

Human Computer Interaction

PROTOTYPING

By: Nguyễn Công Hoan

Reference

- Donald Norman, **The Design of Everyday Things**, MIT Press, 23 Dec 2013
- **Tutorial Teaching** of Prof. Dr. Keith Andrews, Graz University of Technology

PROTOTYPING

“ There’s a mantra at IDEO: “Never go to a meeting without a prototype.” At whatever stage of development, one week, one month, or 6 months. ”

[Tim Brown, President, IDEO, speaking at CHI 2004 in Vienna.]

Perform usability evaluation and obtain feedback as *early* as possible in the design cycle by building and evaluating prototypes. Finally, throw prototypes away and implement final design.

Agenda

- Types of Prototype
- Verbal Prototype
- Low-Fidelity Paper Prototypes
- High-Fidelity Paper Prototypes
- Interactive Sketches
- Working Prototypes
- Computer prototype
- Implementation

Types of Prototype

In increasing order of complexity:

- **Verbal Prototypes:** textual description of choices and results.
- **Paper Prototypes:**
 - Low-Fidelity: hand-drawn sketches.
 - High-Fidelity: more elaborate printouts.
- **Interactive Sketches:** interactive composition of hand-drawn sketches.
- **Working Prototypes:** interactive, skeleton implementation.
- **Computer prototypes:** more interactive

Then, throw prototypes away and implement final design.

Verbal Prototype

- Simple textual description of choices and results.

Low-Fidelity Paper Prototypes

- Paper prototypes simulate screen and dialogue elements on paper.
- First hand-drawn sketches (lo-fi), later perhaps more elaborate printouts (hi-fi).
- Early usability feedback with throwaway designs: *maximum feedback for minimum effort!*
- Greeking (drawing squiggly lines) is used to represent text which would otherwise be a distraction.

Low-Fidelity Paper Prototypes (cont.)

The paper prototype is divided into three main sections: a top navigation bar, a central song archive, and a bottom playlist.

Top Navigation Bar:

- Logo: "IICM on AIR home"
- Login: "LOGIN" with a text input field, "PASSWORD" with a text input field, and "SUBMIT" and "CANCEL" buttons.
- Status: "Status: running"
- Search: "search song" button
- Playlist: "PLAYLIST" button
- Uploading: "uploading Songs" button
- Actual Song: "actual song:" with a wavy line representing a song title.
- Request: "request song" button
- History: "Song history" button

Central Song Archive:

A table with the following headers: "#", "Songname", "Duration", "Rating:", "good", "vote!", and "Request".

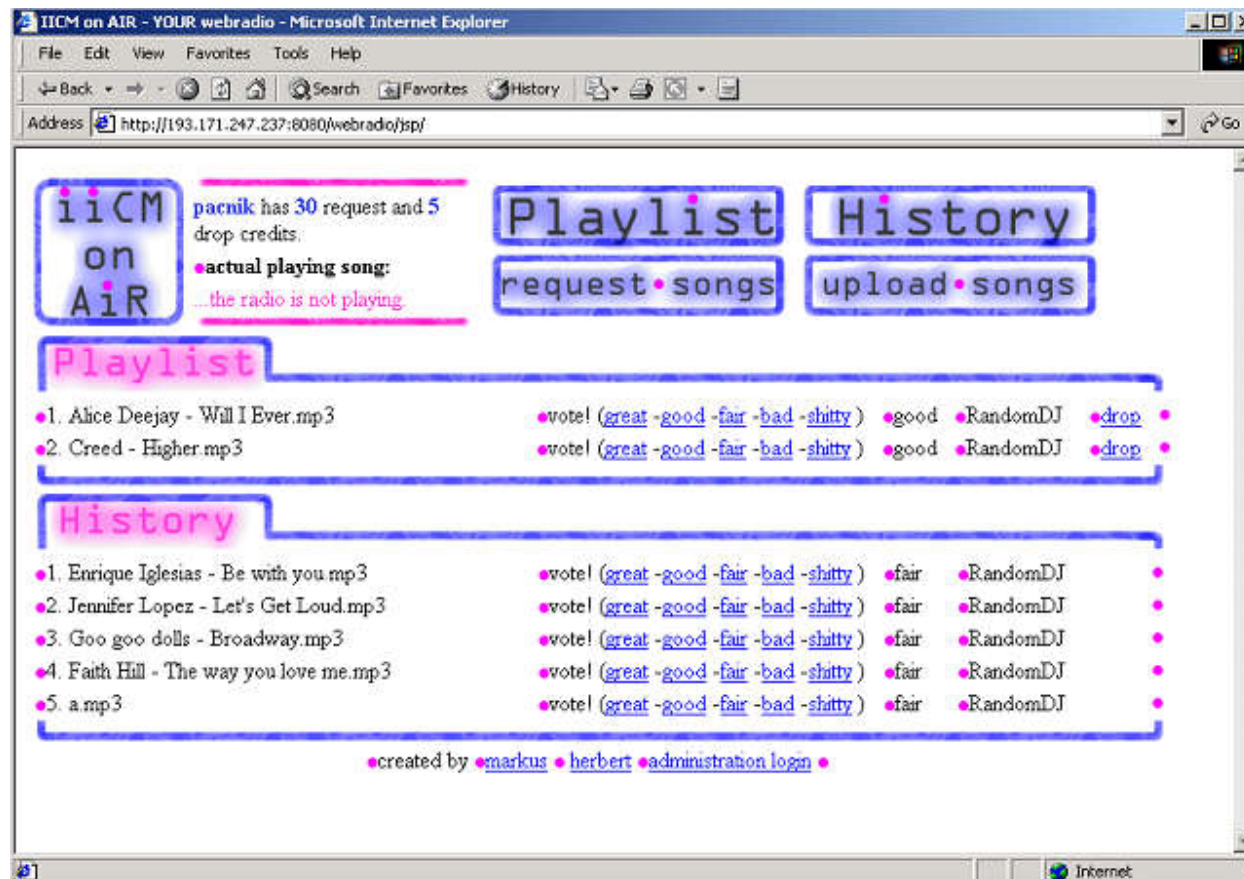
A dropdown menu is shown below the "Rating:" header with the following options: "great", "good", "fair", "bad", and "shitty".

