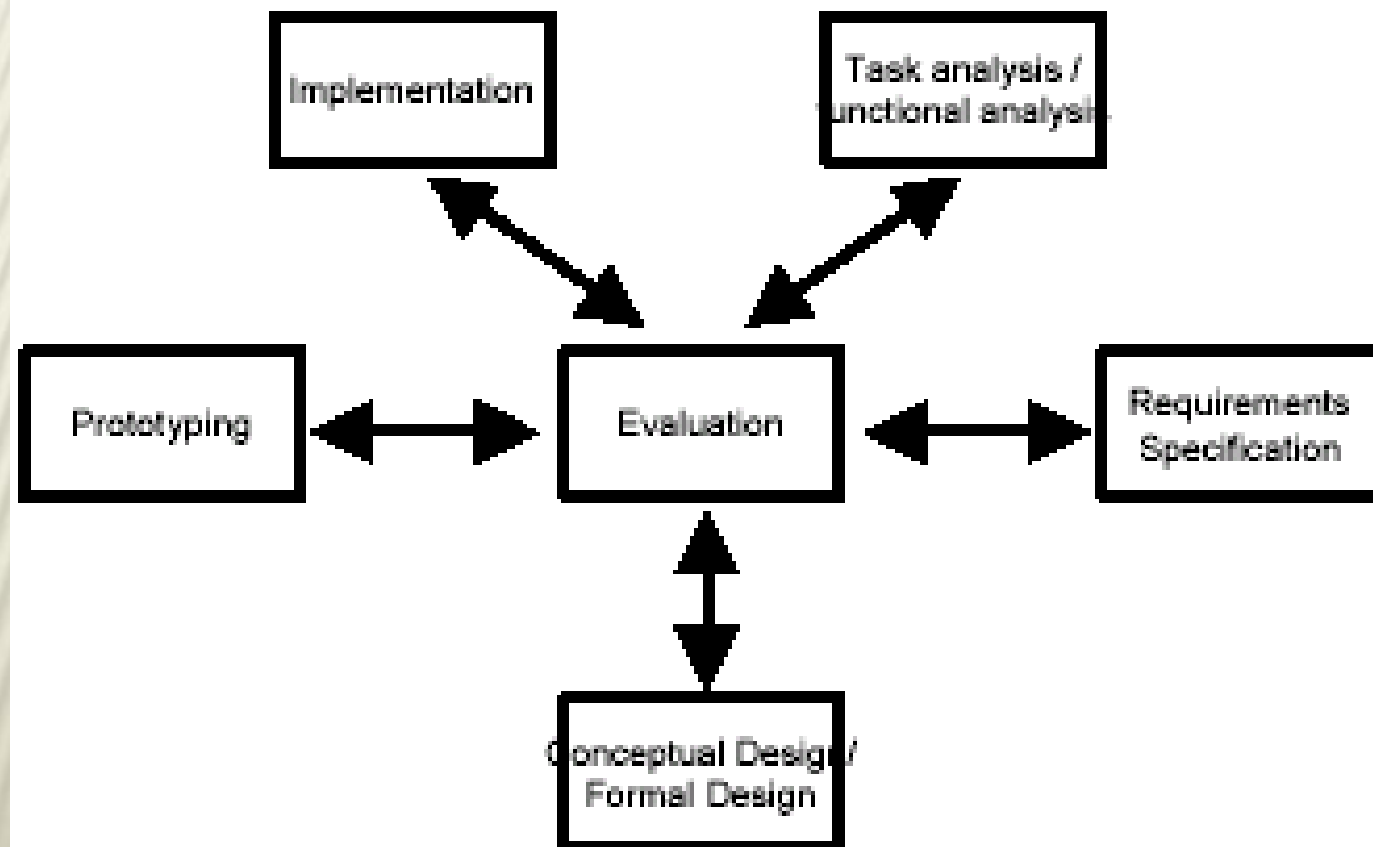


PROTOTYPES – TECHNIQUES AND TOOLS

STAR MODEL



WHY NOT JUST ASK USERS WHAT THEY WANT AND HOW IT SHOULD LOOK?

