Scenarios and Prototyping

Seminar Mensch-Computer-Interaktion
5.6.2007
Overview

• Scenario-based design
  – examples
  – scenario perspective
  – claims analysis

• Prototyping in human-computer interaction
  – fidelity: low-, medium-, high-fidelity prototyping
  – tools & methods: paper prototypes, sketches & storyboards, Wizard of Oz
Scenario-based Design

- User interaction scenario
  - informal narrative description of user activity and experience when performing a task
  - what a user would have to do and see step-by-step in performing a task using a given system

- Scenarios are design-specific
  - how would a task be performed in a particular design
  - task is design-independent

- Use-oriented perspective
  - represent, analyze, plan impact of system on task
Example Scenario 1

• “Say I want to find a book by George Jeffries. I don't remember the title but I know it was published before 1995. I go to the catalog and enter my user password. I don't understand why I have to do this, since I can't get into the library using the catalog without passing through the security gates. However, once my password has been confirmed, I am given a choice of searching by author or by date, but not the combination of author and date. I tend to choose the author option because the date search usually identifies too many entries. After about 30 seconds the catalog returns saying that there are no entries for George Jeffries and showing me the list of entries closest to the one I've sought. When I see the list, I realize that in fact I got the author's first name wrong and it's Gregory, not George. I choose the entry I want and the system displays the location to tell me where to find the book.”

• Note
  – importance of getting author's name right
  – annoyance of having to enter a password
  – limited search options
  – importance of showing list of similar entries
Example Scenario 2

• “The user types in all the names of the meeting participants together with some constraints such as the length of the meeting, roughly when the meeting needs to take place, and possibly where it needs to take place. The system then checks against the individuals' calendars and the central departmental calendar and presents the user with a series of dates on which everyone is free all at the same time. Then the meeting could be confirmed and written into people's calendars. Some people, though, will want to be asked before the calendar entry is made. Perhaps the system could email them automatically and ask that it be confirmed before it is written in.”

• “A businesswoman is traveling from San Francisco to Paris on a business trip. On her way to the airport she narrowly misses a traffic delay. She avoids the traffic jam because her Smartphone beeps, then sends her a text message warning her of the traffic accident on her normal route from her office to the airport.”

Preece, Rogers, Sharp: Interaction Design, 2002
Scenario Perspective

• Explore design space with low degree of commitment
• Scenario representation
  – text, storyboards of cartoon panels, video mock-ups, scripted prototypes
• Scenario properties
  – concrete and detailed enough to infer design implications
  – supports discussion and design of use
  – describes particular instances

<table>
<thead>
<tr>
<th>The scenario perspective:</th>
<th>The &quot;establishment&quot; view:</th>
</tr>
</thead>
<tbody>
<tr>
<td>concrete descriptions</td>
<td>abstract descriptions</td>
</tr>
<tr>
<td>focus on particular instances</td>
<td>focus on generic types</td>
</tr>
<tr>
<td>work driven</td>
<td>technology driven</td>
</tr>
<tr>
<td>open-ended, fragmentary</td>
<td>complete, exhaustive</td>
</tr>
<tr>
<td>informal, rough, colloquial</td>
<td>formal, rigorous</td>
</tr>
<tr>
<td>envisioned outcomes</td>
<td>specified outcomes</td>
</tr>
</tbody>
</table>

John M. Carroll: Scenario-Based Design, 1997
Claims Analysis

- Capture consequences and tradeoffs of design decisions
- Claims analysis: interrogate and explain scenarios
  - {in a given usage situation, a given design feature}
    - causes {desirable consequences}
    - but also {undesirable consequences}

  **Video Information Claim:** video information
  ...is a very rich, intrinsically appealing medium
  ...but is difficult to search, and must be viewed linearly
  in real time

  - pose elaboration questions about events in a scenario
  - hypotheses about the underlying causal relations in a scenario
  - process of critique and refinement
Scenarios in the Development Lifecycle

- Requirements analysis
  - scenario description of current state
- User-designer communication
  - users create scenarios illustrating important design issues
  - joint creation of overview scenarios and scenarios for particular tasks
- Design rationale
  - rationale explains design with respect to scenarios
  - identify design tradeoffs by contrasting scenarios, e.g. query scenario vs. browsing scenario
- Envisionment
  - scenarios encourage creative thinking by leaving details out
- Software design
  - analyze scenarios to identify central problem domain objects
- Documentation
- Evaluation
- Team building
Prototyping

• Building a scaled-down version of an interactive system to collect information to guide its design
  – invaluable for iterative design
• Get early feedback on emerging designs
  – after initial requirements analysis, scenarios, conceptual design
• Continuous input for design decisions
  – during all design phases
• Prototype appropriate for
  – audience
  – design phase
  – design question

Design Evaluation
Prototype Fidelity
low high
Prototyping for Whom?