Bottom Playlist:

A list of five numbered items (1. to 5.) with corresponding horizontal lines for song details.

Paper prototype for an online radio station.

Low-Fidelity Paper Prototypes (cont.)



Working prototype for an online radio station.

High-Fidelity Paper Prototypes

- Elaborate screen designs created with drawing editors such as Adobe Illustrator or Corel Draw.
- Printed out in colour.
- They often look too much like a finished design, and not enough like a prototype.
- Users tend to comment on the choice of fonts and colours, rather than the flow through the application.

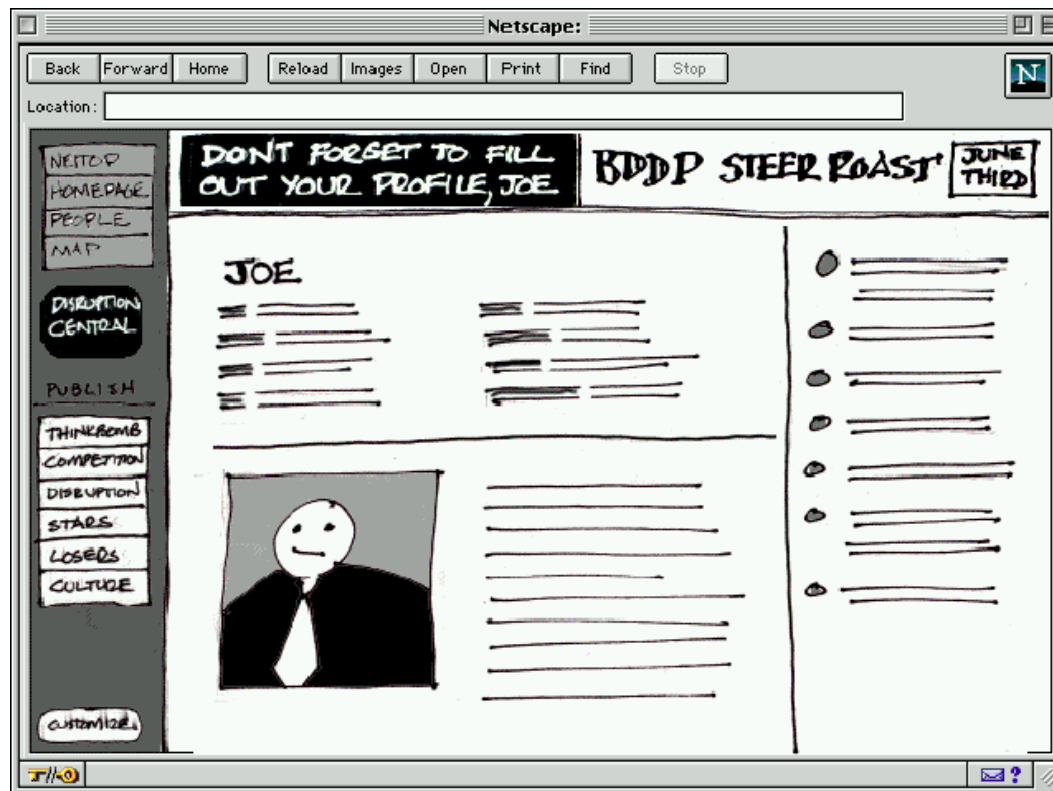
Tips for good paper prototypes

- Make it large
- Make it monochrome (single color)
- Use description where necessary
- You cannot represent tricky interactions like drag & drop,
- animation, progress bar
- Keep pieces organized
- Use folders and envelopes
- Produce multiple alternatives
- Better to get feedback

Interactive Sketches

- Scan in hand-drawn interface sketches.
- Assemble interactive prototype with clickable elements (say with Macromedia Director).
- Retains throwaway, casual look to encourage criticism and discussion

Interactive Sketches (cont.)



