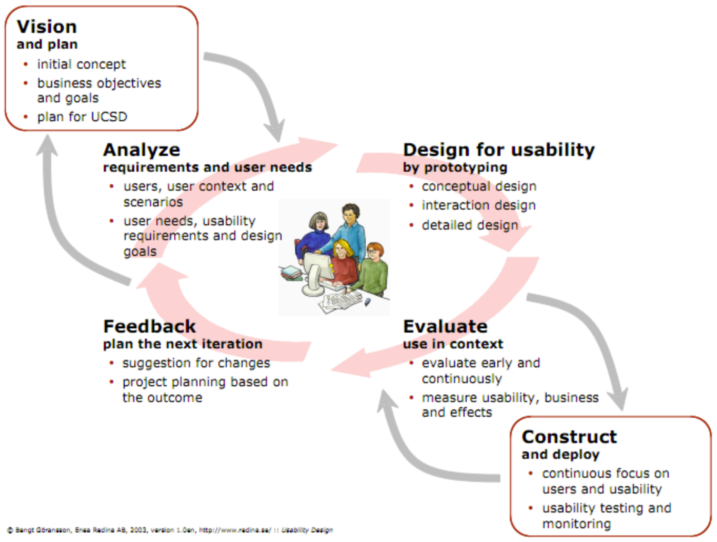
PROTOTYPING

Cristian Bogdan
cristi@kth.se

A UCSD process framework


Vision
and plan
initial concept
business objectives and goals
plan for UCSD
Analyze
requirements and user needs
users, user context and scenarios
user needs, usability requirements and design goals
Design for usability
by prototyping
conceptual design
interaction design
detailed design
Feedback
plan the next iteration
suggestion for changes
project planning based on the outcome
Evaluate
use in context
evaluate early and continuously
measure usability, business and effects
Construct
and deploy
continuous focus on users and usability
usability testing and monitoring

© Bengt Göransson, Enea Redina AB, 2003, version 1.0en, http://www.redina.se/ :: *Usability Design*

What is a prototype?

- Concrete representation of an interactive system/service, or *relevant* part of it
- Tangible artifact
- Relevance depends on what is being explored right now

Prototypes and disciplines

- Architecture: scaled-down model
- Fashion: one of a kind dress
- Computer Engineering: feasibility of a technical process
- Design: express ideas and reflect on them

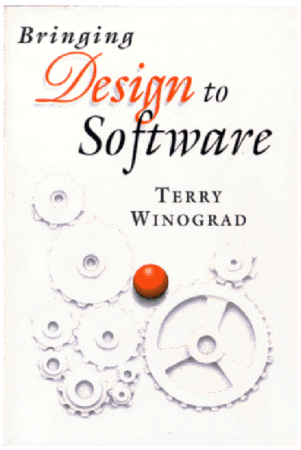
When I prototype…

1. Should I be fast and frugal?
2. Should I be thorough and express all details?

Why prototype?

- It's cheap(er)
- It's fast(er)
- It's easy
  - Can focus on the design issues rather than the technique/technology
- It allows exploration
- Reflective conversation with materials
- It's involving
- It's provocative (brings feedback)
- It's concrete (shared understanding)
  - Uncover misunderstandings early

Reflective conversation with materials



- Winograd, 1996
- There is no direct path between a designer's intention and the outcome
  - Architecture student sketching on paper
  - Structural engineering student working with computer simulation
- Reflection in action (Schon)
- The Apple power switch (Norman)

Where is the focus?

