

Low-fidelity Prototyping

CS 160

Professor John Canny

2/8/2006

1

Outline

- 📄 Review of project ideas
- 📄 Low-fidelity prototyping/WoZ technique
- 📄 User Testing
- 📄 Sketching user interfaces electronically

Top 10(+1) project ideas that didn't make it

Some very creative project ideas (one from each group) that you may not see again, at least not in this course...

Restaurant Pre-order

Don't wait to arrive at the restaurant, download the menu and order before you get there...
(G10)

Carpool Co-ordinator

Carpools often require last-minute coordination, and route re-planning. A good location-based mobile app. (G6)

Soccer Mom Alert

"Practice is Over" alert. Another useful LBS application (G11)

Party Finder

Another hard coordination task is hooking up with friends on the weekend at multiple parties - people start at one, end up somewhere else - great for LBS on mobile devices. (G3)

Your boss is coming...

Usually people think about managers tracking employees, but the opposite is just as useful. Boss tracking can enhance sense of security and reduce stress at work... (G9)

Location-Based Coupons

Coupon capture with Bluetooth is a great way for vendors to track user behavior, and users get to control privacy by deciding whether to use them... (G2)

Hum-based music retrieval

If you remember the tune, but not the song name or performer, hum it to search. Mobile phone lets you do this whenever you remember the tune. (G8)

"I'm Feeling Hungry"

Rather than tracking your meals, this app orders a random meal and surprises you. Very good way add balance and variety to a diet with a simple design. (G1)

Pet Interpreter

About 60% of American households contain a non-English speaker...

because they are a cat or a dog. Dogs and cats make distinctive sounds and use posture to communicate, so you could build an "interpreter" (G7).

* There is at least one device on the market, the "Bow-lingual" that claims to "translate" 178 dog barks.

Wingman/Lose-a-Creep Service

Unfortunately, dating and bar-hopping can be hazardous (especially if your mobile dating/matching app. goes wrong). This service finds local guardians who will help drive nightmare dates away... (G4)

Animal Safari

Language learning is a great application in developing regions like India and Africa. Game-like design can make learning fun and compelling. In Safari Game: (G5)

- (a) Users take pictures of animals, and say or type their name into the phone.
- (b) Users run away from animals while describing them: huge teeth, thunderous growl, 50 mph speed bursts...

Outline

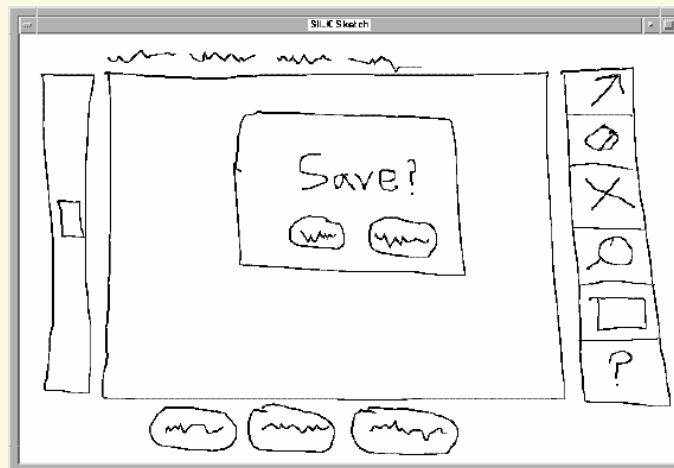
- 📄 Review of project ideas
- 📄 Low-fidelity prototyping/WoZ technique
- 📄 User Testing
- 📄 Sketching user interfaces electronically

Why Do We Prototype?