- ✘ Users lack the ability to imagine the ramifications of design decisions
- ✘ Users are often unable to comment on technical design documents
- ✘ Providing complete, consistent and readable representation of user interaction virtually impossible

PROTOTYPES

- ✖ May be thrown out or evolve
- ✖ May serve multiple purposes
- ✖ Must be built quickly and cheaply

WHY PROTOTYPE?

- ✖ Test technical feasibility
- ✖ Clarify vague requirements
- ✖ Get usability feedback
- ✖ Compare design approaches
- ✖ Generate interest or “buy-in”
- ✖ Avoid added expense

WHAT TO PROTOTYPE

- × Technical aspects
- × Workflow, task design
- × Screen layouts and information display
- × Graphic design, look-and-feel
- × Content and nomenclature
 - + e.g. taxonomy of objects/widgets, etc.
- × Controversial or critical areas
 - + e.g. security issues

TYPES OF PROTOTYPING

- × Fidelity
 - + Low Fidelity
 - + High Fidelity
- × Structural
 - + Horizontal
 - + Vertical

PROTOTYPE FIDELITY

- ✖ Fidelity measures how similar the prototype is to the finished interface
- ✖ Low fidelity: cheaper materials, omit details
- ✖ High fidelity: look more like finished product

DIMENSIONS OF FIDELITY (1)

- ✖ Breadth: % of features covered by the prototype
 - + Low-fi in breadth: Only enough features for a certain task
- ✖ Depth: degree of functionality of the prototype
 - + Low-fi in depth: limited choices, canned responses, no error handling

DIMENSIONS OF FIDELITY (2)

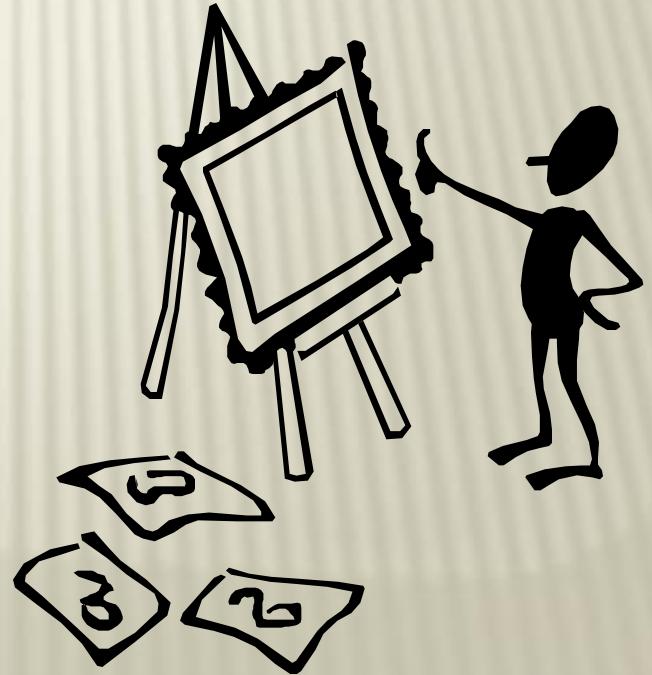
- × Look: appearance, graphic design
 - + Sketchy, hand-drawn
- × Feel: input method
 - + Physical methods by which the user interacts with the prototype
 - × E.g.: Pointing and writing feels very different from mouse and keyboard

LOW FIDELITY TYPES

- × Low-Fi Prototypes
 - + Textual Scenarios
 - + Mockups / Design Sketches
 - + Storyboards (limited)

LOW FIDELITY PROTOTYPES

- ✖ Paper-based prototypes
 - + a paper mock-up of the interface look, feel, functionality
 - + “quick and cheap” to prepare and modify
- ✖ Purpose
 - + brainstorm competing representations
 - + elicit user reactions
 - + elicit user modifications / suggestions



LOW FIDELITY PROTOTYPES – EXAMPLE 1

× Sketches

- + drawing of the outward appearance of the intended system
- + crudity means people concentrate on high level concepts
- + but hard to envision a dialog's progression