- Central role of prototypes in communication and coordination
- Prototypes as artifacts for communication in multi-competence teams / stakeholders
  - designers
  - developers
  - users
  - clients
  - management
- Different team members $\rightarrow$ different roles $\rightarrow$ different activities $\rightarrow$ focus on different aspects of a prototype
Forms of Prototyping

• Low-fidelity prototyping
  – Paper and post-it prototypes
  – Low-fidelity testing session
  – Wizard of Oz
  – Level of fidelity: Ubicomp kitchen

• Medium-fidelity prototyping
  – Facade tools
  – Storyboarding
  – Storyboarding tools: SILK, Denim

• High-fidelity prototyping
  – Interface builder tools
  – High-fidelity vs. low-fidelity prototypes
Prototyping Tool Requirements

• Make information gathering process effective and cheap
  – ease of use
  – fast turn-around
  – flexibility
  – data collection capabilities
  – executable prototypes
  – collaborative design
Low-Fidelity Paper Prototypes

• First prototype, quick and cheap
• Paper and pencil mockup of user interface
  – rough sketches of the main interfaces and dialogs
  – textual description of interface functions and relationships between screens (storyboards)
• Goal: brainstorming, first user feedback
• Example: multi-touch table
  – paper prototype
  – initial system
Low-Fidelity Paper Prototypes

• Building prototypes on paper and testing them with real users
  – construct paper prototype during first few days
  – demonstrate behavior of an interface very early
  – distill lessons from test observations

• Quality as a function of the number of iterations and refinements

• Advantages for low-fidelity prototyping
  – still cheap to make changes
  – maximizes number of design iterations before committing to code
  – allows to try more ideas than with high-fidelity prototypes
Post-It Prototype

• A more interactive form of the paper prototype
  – dialogs, menus, windows as Post-It notes in multiple layers
  – simulation of dynamic interface elements by manipulation of the notes
  – “data” on Post-Its are quickly changeable

• PICTIVE
  – Post-It prototype
  – videotape design sessions

Plastic Interface for Collaborative Technology Initiatives through Video Exploration
Paper/Post-it Prototype Process

1. Collaboratively creating the prototypes
2. Reviewing the prototypes

Source: http://www.pocketpcmag.com/_archives/may03/e_prototyping.asp
Building a Low-Fidelity Prototype

• Assemble material
  – paper: large heavy paper for designs, cards for note taking
  – adhesives: tape, glue sticks, correction tape
  – markers: colored pens and pencils, highlighters, fine liners
  – overhead sheets
  – scissors
Building a Low-Fidelity Prototype

• Set a deadline for meetings with users
  – forced to make preliminary decisions and get them down on paper
  – quality comes through iterative refinement

• Construct models
  – anything that moves, comes and goes, changes appearance, on its own piece of paper
  – easy to play computer

• set up library of empty standard widget images
Paper Prototype Examples

Source: http://www.pocketpcmag.com/_archives/may03/e_prototyping.asp

Deutsche Telekom Laboratories, TU Berlin
Seminar HCI, Michael Rohs
Low-Fidelity Testing

- Select users
- Prepare test scenarios, drawn from task analysis
  - familiar data, realistic tasks
- Practice
  - team members know their roles, no “computer” delays

Marc Rettig: Prototyping for Tiny Fingers
Low-Fidelity Prototype Revision

- Evaluation of test results
  - arrange paper prototype on table
  - pile note cards next to component
- Summarize and prioritize problems
  - written report on findings
- Prototype refinement
  - agenda for meeting to discuss design changes
  - attach post-it notes with changes to each component
Paper Prototype Pros & Cons

• Pros
  – not too detailed, designers / users concentrate on abstract dialog concepts
  – easy to use
  – fast turn-around
  – encourage team design

• Cons
  – difficult to capture interface behavior
  – not executable
Wizard of Oz …

• The first “Chess Computer”

Source: [http://web.onetel.net.uk/~johnramling/turk.html](http://web.onetel.net.uk/~johnramling/turk.html)
… Wizard of Oz

- Method for testing a non-existing system
- Human “wizard” simulates system responses
  - interacts with user via a simulated software user interface
- Useful for adding complex vertical functionality
  - speech and gesture recognition, language translation
Limiting Prototypes

• Vertical prototypes
  – includes in-depth functionality for only a few selected features
  – common design ideas can be tested in depth

• Horizontal prototypes
  – the entire surface interface with no underlying functionality
  – a simulation; no real work can be performed

• Scenario
  – scripts of particular fixed uses of the system; no deviation allowed

