<td>start</td>
<td>end</td>
</tr>
<tr>
<td>single</td>
<td>sequence</td>
</tr>
</tbody>
</table>
Video brainstorming

Goal: Capture the interaction between the user and the system being designed

For each idea:
Choose a director who has complete control over:
The description of the idea
How to video the idea
Which assistants do what
Scribe: fills out video title card and idea list, keeps materials
Camera person: videos the title card and the action
Makers: create the paper prototype
Actors (talent): perform the interaction, record voice-overs
Exercise: Video brainstorming

Goals:
- Capture as many ideas as possible
- Illustrate the interaction: show the user’s experience
- Explore a theme and variations
- Only one director per idea
- Do not waste time arguing, the director decides
- If you disagree, be the director for take 2
- Each idea is short:
  - these are not scenarios
- Use post-its, transparencies, etc.
Generative Design

Discovery
  Who is the user?

Invention
  What is possible?

Design
  What should it be?

Evaluation
  Does it work?

Redesign
  How to improve it?

user profile
  design space
  technical possibilities
  design brief
  design alternatives
  analysis
  implications for design
  interpret
  axes
  user insights
Design
What should it be?
Design requires **choices**

Prototypes help express specific concepts at different levels of representation. **Goal:** *quality*, not quantity of ideas.

**Careful!** Each choice limits options. But also poses new questions and may suggest new possibilities.
Design: What should it be?

Collect information
- Design brief
- plus results from earlier phases

Analyse information
- Function-Interaction table
- Design alternatives

Resources for design
- Design scenario
- Storyboard
- Mockups & Paper prototypes
- Video prototype

Earlier results
- Design scenario
- Video prototype
- Storyboard
- Mockup

Alternatives
- Table
## Design: *What should it be?*

<table>
<thead>
<tr>
<th>Gather inputs</th>
<th>Analyze data</th>
<th>Design resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Profile</td>
<td>Conceptual Model</td>
<td>Design alternatives</td>
</tr>
<tr>
<td>Use Scenario</td>
<td>Function-interaction table</td>
<td>Design scenario</td>
</tr>
<tr>
<td>Design space</td>
<td>Rapid prototypes</td>
<td>Storyboard</td>
</tr>
<tr>
<td>Key ideas</td>
<td></td>
<td>Video prototype</td>
</tr>
<tr>
<td>Design Brief</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Exercise: Design space

Identify the key ideas

Extract different design dimensions that characterize the ideas

Place the ideas along the design dimensions
- at least three ideas per dimension
- generate new ideas if you find gaps
- explore the intersections of different dimensions

Select a subset of dimensions and ideas to create the design space
Design space
Creating a concept

Define your project within the scope of the design space

Identify a real, specific problem.
  Real problems tend to be complex and messy
  Look for a small, simple aspect of a real problem
  Rather than a stereotypical ‘toy’ problem

Trade-off between power and simplicity:
  Less is More

Be curious, be creative, seek surprises and new opportunities
Choose a concept

Observe users

Generate ideas

Create a design concept
Lego Znap

Pose design questions about how users will interact

Propose ways to address them

How to introduce ‘big moments’ into the model?
Lego Znap

How to encourage kids to add to their system?

Q. How can we encourage the kids to add to their system?

A. Very cheap "bubble gum" pieces with collectable themed trading cards - "rare piece" and inspirational pictures which can be collected.
Describe a design concept

How will the system work?
  Functionality what should it do?
  User guide how does it work?
  Scenario what happens in real-world contexts?

Justification
  What are the alternatives?
  What are the advantages and disadvantages of this solution?
Iterating on a design concept

Based on the use scenario, personas and user profile together with the key ideas from your design space

Discuss your design concept:
Consider how the users in the scenario will react
  Does it respond to real user needs?
  Is it specific?
  Is it technically possible?