Computer Telephone

Last Name:

First Name:

Phone:

Place Call

Help

EXAMPLE 2 –STORYBOARDING WITH PAPER

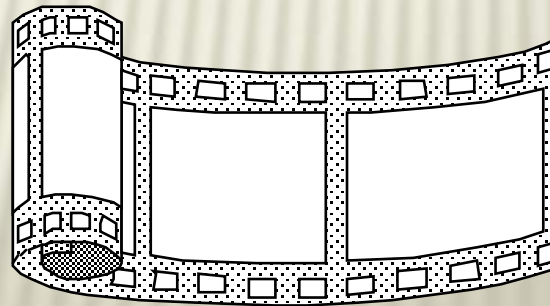
× Storyboarding

- + a series of key frames

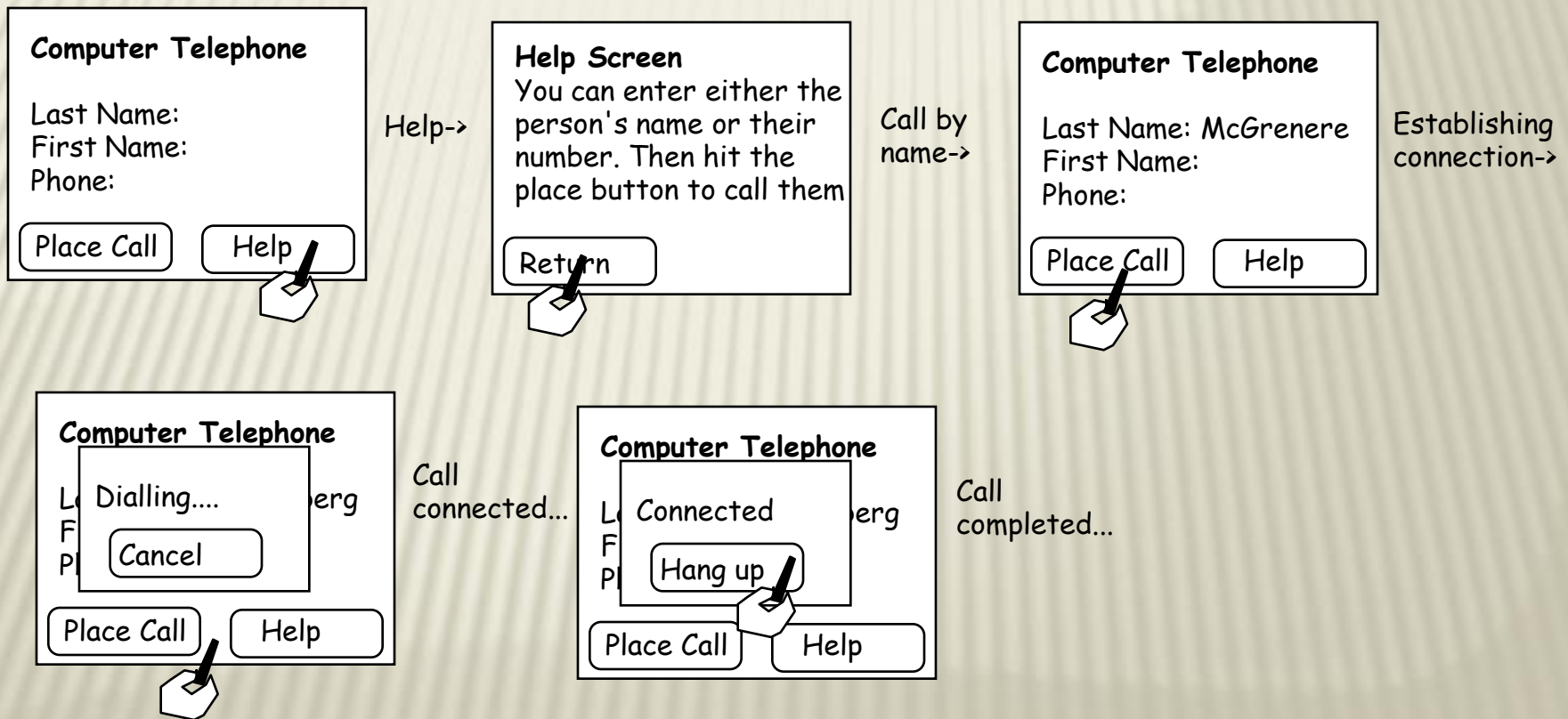
- × originally from film; used to get the idea of a scene