1. On the idea prototyped?
2. Both on the idea and on learning more about the tool?

Prototype purpose and represenatation

- The purpose varies a lot
- Depending on what is being explored right now
- Consider the purpose at each stage
- Choose the most appropriate *representation* for that purpose

Prototype classifications

1. **Representation**: off-line vs on-line
2. **Precision** (low or high, also referred to as fidelity)
3. **Interactivity** (the "look" only, or also parts of the "feel")
4. **Evolution**: rapid/throw-away, iterative, evolutionary

1. Representation

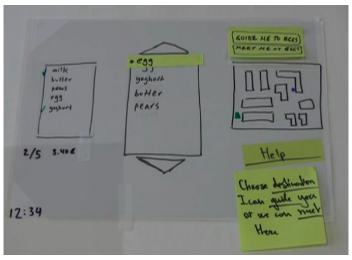
- Offline
  - no need for a computer, or code
  - Paper sketches, storyboards, cardboard mock-ups, videos
  - Early, quick, throw-away
- Online (software)
  - Computer animation, interactive video presentation, scripting, interface builder
  - Higher cost and skill
  - Later stages

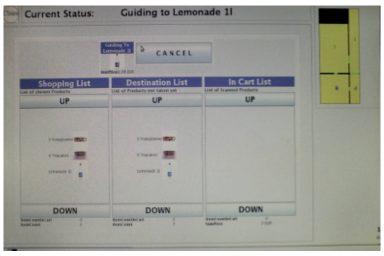
Offline prototypes

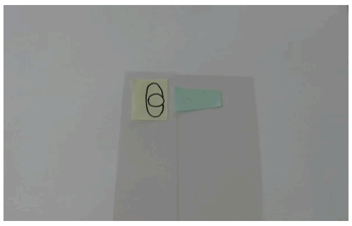
- Rapid iteration and exploration
- Prevents falling in love with first solution
- No intermediary between idea and implementation
- Less likely to constrain thinking due to the programming environment used
- A wide range of people can participate
  - Increase participation and communication

Offline rapid prototypes

- Paper and pencil
- Mock-up
- Wizard of Oz
- Video prototyping



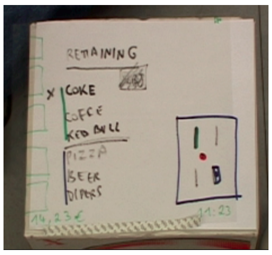




Adapting the prototyping material

Design of a Robotic Shopping Trolley

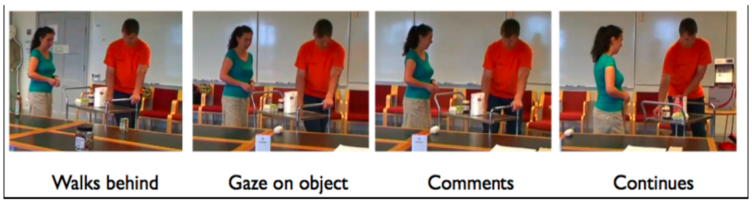
Multidisciplinary enactment
• (compare with requirement specification meeting)
• User, touchscreen interface
• Robot engine
• Communication system



Design and enactment by users
The two time evolutions
• robot movement
• touchscreen interface

Study on movement as communication KTH/TUW

- Can robot movement help improve human-robot communication?
- Participatory, overt "Wizard-of-Oz" (with 12 users)
  - –Enactment and reflection with openly simulated robots
  - –One person acts as robot motor
  - –Spoken prompts generated with off-the-shelf text-to-speech



When to do online prototypes?

- Beaudoin-Lafon and Mackay
  - Programmers often argue in favor of software prototypes, even at the earliest stages of design. Because they are familiar with programming languages, programmers believe it will be faster and more useful to write code rather than “waste time” creating paper prototypes

When to do online prototypes?

- In 20 years of prototyping, we have yet to find a situation where this is true

Programmers arguments

- Familiarity with programming language
  - May be, but are you fast?
  - Will you have time to reflect on the design, rather than think of your programming language and environment?
  - Will others in the team be able to use this language or can you do all the work?
    - Most people have a minimum set of drawing skills

Programmers arguments

- Re-use of code (evolutionary)
  - Not "waste time"
- But
  - Design is not a linear process
  - "Plan to throw one away" (Fred Brooks 1975!)
  - Focus on the aspects and *constraints* relevant for design
  - Not on constraints imposed by the programming language/platform