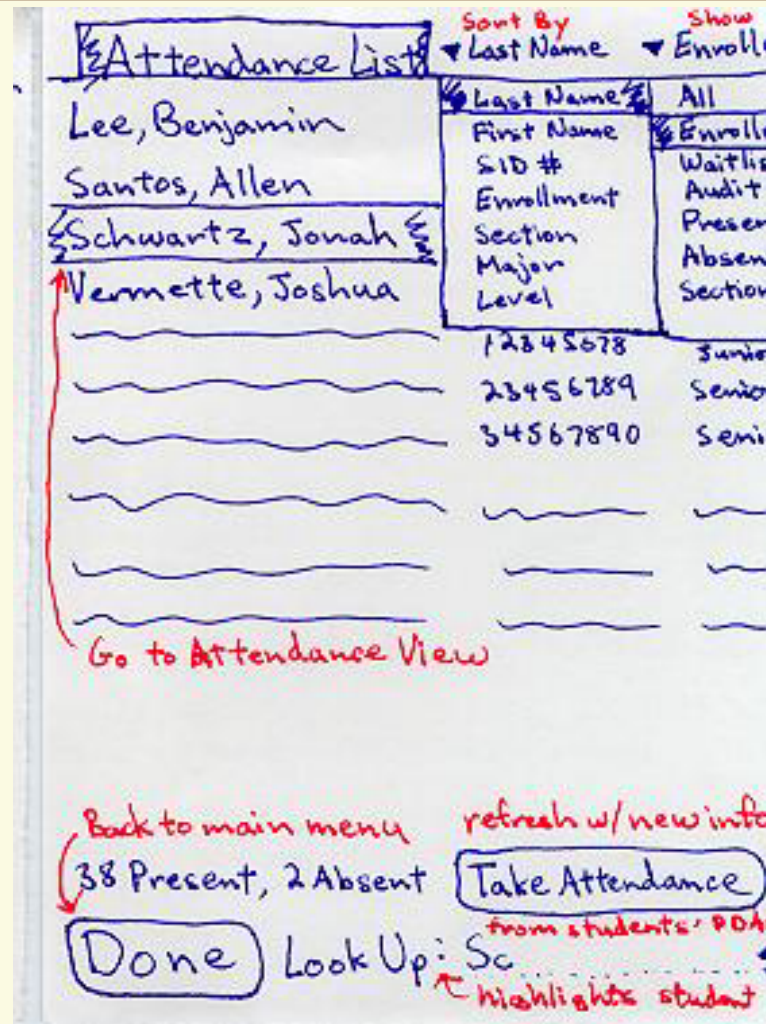
- 📄 Get feedback on our design faster
- 📄 Experiment with alternative designs
- 📄 Fix problems before code is written
- 📄 Keep the design centered on the user

Fidelity in Prototyping



- 📄 Fidelity refers to the level of detail
- 📄 High fidelity
 - * Prototypes look like the final product
- 📄 Low fidelity
 - * Look like a sketch with many details missing



Low-fidelity Sketches

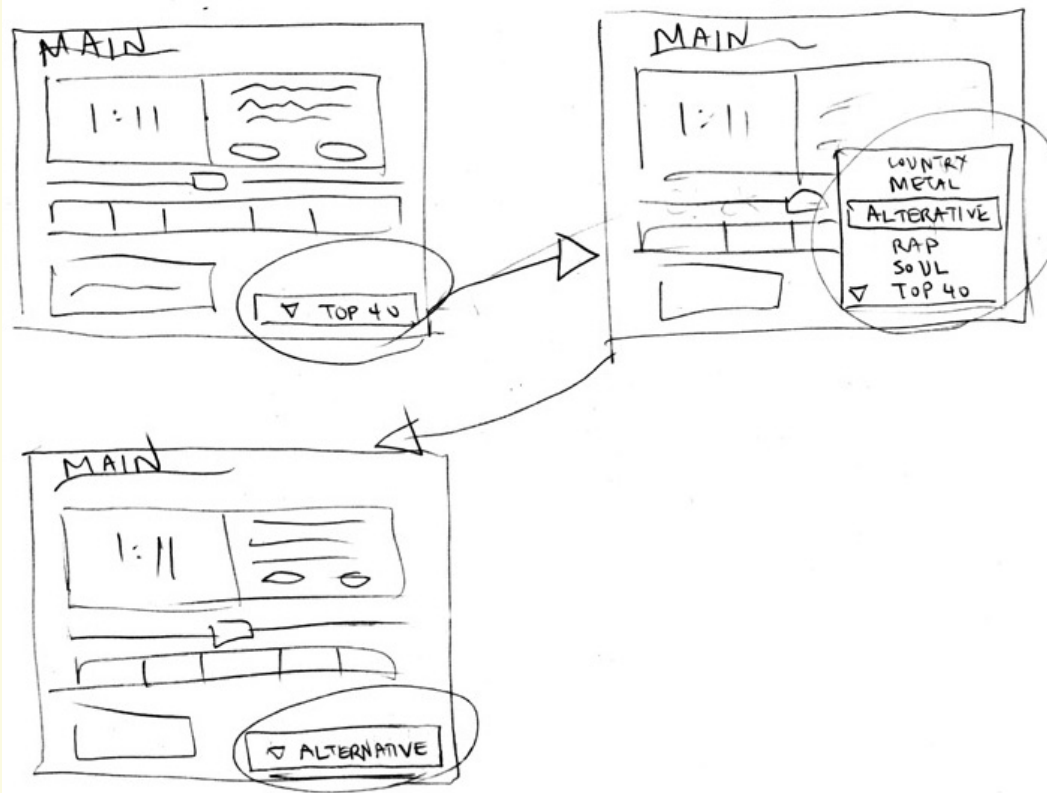


Scenarios

-  For the current assignment, you will create some *scenarios*.
-  A scenario is a *particular sequence of steps* that achieves a task.