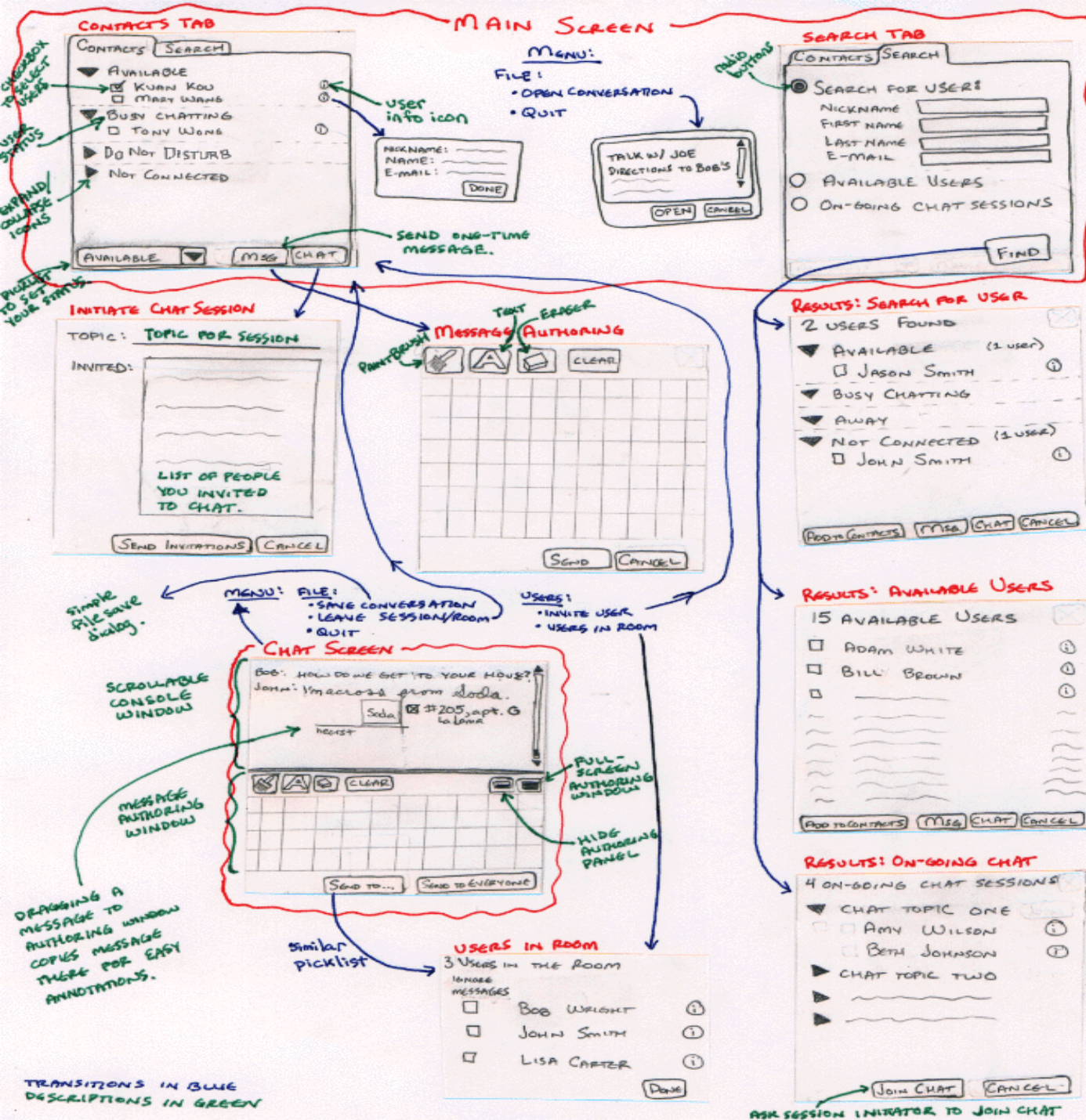
- × Paper screens of the interface at particular points in the interaction

