6. Design

Ideas about Design
(Not a substitute for reading the text)

Interaction Design

- A process:
  - a creative activity
  - a goal-directed problem solving activity
  - a decision-making activity to balance trade-offs
  - informed by intended use, target domain, materials, cost, and feasibility

- A representation:
  - a plan for development
  - a set of alternatives and successive elaborations and justifications
“Four Basic Activities“

1. Identifying needs and establishing requirements
2. Developing alternative designs
   - Many approaches divide this into conceptual and physical design
3. Building interactive versions of the designs ("prototyping")
4. Evaluating designs

“Three Characteristics“

1. Focus on users early and often in the design and evaluation of the artifact (how?)
2. Identify, document and agree on specific usability and user experience goals
3. Iteration is inevitable (driven by evaluation based on the goals)

Practical Issues (discuss)

- Who are the users?
- What counts as a "need"?
- Where do alternatives come from?
- How do you choose among alternatives?
Bill Verplank

“One of the first things you learn in design is to put forward a number of alternatives so that you can then compare them. Having a lot of display space is important for doing this because you can then make them visual. One of the things you can do with visual things is superimpose them, or put them side by side and quite often when you start doing that you like one better than another. Until you’ve made a comparison you have no idea why you prefer one over another. The criteria emerge from the comparison.”

“Part of learning these skills is just looking at creative people, seeing the volume of stuff they do, and realizing the role of chance. ... It’s not just picking the right idea, but recognizing the right idea in all the mess that you produce. Having that rich field of things to compare and contrast that you’ve either generated or collected is something that designers need to know how to produce. Evaluation also comes into brainstorming: when you stop generating ideas you have to start evaluating them.”

Most experts agree ...

Design the functionality before the interface

• Helps to focus on activities that effectively meet users’ needs and goals
• Helps deal with complexity of design: too many choices to make in interface

Levels of design (Moran)

• Conceptual
  - Task: what do the users want to do, or what activity is to be supported?
  - Semantic: what objects, actions and methods are needed?
• Physical
  - Interaction (Operational): how is the task mapped to interactions on I/O devices?
  - Syntactic (Representational): how are the semantics reflected in information displays?
Example Design Methods

A brief introduction to "holistic", scenario-driven, and model-driven methods.

Themes

As we review the methods, consider ...

- Tradeoff between avoiding premature commitment versus connecting with users
- How representations drive the design (and change through the design process)
- Use of abstraction
  - Abstracting from Situations
  - Abstracting from Implementations

Holistic Design

- Influenced by ethnography
  - situation-centered design (immersion)
- Unstructured:
  - levels of design not distinguished
  - no rigorous ordering of design
- Instead
  - Explore conceptual models
  - Do so in context of visual appearance and behavior of interface
Holistic techniques: Sketching

- Visual Brainstorming and Rapid Paper Prototyping
  - use easy to manipulate tools
  - minimize polish and hence commitment
  - explore alternate visual metaphors for conceptual model

1. Initial sketches
2. Cardboard representations evaluated with users
3. Scenarios, prototypes

Holistic techniques: Scenarios

Scenario: personalized fictional story of use
  - Construct variety of scenarios to consider range of users and activities

Snapshot: image showing possible interaction

Storyboard: sequence of snapshots illustrating a scenario
  - can construct these interactively with users
• Scenarios and storyboards help communicate with users, and force designer to confront implications
• Potential danger of premature commitment & confusing functionality with interface

Scenario-based Design  Rosson & Carroll

- A refinement approach to scenarios
- Abstracts from implementation but not situation
Scenarios in Usability Engineering

Stories of people and their activities, sometimes includes computer use, always includes goals

• Typical elements of the story are:
  - a setting
  - one or more actors or agents
  - an orienting or motivating goal or objective
  - mental activity, plans or evaluation of behavior
  - a “storyline” sequenced by actions and events

• Emphasis on use, i.e., people’s needs, expectations, actions, and reactions

Problem Scenario

A problem scenario describes the current situation
Revise until you and the users agree that you understand the situation

(Some other work is done first for requirements analysis)

Activity Scenario

Rewrite the problem scenario to describe the activity you would like to support
No details on how it is supported (abstracts from implementation)
Revise until you and the users agree that you share the same objectives
Information & Interaction Scenarios

Revise activity scenario to describe
- How information is represented
- Sequences of interactions with the system
- How interaction is supported by interface widgets and devices
Often includes sketches, storyboards, and "scenario engines"
Continue until usability tests show that users can interpret the design

Tradeoffs (Rosson)