Low-fidelity Scenario

SCENARIO 1 "I want to listen to alternative music"



Low-fi Storyboards

- 📄 Where do storyboards come from?
 - * Film & animation
- 📄 Give you a "script" of important events
 - * Scene changes and important story events
- 📄 In UI design, the storyboard is non-linear to support user action choices.

- 📄 **Warning:** You can also "storyboard" a linear scenario, and this term is used somewhat inconsistently.

Storyboards

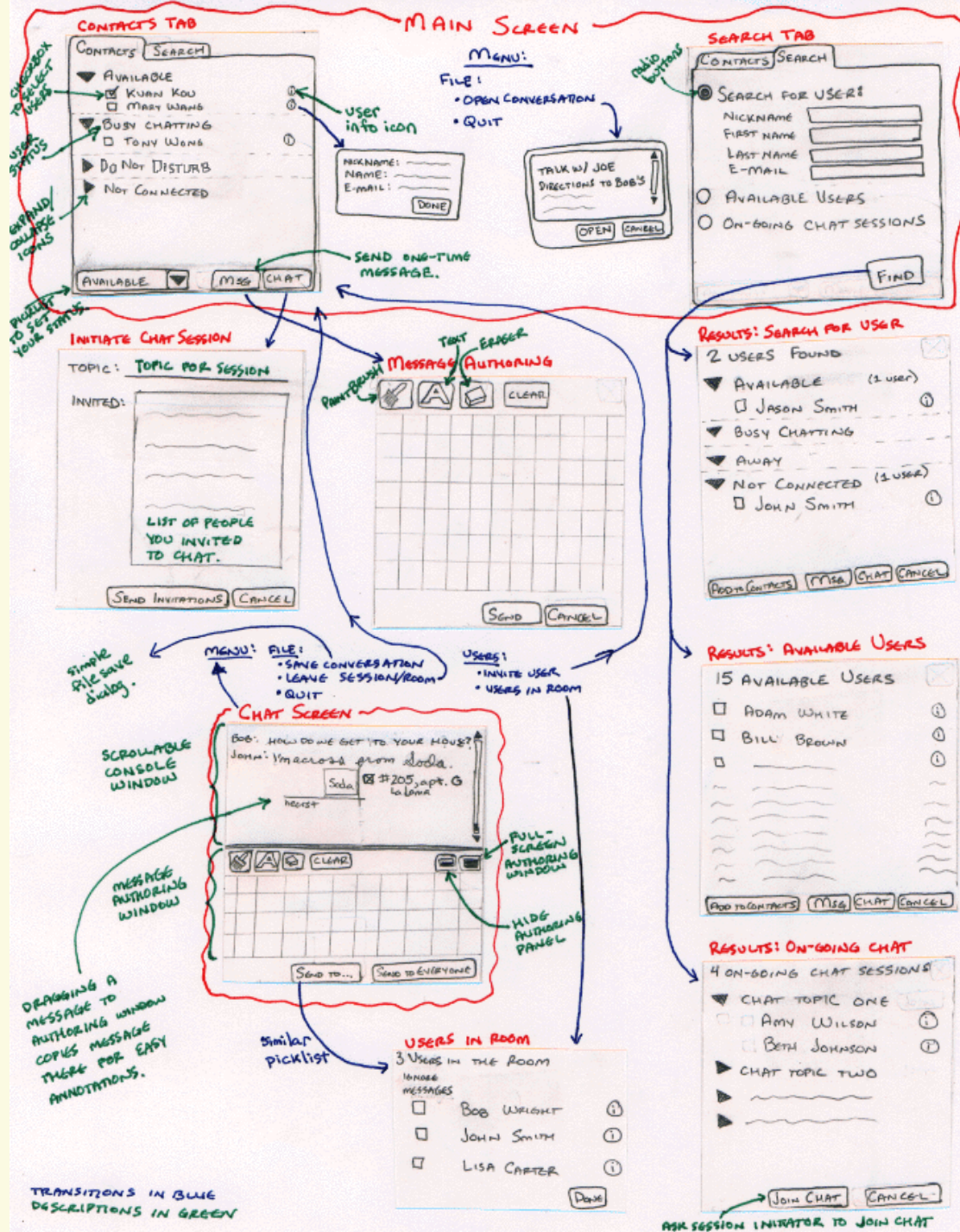
Color Coding

📄 Black: page content

📄 Red: page titles

📄 Green: annotations

📄 Blue: links



Low-fi Prototype Advantages

- ☞ Traditional methods take too long
- ☞ Can *simulate* the prototype
 - * Cards + sketches = prototypes
 - * Designer + WoZ method = simulator
- ☞ Kindergarten implementation skills
 - * Allows non-programmers to participate in the design process - participatory design

Hi-fi Prototype Dis-Advantages:

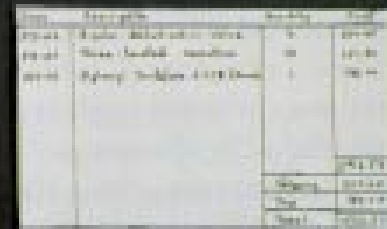
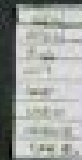
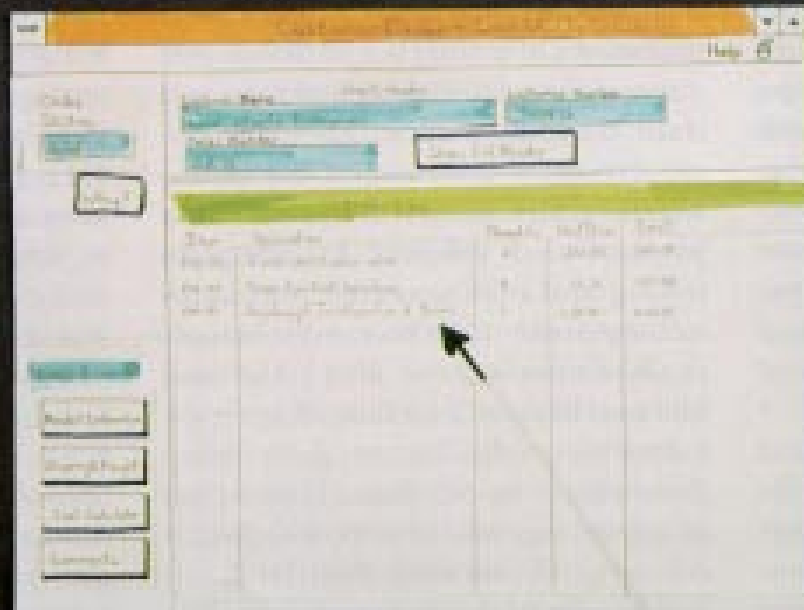
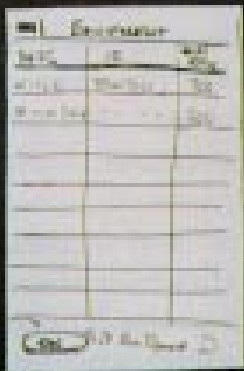
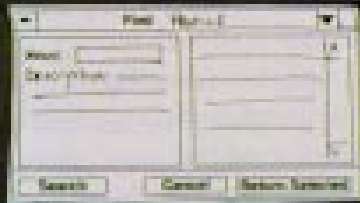
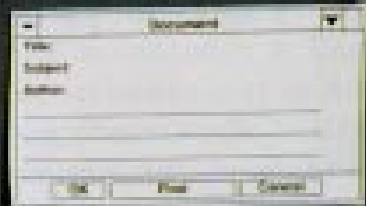
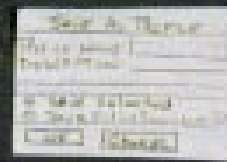
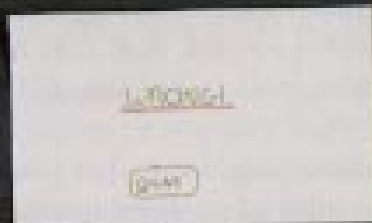
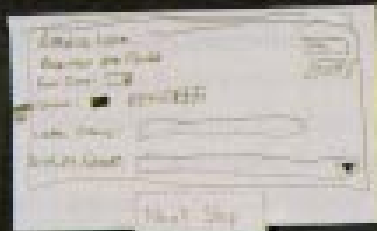
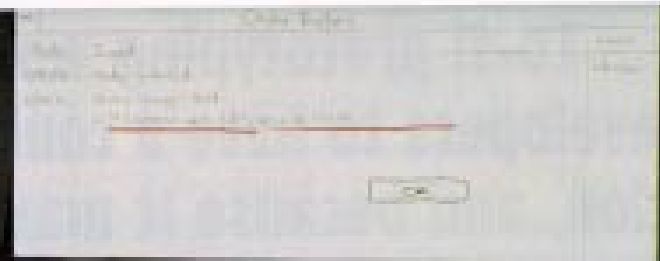
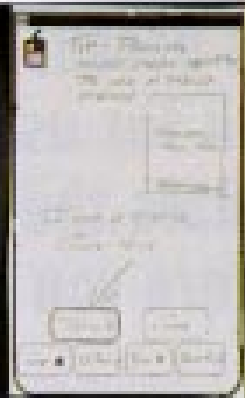
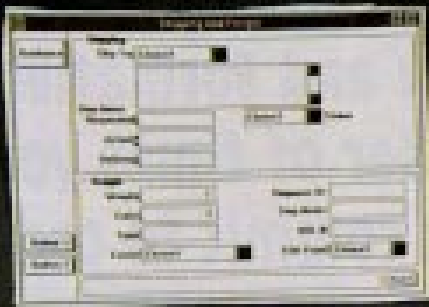
- 📄 Distort perceptions of the tester
 - * Formal representation indicates "finished" nature
 - * People comment on color, fonts, and alignment
- 📄 Discourages major changes
 - * Testers don't want to change a "finished" design
- 📄 Time is lost on details
 - * Discussion tends to be swallowed up on details, not the big-picture issues that matter most.