- + users can evaluate quickly the direction the interface is heading



PAPER STORYBOARD OF A COMPUTER BASED TELEPHONE



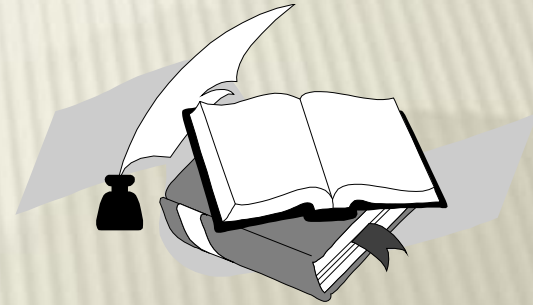


Ink Chat

EXAMPLE 3 - TUTORIALS

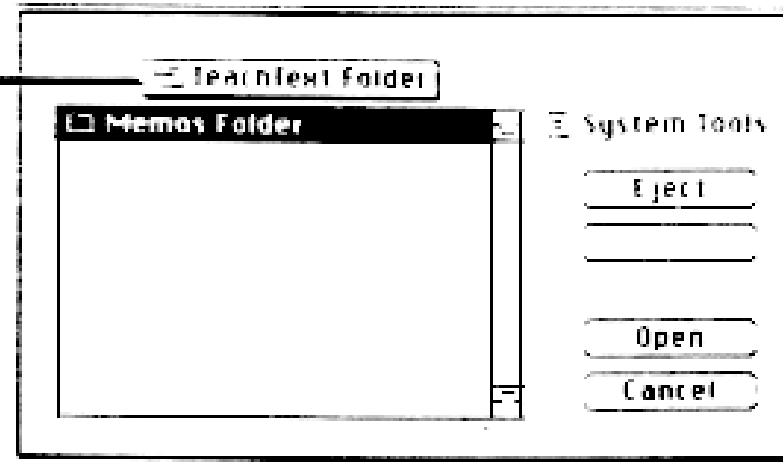
✖ Tutorials and manuals

- + write them in advance of the system
- + what are they?
 - ✖ tutorial for step by step description of an interaction
 - ★ an interface “walk-through” with directions
 - ✖ manual for reference of key concepts
 - ★ in-depth technical description
- + if highly visual, then storyboard is set within textual explanations
- + does this work?
 - ✖ people often read manuals of competing products to check:
 - ★ interface
 - ★ functionality
 - ★ match to task



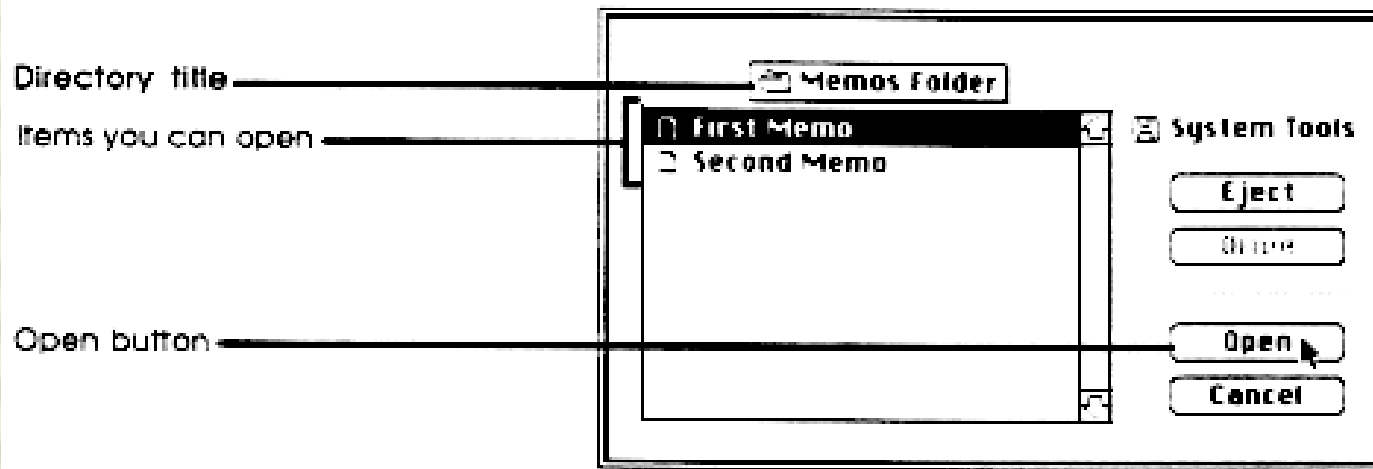
You see this dialog box:

Directory title



A directory title shows you the name of the folder you're presently working in—in this case, the TeachText Folder. The box beneath it shows you all the other items in the TeachText Folder that you can open with this application—in this case, only the Memos Folder.

- To open the Memos Folder, click the Open button.



As you open the Memos Folder, you move down through the hierarchy. The directory title changes to remind you where you are in the hierarchy, and the box shows you what's on the new level you just moved to—in this case, the two documents in the Memos Folder. The selected document is the one that will open when you click the Open button. If you want to open the other document, click anywhere on the other document's name to highlight it, and then click the Open button.

TIPS FOR CREATING PAPER PROTOTYPES

- ✕ Make it BIG
- ✕ Make it monochromatic
- ✕ Replace tricky visual feedback with audible descriptions
- ✕ Keep pieces organized

HOW TO TEST A PAPER PROTOTYPE

× Roles for the design team

+ 'Computer'

- × manipulates prototype
- × Doesn't give any feedback that the computer wouldn't

+ Facilitator

- × Explains purpose and process of the user study
- × Obtains the user's informed consent
- × Presents the tasks one by one
- × Encourages user to think aloud while the user is working on a task

+ Observers

- × Just for watching, not helping the user
- × Takes notes

WHAT YOU CAN LEARN FROM A PAPER PROTOTYPE

- ✖ Conceptual model
 - + Do users understand it?
- ✖ Functionality
 - + Does it do what is needed?
- ✖ Navigation and task flow
 - + Can users find their way around?
 - + Are task preconditions met?
- ✖ Terminology
 - + Do users understand labels?
- ✖ Screen contents
 - + What needs to go on the screen

WHAT YOU CAN'T LEARN

- ✗ Look: color, font, white space etc.
- ✗ Feel: interaction with paper prototype may be different than with final UI
- ✗ Response time (backend simulated by 'human computer')
- ✗ Are small changes noticed?
 - + Even the tiniest change to a paper prototype is clearly visible to the user
- ✗ Exploration versus deliberation
 - + Users are more deliberate with a paper prototype; they don't explore as much