An interactive sketch made in Shockwave. Screen designs sketches are scanned and assembled into an interactive prototype with Macromedia Director

Working Prototypes

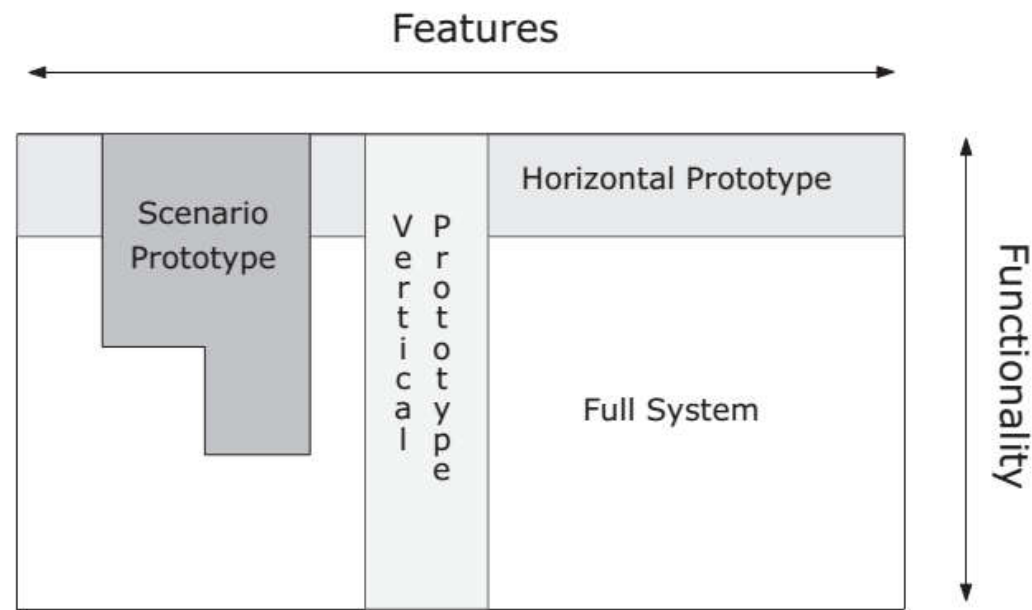
- *Simple algorithms*: ignore special cases.
- *Fake data*: similar data, images instead of video, etc.
- *Wizard of Oz*: human expert operating behind the scenes to simulate interface responses.

Dimensions of Working Prototypes

Working prototypes cut down on either the number of features, or the depth of functionality of features:

- *Vertical Prototype*: in-depth functionality for a few selected features
- *Horizontal Prototype*: full interface features, but no underlying functionality.
- *Scenario Prototype*: only features and functionality along the specific scenarios or paths through the interface which are to be evaluated.

Dimensions of Working Prototypes (cont.)



Working prototypes vary according to the breadth or depth of features implemented.

Computer prototype

- Interactive software simulation
- High-fidelity in look & feel
- Low-fidelity in depth
- May be no backend, covering horizontally
- Does not have a human simulating the backend like paper prototype

Advantages of computer prototype

- Faster than coding
- No debugging
- Easier to change and throw away
- Separate UI design ideas from what offered by UI
- Toolkit (e.g., Visual Studios, C++ Builder)
- Your thinking is not limited to available widgets
- Non-programmers can do it

Computer prototyping techniques

- Storyboard
 - Sequence of painted screenshots, sometimes connected with links
- Form builder
 - Creating real windows with widgets such as buttons, windows, labels, etc.)
- Wizard of Oz
 - Computer frontend, human backend

Storyboarding tools

- Pencil Project
- Photoshop
- Balsamiq Mockup
- Mockingbird
- Excel
- Visio
- Etc.

Storyboarding tools

- Pros
 - You can draw anything
 - Fast
- Cons
 - No interaction
 - No text entry
 - Widgets aren't active

Computer prototyping techniques (cont.)

- Form builders
- FlexBuilder
- Silverlight
- Visual Basic
- C++ Builder
- Visual C#
- Qt Designer

Form builders

- Pros
 - Actual controls ➔ high-fidelity in terms of look
 - You can reuse the design for implementation ➔ save effort from doing again
- Cons
 - Limits thinking to standard and available widgets
 - Content in each widget is not visible

Wizard of Oz

- “Wizard of Oz” = “man behind the curtain”
- Software simulation with human in the loop to help
 - Human “wizard” mimics computational functionalities
 - system response interprets user input
 - controls computer to simulate appropriate output
 - Wizard is not always hidden
- Example
 - Simulate the speech recognition which is not available (human is needed to recognize speech)
- Faking the interaction

Implementation

- Implement final design.
- Competitive analysis of software components:
 - Use existing interface framework as far as possible (Motif, MS-Windows, Java Swing) – saves a *lot* of work.
 - Use existing components and applications rather than re-inventing the wheel.