Hi-fi Prototype Dis-Advantages:


- Using hi-fi tools, you need to specify a string, the font, the style, the size, etc.
- Black found designers will push one design too far -- tunnel vision
- Testers focus on unimportant low-level details
- Testers give more "useful" comments on sketches than on finished-looking interfaces
- 70 seconds for ONE screen in a design of many screens**

Lo-Fi Prototyping Basic Materials

- 📄 Large, heavy, white paper (11 x 17)
- 📄 5x8 in. index cards
- 📄 Tape, stick glue, correction tape
- 📄 Pens & markers (many colors & sizes)
- 📄 Overhead transparencies (for small items)
- 📄 Scissors, X-acto knives, etc.
- 📄 Sources: Office Depot, "The Art Store", ...



Constructing the Model

- 📄 Draw a window frame on large paper
- 📄 Put different screen regions on cards
 - * Anything that moves, changes, appears or disappears
- 📄 Use greek-ing for text if needed 
 - * Squiggles stand for text not written yet
- 📄 Use photocopier to make many versions

Prepare Scenarios

- 📄 Prepare scenarios for each task you're going to support.
- 📄 Check that the prototype supports at least these.
- 📄 Practice, but don't assume the user executes perfectly.

Making it interactive

- ☞ For Wizard-of-Oz simulation, you need:
- ☞ Every menu and dialog for the tasks
- ☞ All the buttons a user might press
 - * Remember help and cancel functions
- ☞ Practice! You're supposed to be simulating a 2 GHz computer, so avoid long "thinking" pauses when you do the experiment, and especially "bugs" in the system.

Wizard of Oz Tips

Rehearse your actions

- * For a complicated UI, make a flowchart which is hidden from the user
- * Make list of legal words for a speech interface

Stay "in role"




- * You are a computer, and have no common sense, or ability to understand spoken English.

Facilitator can remind user of the rules if the user gets stuck

Outline

- 📄 Review of project ideas
- 📄 Low-fidelity prototyping/WoZ technique
- 📄 User Testing
- 📄 Sketching user interfaces electronically

Preparing for a Test

-  Create "informed consent" forms:
-  Explain who you are, and what the goal of your experiment is.
-  Include phrases like:
 - * This is not part of your (class or job) evaluation
 - * You can quit at any time
 - * Your name will not be included in any reports
 - * Identifying data will be anonymized
 - * Participation is voluntary...

Conducting a Test

Four testers (minimum)

- * Greeter - puts users at ease & gets data
- * Facilitator - only team member who speaks
 - + gives instructions & encourages thoughts, opinions
- * Computer - knows application logic & controls it
 - + always simulates the response, w/o explanation
- * Observers - take notes & recommendations

Typical session is 1 hour

- * Preparation, the test, debriefing

Conducting a Test (cont.)


Greet

- * Get forms filled, assure confidentiality, etc.




Test

- * Facilitator hands written tasks to the user
- * Facilitator keeps getting output from participant
 - + "What are you thinking right now?", "Think aloud"
- * Observers record what happens
 - + Avoid strong reactions: shaking head, frowning, laughing, impatience - biases the test

Conducting a Test (cont.)

-  Debrief afterwards: Do a post-evaluation questionnaire
 - * Ask questions about parts you saw problems on
 - * Gather impressions
 - * Get suggestions for improvements/new features
 - * Give thanks

Evaluating Results

-  Sort & prioritize observations
 - * What was important?
 - * What parts of the design had problems?
-  Create a written report on findings
 - * Gives agenda for meeting on design changes
-  Make changes & iterate

Break

Outline

- 📄 Review of project ideas
- 📄 Low-fidelity prototyping/WoZ technique
- 📄 User Testing
- 📄 Sketching user interfaces electronically

Research on UI Design tools: Design Exploration

Brainstorming

- * Put promising design ideas in a tangible form
- * Consider/test design ideas rapidly

Incomplete designs

- * You don't need to cover all cases
- * Support important tasks only

Present several designs to client

- * Early feedback
- * Helps clarify their needs

Goal of Research in Informal UI Design Tools

- 📄 Allow designers to
 - * Quickly sketch interface ideas
 - * Test these ideas with users
 - * Transform to a more finished design without reprogramming

Drawbacks of Current Tools

Examples:

- * Visual Studio
- * Java IDEs
- * Viseo
- * Flash

Drawbacks of Current Tools

- 📄 Require specification of lots of detail
 - * They are hi-fi so require lots of choices:
e.g., widgets, fonts, alignments, colors
 - * Designers put most of their attention on these unimportant details
- 📄 Take much longer to use
 - * Poor support for iterative design
 - * Sketched interface took 5 times longer with traditional tools