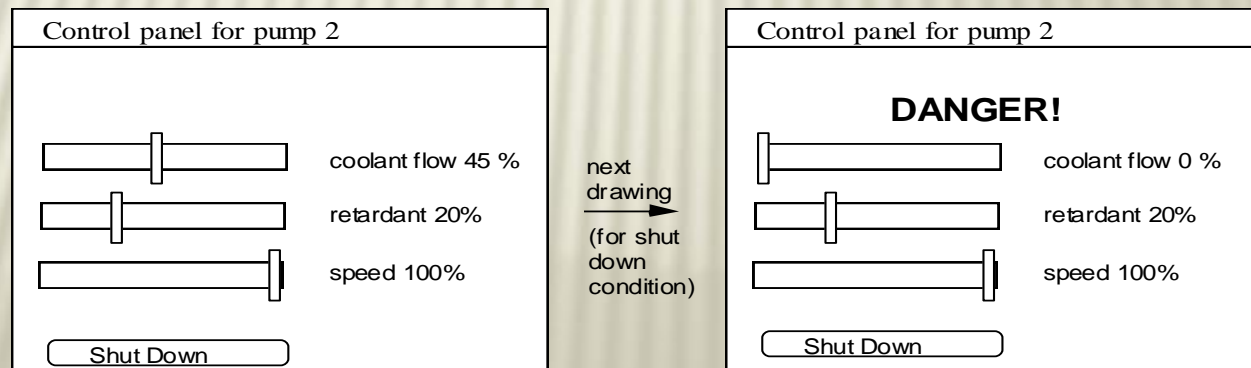
HIGH FIDELITY PROTOTYPING

× High-Fidelity Prototypes

- + Interaction with Computer or with something that appears to be a real device....('machine interaction')

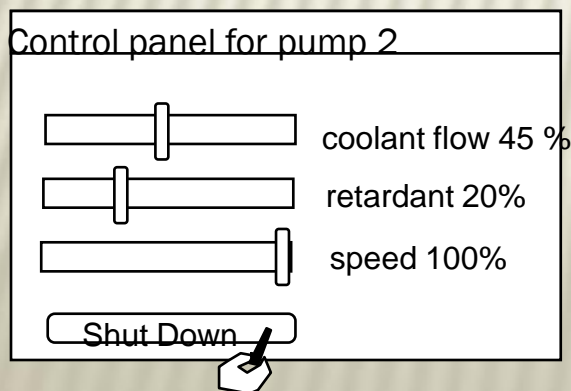
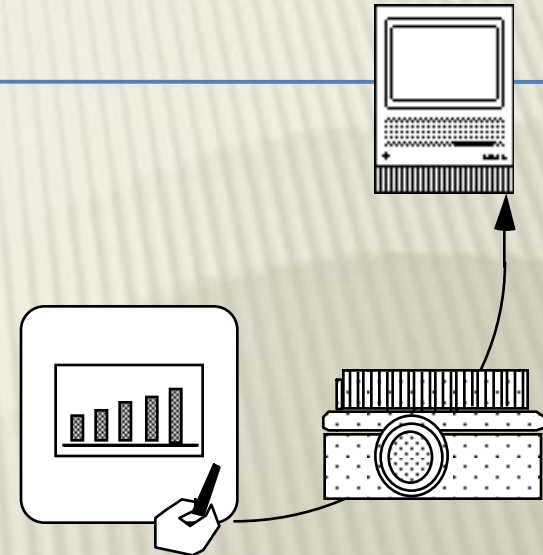
HIGH FIDELITY PROTOTYPES

- ✖ Painting/drawing packages- that create screens with story line
 - + draw each storyboard scene on computer
 - ✖ neater/easier (?) to change on the fly than paper
 - + a very thin horizontal prototype
 - + does not capture the interaction “feel”