Rapid online prototypes

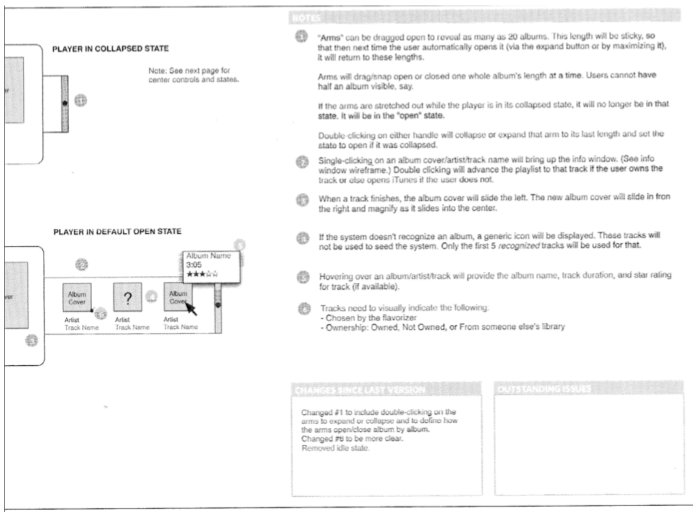
- Higher precision than offline
- Some dynamic interactions are difficult to visualize offline, but easier to animate
- *Non-interactive simulations*
  - A movie of some kind.
  - Macromedia Director, or Flash
    - Scenes can be paper or computer-drawn, screenshots
  - But also PowerPoint, drawing programs (Illustrator, Photoshop),
    - Use of layers is useful to describe different phases
  - *Manual simulation*: hide and show layers, change slides

Rapid online prototypes

- Interactive simulations
  - Typically result in fixed-path interactive prototypes
  - Photoshop layers can also be used for interaction
  - Hypercard, the card metaphor
  - Director/Flash: behaviors attached to symbols
  - PowerPoint: Action Settings/Hyperlink
  - HTML hyperlinks
  - **Wireframe** technologies (e.g. Axure)
  - Scripting languages: can be cryptic but rapid (e.g. Tcl/Tk. Not strongly typed, ignore non-fatal errors
    - button.dialogbox.ok -text OK -command {destroy.dialogbox}

Click Me to go to slide 8

Wireframes


PLAYER IN COLLAPSED STATE
Note: See next page for center controls and states.
PLAYER IN DEFAULT OPEN STATE
Album Name
3:05
★★★☆☆
Album Cover
?
Album Cover
Artist
Track Name
Artist
Track Name
Artist
Track Name

NOTES

1. "Arms" can be dragged open to reveal as many as 20 albums. This length will be sticky, so that then next time the user automatically opens it (via the expand button or by maximizing it), it will return to these lengths.

   Arms will drag/snap open or closed one whole album's length at a time. Users cannot have half an album visible, say.

   If the arms are stretched out while the player is in its collapsed state, it will no longer be in that state. It will be in the "open" state.

   Double-clicking on either handle will collapse or expand that arm to its last length and set the state to open if it was collapsed.

2. Single-clicking on an album cover/artist/track name will bring up the info window. (See info window wireframe.) Double clicking will advance the playlist to that track if the user owns the track or else opens iTunes if the user does not.

3. When a track finishes, the album cover will slide the left. The new album cover will slide in fron the right and magnify as it slides into the center.

4. If the system doesn't recognize an album, a generic icon will be displayed. These tracks will not be used to seed the system. Only the first 5 *recognized* tracks will be used for that.

5. Hovering over an album/artist/track will provide the album name, track duration, and star rating for track (if available).

6. Tracks need to visually indicate the following:
   - Chosen by the flavorizer
   - Ownership: Owned, Not Owned, or From someone else's library

CHANGES SINCE LAST VERSION

Changed #1 to include double-clicking on the arms to expand or collapse and to define how the arms open/close album by album.
Changed #6 to be more clear.
Removed idle state.