Paper Sketches

Advantages

- * Support brainstorming
- * Do not require specification of details
- * Designers feel comfortable sketching

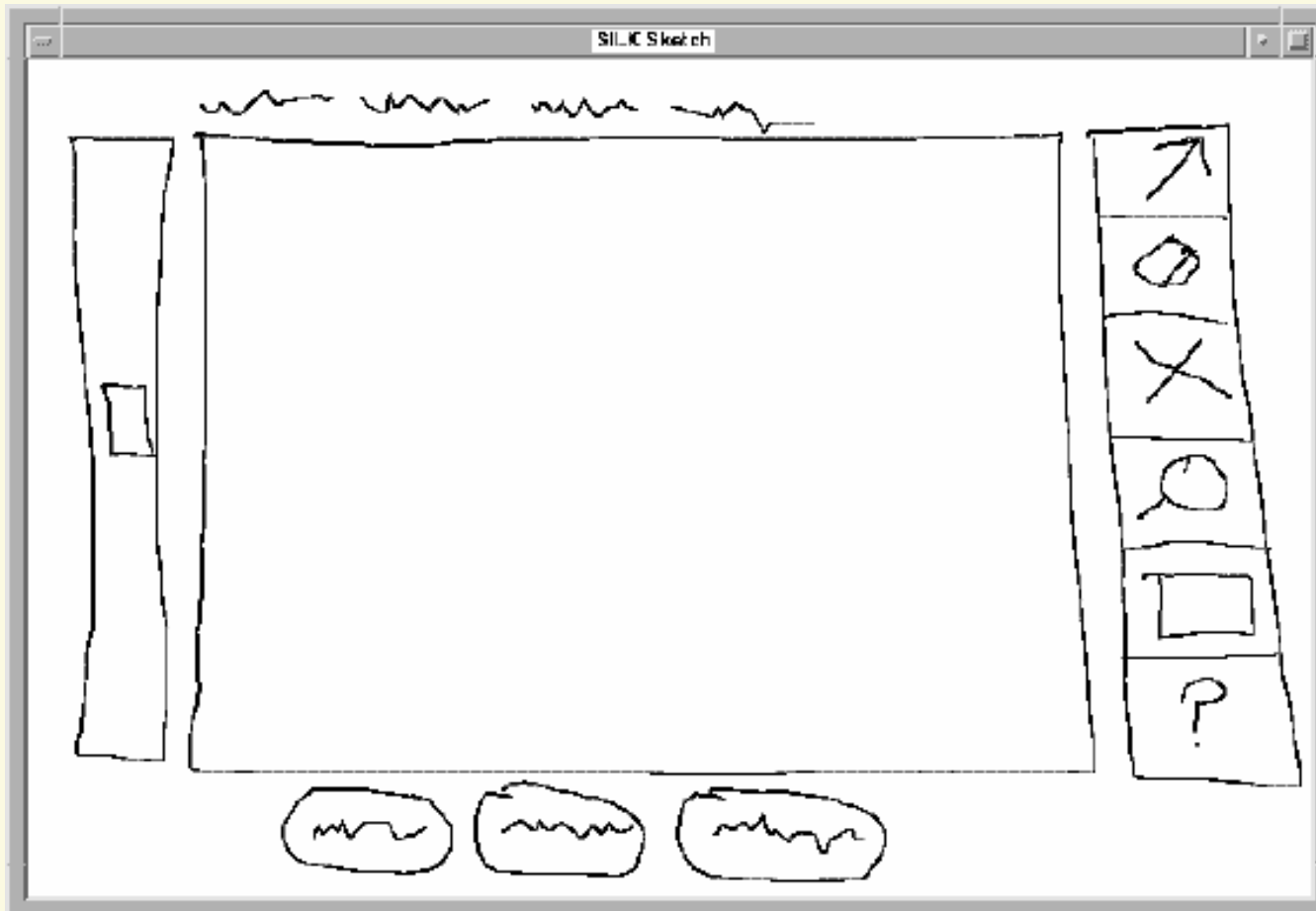
Drawbacks

- * Do not evolve easily
- * Lack support for "design memory"
- * Force manual translation to electronic format
- * Do not allow real end-user interaction (need WoZ)

What is SILK????