Prototyping Example: Ubicomp

• Kitchen-Net
  – locating items in an industrial-sized kitchen
  – responds to spoken queries for items
  – set of screens placed around the kitchen
Kitchen-Net Setup: Low-Fidelity…

• Wizard of Oz
  – simulate voice recognition system
  – simulate system for locating users and items

• Paper prototype
  – paper sketches and post-it notes to represent screens

• Human “computer” updated screens in response to spoken queries and movements
  – implementation time: 1 day; changes: very quickly
… Setup: High-Fidelity

- Interactive prototype
  - Tablet PCs as Kitchen-Net screens, automatically responding to events logged by the wizard
  - no need for human “computer”
  - implementation time: 2 weeks; changes: 1 day
Prototyping Fidelity: Results

• Problems with paper prototype
  – elicited less comments and suggestions
  – paper screens did not convince participants of system usefulness (lack of fidelity)
  – too many human “computers” needed, because of large kitchen area (lack of automation)
  – presence of “computers” affected participants’ interaction (lack of automation)
Facade Prototypes …

• Drawing editors with ability to specify input behavior
  – looks like real application, but no application logic behind
  – switch between screens

• Examples
  – presentation programs like PowerPoint
    • animation features
    • buttons for navigation to specific slides
  – Apple Hypercard
    • WIMP interfaces
    • look and feel level
  – Macromedia Flash, Dreamweaver
    • interfaces with large graphical component
    • interface functionality level
... Facade Prototypes

- **Pros**
  - easy to use
  - fast turn-around
  - reliable feedback
  - executable (with limitations)

- **Cons**
  - do not produce reusable code
  - facade prototype and application might be built by different teams
    - Lessons learned from prototype do not carry over to implementation
Storyboarding

- A series of key frames
  - originally from film; used to get the idea of a scene
  - start of design process with rough pictures of screen layouts
  - snapshots of the interface at particular points in the interaction

- Users can quickly evaluate interaction flow

Source: http://www.pocketpcmag.com/_archives/may03/e_prototyping.asp
Initial screen

Scan the stroller ->

Change color ->

Place the order ->
Scan the shirt ->

Touch previous item ->

Delete that item...

Altern path...
Interactive Storyboards …

- **SILK** (Landay, Myers; CMU)
  - quickly sketch an interface electronically
  - specify storyboard by drawing arrows
  - SILK recognizes arrows and performs screen transitions
- **Storyboards become alive**
  - sequencing between screens can simulate functionality without worrying about implementation

Figure 1: Rotate the rectangle upon button presses.
Figure 2: Make a dialog box appear when the button is pressed.
... Interactive Storyboards

- DENIM
- An Informal Tool For Early Stage Web Site and UI Design

- [http://guir.berkeley.edu/projects/denim/](http://guir.berkeley.edu/projects/denim/)
Interface Builders

- **Interface construction tools**
  - Visual Studio (C++, C#, Basic), Delphi, Eclipse, Together, …

- **Pros**
  - easy to use (static part of interface)
  - fast turn-around (static part of interface)
  - extensive control over interface appearance

- **Cons**
  - only for static part of interface, not for contents of main window
  - forced to commit to concrete widgets too early
High-Fidelity vs. Low-Fidelity Prototypes

- Too long to build and change
  - weeks vs. hours
- Get feedback on „fit and finish“ issues but need feedback on „big things“
  - expressiveness and power of basic metaphor, flow of conversation, terminology
- Developers resist change
- High-fidelity prototype may set expectations that are hard to meet
  - “... you are almost done …”
## Overview Tools: Pros/Cons

<table>
<thead>
<tr>
<th>Feature</th>
<th>Paper</th>
<th>Facade Tools</th>
<th>Interface Builders</th>
<th>Actual Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ease of Use</td>
<td>++</td>
<td>+</td>
<td>0</td>
<td>--</td>
</tr>
<tr>
<td>Fast Turnaround</td>
<td>++</td>
<td>++</td>
<td>+</td>
<td>--</td>
</tr>
<tr>
<td>Flexibility and Control</td>
<td>++</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Executable Prototypes</td>
<td>--</td>
<td>+</td>
<td>0</td>
<td>++</td>
</tr>
<tr>
<td>Team Design</td>
<td>+</td>
<td>--</td>
<td>--</td>
<td>0</td>
</tr>
</tbody>
</table>
Summary

• Scenarios
  – informal stories of user activity and experience
  – useful for reasoning about design without early commitment

• Prototyping
  – feedback on design ideas
  – low-fidelity vs. medium-fidelity vs. high-fidelity
  – many approaches, methods and tools
  – storyboarding and sketching
Quellen

• Preece, Rogers, Sharp: Interaction Design – Beyond Human-Computer Interaction, John Wiley & Sons, 2002. – Chapter 8


• Wikipedia: Software Prototyping