OUTSTANDING ISSUES

2. Precision

- Relevance of details with respect to the prototype purpose
- Precision can vary within the same prototype to express what the focus is
  – Hard to do with online prototypes
  – Sketchy widgets (e.g. Balsamiq) help but not all the way
- Tension:
  – what the prototype states (relevant details)
  – what the prototype leaves open (irrelevant details)
- Sketches will always bring more feedback

3. Interactivity

- Interactive low-precision? Offline?
- Enactment, role-playing
- Levels of interaction
  - Fixed prototypes
  - Fixed path prototypes (limited interaction)
  - Open prototypes (large sets of interaction)
- Illustrating possible interactions is different from interactivity!

4. Evolution

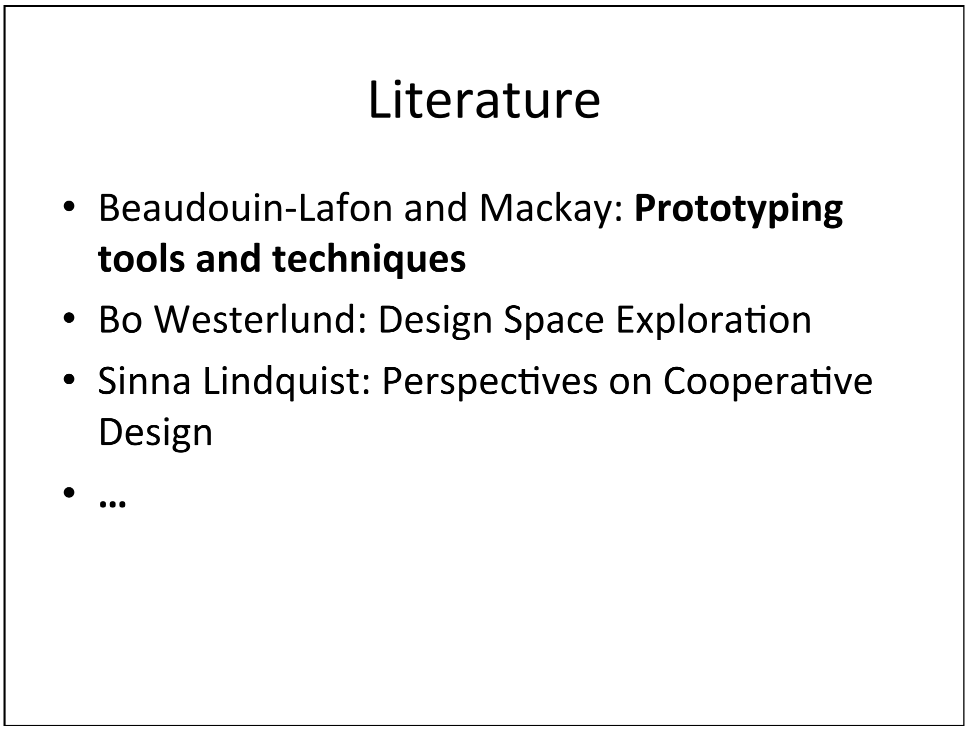
- Rapid
  - Cheap, easy, exploring
  - Online of offline
- Iterative
  - Iterate to vary a theme
  - Iterate to increase precision
- Evolutionary
  - Iterative prototypes that evolve into the final system (or part of it)
  - Extreme programming, agile methodologies

Prototyping strategies

- Horizontal prototypes
  – Cover an entire layer of the design, iterative, increasing precision
- Vertical prototypes
  - Assess the feasibility of a feature down to the lower system layers
- Task-oriented prototypes
- Scenario-based prototypes

Conclusions

- Prototype with whatever material/language/environment allows you and your team to be fast and focus on the design, not implementation details
  – Exception: vertical prototypes where you investigate technology limitations and their UI addressing
- Consider how the design ideas will be communicated to a prototype developer who is not a design team member. Offline prototype?
- Evolutionary prototypes may seem attractive but are often not good to start with



Literature

- Beaudouin-Lafon and Mackay: **Prototyping tools and techniques**
- Bo Westerlund: Design Space Exploration
- Sinna Lindquist: Perspectives on Cooperative Design
- …