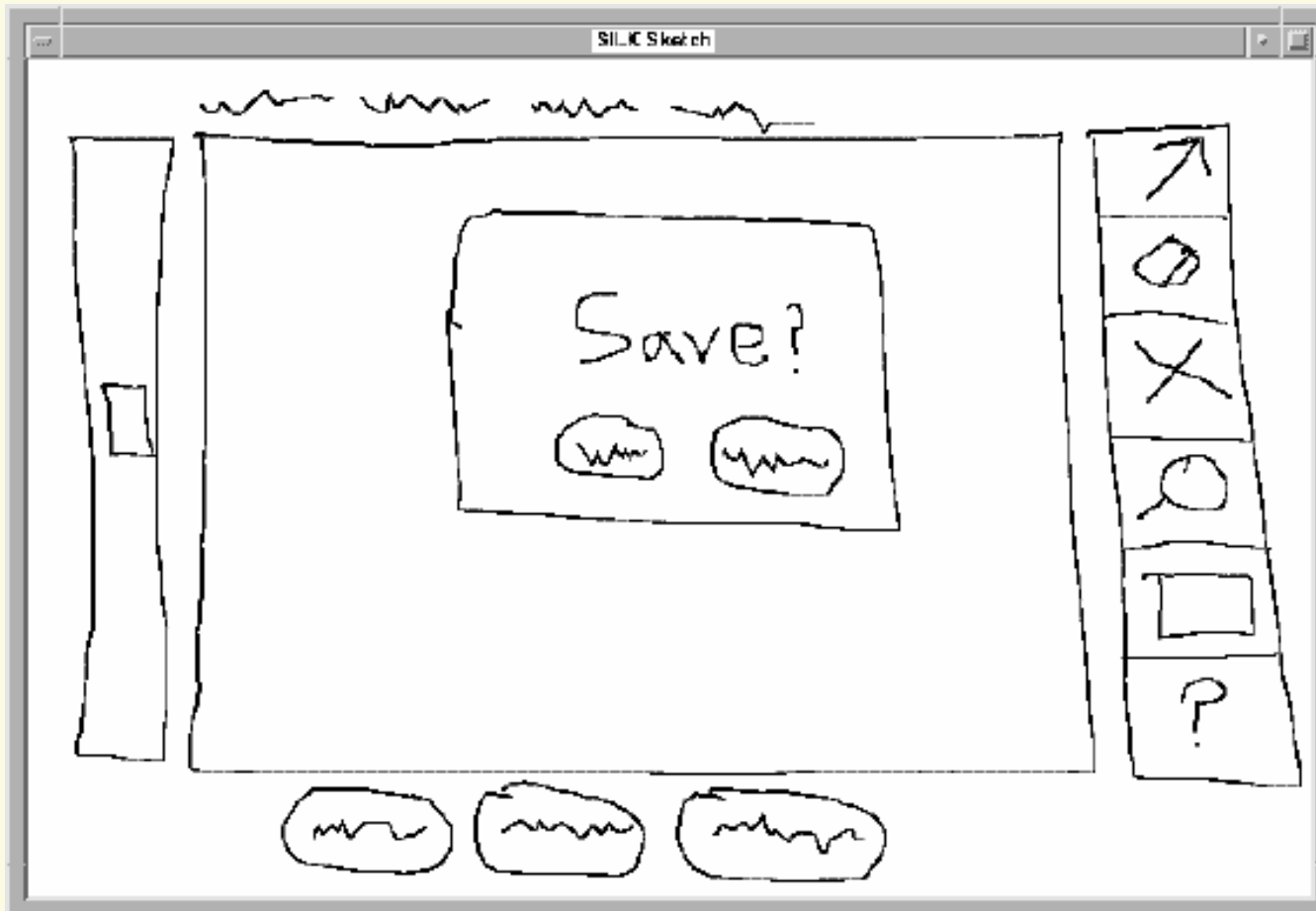
Sketching
Interfaces
Like
Krazy

Quickly Sketch this...



2/8/2006

Add Behavior...



2/8/2006

Designing Interfaces with SILK

- ☞ Designer sketches ideas rapidly with electronic pad and pen
 - * SILK recognizes widgets
 - * Easy editing with gestures
- ☞ Designer or end-user tests interface
 - * Widgets behave
 - * Specify additional behavior visually
- ☞ Automatically transforms to a "finished" UI
 - * Although this isn't necessarily the way to go...

Designing Interfaces with SILK

- ☞ Behavior of widgets takes over some of the tedious aspects of Woz:
 - * Recognizing and reacting to commands
 - * Moving dialog boxes around
 - * Following a script/flowchart