Build on your design resources:

User perspective:
  User profile
  Personas
  Use characteristics
  Use scenario

System possibilities:
  Design problem
  Design space dimensions
  Key or favorite ideas
  Design space
Exercise: Design concept

What is your design concept?
Prototyping interaction

Design scenario
  Imagine the system from the user’s perspective

Wizard of Oz
  Simulate the system live
  with a human operator ‘behind the curtain’

Video Prototype
  Illustrate the use of the system in context
  “sketch” dynamic, interactive user experiences

Simulation
  Create a working subset of the system
What is a prototype?

Prototype =
concrete representation of an interactive system

Characteristics

Representation: form of prototype
- sketches - simulations
Precision: level of detail
- informal - complete
Interactivity: interaction
- watch - interact
Evolution: lifecycle of prototype
- throw out - iterative

The choice of prototype depends upon the
design phase and the specific needs of the designers
Video supports every phase of design

3. Design: Video prototypes illustrating use in context

2. Brainstorming: Video prototypes of design ideas illustrating interaction patterns

Design principles

Interaction techniques

Contexts of use

Interaction patterns

4. Evaluation: Video clips of use of new tool

1. Observation: Video clips of use scenarios
Video supports every phase of design

Evaluation:
Users try new system

Observation:
Users in context

Interaction patterns

Use scenarios

Design principles

Interaction techniques

Prototyping:
Design possibilities

Brainstorming:
Explore new ideas
Prototyping helps the designer …

Consider different design alternatives

Ensure usability under diverse conditions

Help users and other stakeholders imagine the interface

Focus on problematic parts of the interface
Representation

Paper prototypes
Easy and fast to create and to throw away
Most useful at the beginning of the design process
examples: sketches for an idea for an icon,
storyboard sequences, mockups of screens,
video prototypes of a complex interaction

On-line prototypes
Use the computer, longer to create, more polished
More appropriate later in the design process
examples: animations, interactive videos,
scripting languages, interface builders
Precision

Low fidelity (lofi) prototypes with little detail
  Great for rapid exploration of ideas
  example: paper sketches, SILK

High fidelity (hifi) prototypes, very detailed
  Good to communicate specific design considerations
  example: dialog box with layout alternatives

Note: A detailed representation is not always precise
  It is possible to omit aspects that have not yet been decided
Details

A system can be good in theory but unusable in practice because of flaws in the interface ... even small ones.

Good prototypes let designers work with different sets of details at the same time.

Good prototypes allow users to envision the final system: but also to feel comfortable suggesting changes.
Level of Interactivity

Non-interactive (fixed)
No interaction, but can show potential interaction
example: a video clip showing user interacting with a device

Low interaction (pre-determined path)
Can test several alternative forms of interaction
example: designer shows a screen shot, user indicates her action, the designer shows the result

High interaction (open)
Users interact with the system, with some limitations
example: Wizard of Oz or computer-based simulation
Wizard of Oz

Technique for prototyping novel user interfaces

Wizard of Oz:
Designer ‘plays computer’ to create an interactive experience for the user

Useful for creating video prototypes but also for creating live experiences that rapidly explore different design alternatives
Wizard of Oz

The designer/wizard interprets the actions of the user and controls the responses of the system. The user experiences what the ‘real’ system might be like.

The system may be:
- non-existent
- partially built
- completely functional

Best for certain types of interaction (based on wizard’s reaction time)
Evolution

Rapid prototypes: Early exploration of diverse alternatives
  Easy to create, check, throw away afterwards
  example: paper prototype or interface like SILK

Iterative prototypes: create individual modules
  Create successively more refined versions
  example: series of prototypes, successively more detailed

Evolving prototypes: may become the final product
  Different completed sections are successively added
  example: a software module has functionality added before being added to the final system
Prototyping strategies

Horizontal: complete one layer of functionality at a time
  example: develop the details of the interface without a working database

Vertical: complete functionality of part of the system
  example: develop the spelling checker fiirst

Task: create functionality necessary for a single task
  example: develop the interface for adding and editing an image

Scenario: create functionality needed to run a scenario
  example: develop the functions needed to edit three images and spell-check a document within a design scenario