- Design by definition is invention, creativity
  - never just one approach, never one correct answer
  - but some answers are demonstrably better
- Interactive system design tremendously complex
  - many interdependencies, eg schedule, cost, competitive advantage, local expertise,...
  - users and their needs are one large set of dependencies
- Tradeoffs are useful in analyzing these relations
  - focus on tradeoffs affecting users' experiences
  - guides design thinking, also serves as design rationale

Claims Matrix

SBD uses a claims matrix to evaluate tradeoffs:
- Lists proposed features along with desirable and undesirable aspects of those features
- Claims are derived from and illustrated by the scenarios
- Evolves across many scenarios and ties them together
See http://ucs.cs.vt.edu/ for case library
Usage-Centered Design

- Constantine and Lockwood
- Focus on design for Use, not Users
- Also a refinement approach: don't commit too early!
- But abstracts from situation as well as implementation (uses abstract models)
- Challenge: can users understand the abstractions?

Usage Centered Design Overview

Essential Use Cases (Task Model)

<table>
<thead>
<tr>
<th>User Intent</th>
<th>System Responsibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coach</td>
<td></td>
</tr>
<tr>
<td>Select study-group</td>
<td>Display workspace</td>
</tr>
<tr>
<td>Select assignment</td>
<td></td>
</tr>
<tr>
<td>Make Anonymous Work Visible</td>
<td>Broadcast identifiers, Modify group workspace</td>
</tr>
<tr>
<td>Learner or Coach</td>
<td></td>
</tr>
<tr>
<td>Select study-group</td>
<td>Display workspace</td>
</tr>
<tr>
<td>Select work to upload</td>
<td>Display success status</td>
</tr>
<tr>
<td>Add explanation</td>
<td></td>
</tr>
<tr>
<td>Indicate completion</td>
<td></td>
</tr>
<tr>
<td>Coach</td>
<td></td>
</tr>
<tr>
<td>Enter Study-group</td>
<td>Display workspace</td>
</tr>
<tr>
<td>Select assignment</td>
<td></td>
</tr>
<tr>
<td>Coach</td>
<td></td>
</tr>
<tr>
<td>Edit Work Collection</td>
<td>Display new workspace</td>
</tr>
<tr>
<td>Select study-group</td>
<td>Display workspace</td>
</tr>
<tr>
<td>Make Anonymous Work Visible</td>
<td>Broadcast identifiers, Modify group workspace</td>
</tr>
<tr>
<td>Learner or Coach</td>
<td></td>
</tr>
<tr>
<td>Select study-group</td>
<td>Display workspace</td>
</tr>
<tr>
<td>Select work to upload</td>
<td>Display success status</td>
</tr>
<tr>
<td>Add explanation</td>
<td></td>
</tr>
<tr>
<td>Indicate completion</td>
<td></td>
</tr>
</tbody>
</table>
Mapping Task to Content Model

Navigation Map

Abstract Layout
Usage Centered Design Summary

Abstraction and Accessibility

• The major methods differ on whether their design representations:
  - Abstract from the implementation (scenario-based, model-based)
  - Abstract from the situation of use (model-based, extreme-programming)

• A fundamental trade-off is avoiding premature commitments versus the accessibility of design representations to users.
Representations for Design

“A design is an information base that describes aspects of this object, and the design process can be viewed as successive elaborations of representations, such as adding more information or even backtracking and exploring alternatives.”

What representations are good for supporting this process?

Suitable Representations

- Meet needs of both designers and users
  - Accurate enough
  - Simple enough
  - Makes important issues salient
- Needs change throughout the design process: range of representations
- Many methods and associated representations are available: pick the ones that help you the most

Lifecycle Models

In which HCI meets Software Engineering
“Waterfall” lifecycle

- Requirements analysis
- Design
- Code
- Test
- Maintenance

“no one does this anymore” …?

Spiral Lifecycle Model

- Derived from empirical studies of designers
- Evaluation is central
  - Start anywhere
  - Evaluate
  - Iterate

From cctr.umkc.edu/~kennethjuwang/spiral.htm

Star Model

- Derived from empirical studies of designers
- Evaluation is central
  - Start anywhere
  - Evaluate
  - Iterate
Simple Interaction Design Model

Assignment

Write a "root concept" document for your project
A "root concept" is a shared understanding of the project's high level goals used to guide initial design

- **Vision**: what are we trying to achieve?
- **Rationale**: why will technology help?
- **Stakeholder Groups**: those with vested interest (not just "users")
- **Assumptions and Constraints**: What decisions have we already made? What requirements have been imposed?

About a page