DENIM: Designing Web Sites by Sketching

- Early-phase information & navigation design

- Supports informal interaction

 - * sketching, pen-based interaction

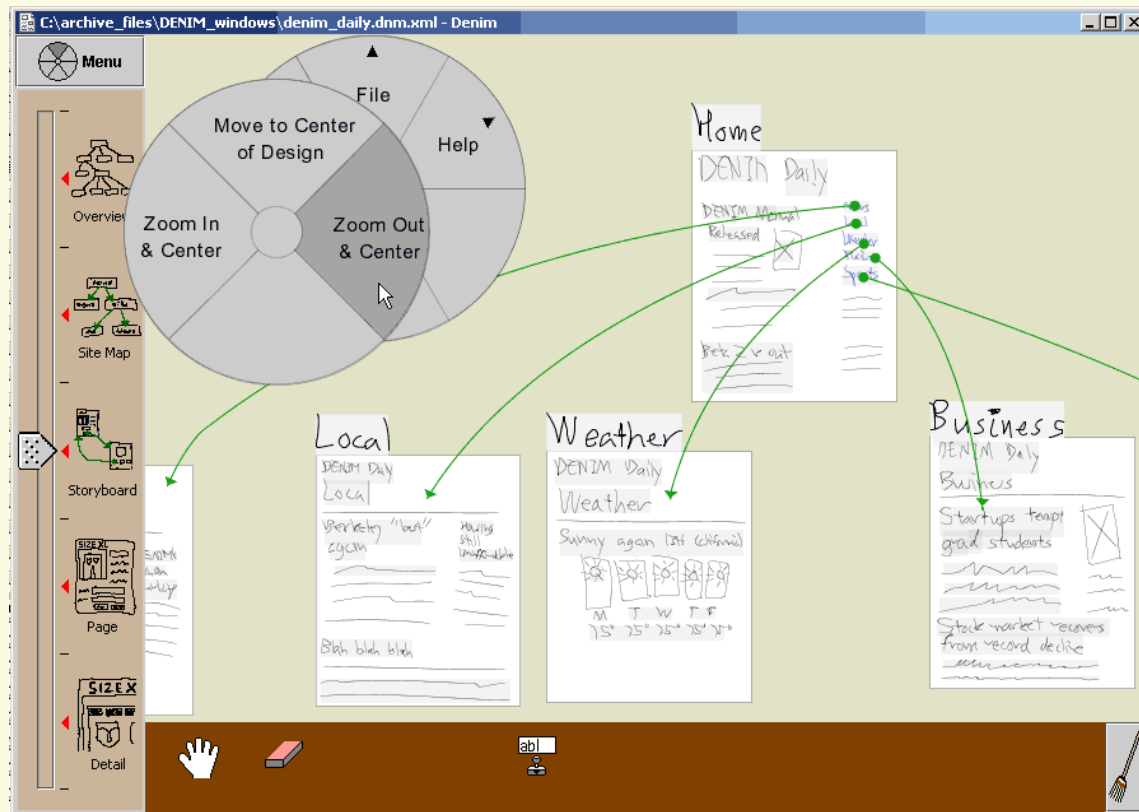
- Available and used by CS160 project groups:

<http://quir.berkeley.edu/projects/denim/>



DENIM: Designing Web Sites by Sketching

- DENIM's features are based on interviews with many designers.

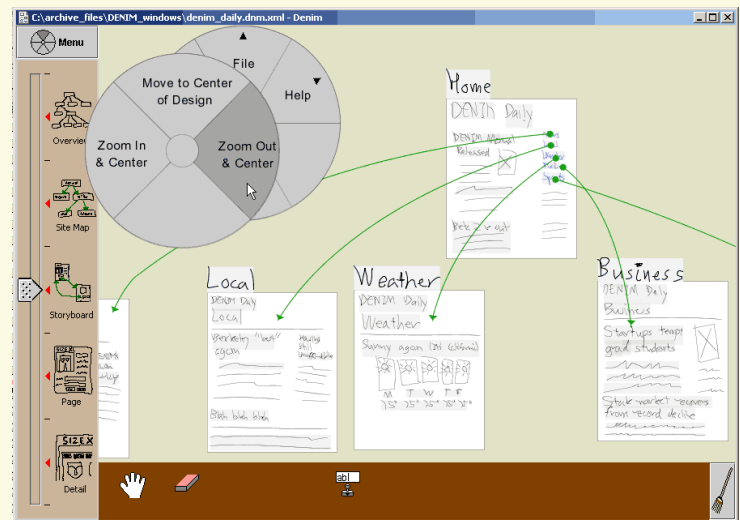


2/8/2006

52

DENIM: Designing Web Sites by Sketching

- ❏ DENIM supports navigation between pages, at both page and anchor level.
- ❏ It has multiple "scale" views for the entire site, part of the site, a pair of pages, a single page, or part of a page.



Caveats

- ☞ There is a down-side to the informal design approach:
- ☞ Often hard to involve *paying* clients as subjects - they treat the fidelity of the interface as a sign of development effort.
- ☞ Mitigators: involve them early and often, correspond with the same people, explain the process up front (set expectations).

Summary

- ☞ Informal prototypes allow you to design (and test!) before writing code.
- ☞ Rapid evolution and elimination of many problems happens in this phase.
- ☞ Paper+ink is the traditional tool
- ☞ Some emerging research tools (SILK, DENIM) also support informal design.