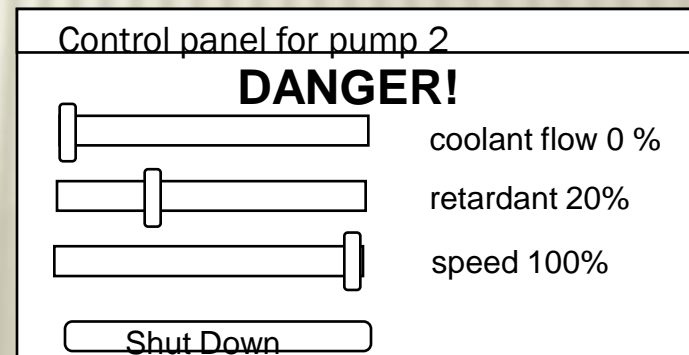


HIGH FIDELITY PROTOTYPES

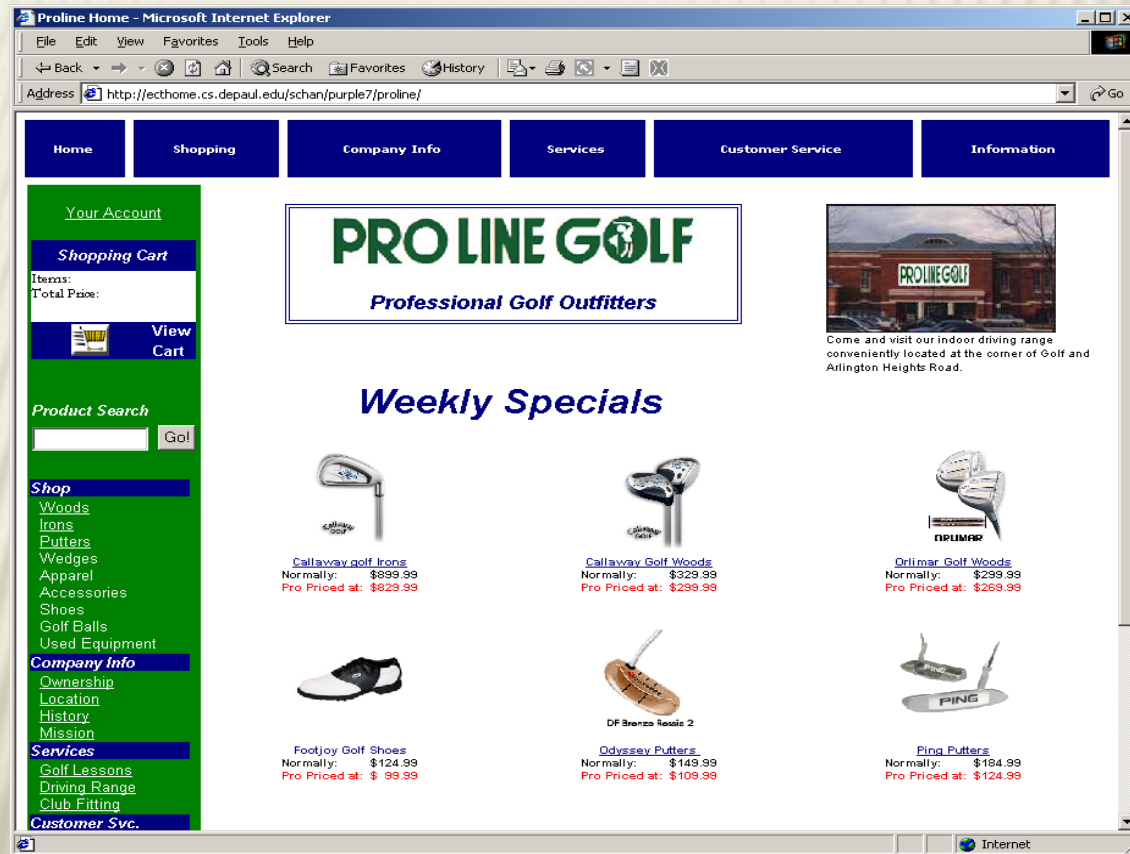
- × Scripted simulations and slide shows
 - + encode the storyboard on the computer
 - × created with media tools
 - × scene transition activated by simple user inputs
 - × a simple horizontal and vertical prototype
 - + user given a very tight script/task to follow
 - × appears to behave as a real system
 - × but script deviations blows the simulation



next
drawing
→
(on mouse
press over
button)



A HIGH-FIDELITY PROTOTYPE



WHAT CAN YOU LEARN FROM COMPUTER PROTOTYPES

Everything you learn from paper prototypes plus:

- ✖ Screen layout
 - + Is it clear, overwhelming, distracted, complicated?
 - + Can users find important elements?
- ✖ Colors, fonts, icons, other elements
 - + Well-chosen?
- ✖ Interactive feedback
 - + Do users notice and respond to status bar messages, cursor changes, other feedback
- ✖ Fitt's law issues
 - + Controls large enough? Too close together? Etc.

COMPUTER PROTOTYPING TECHNIQUES

- × Storyboard
- × Sequence (graph) of painted screenshots connected by hyperlinks (“hotspots”)
- × Form builder: a tool for drawing real, working interfaces by dragging widgets from a palette and positioning them on a window
- × Wizard of Oz: hybrid between a computer prototype and a paper prototype

STORYBOARDING TOOLS

- × HTML: each screen is an image map
- × Flash:
 - + Tool for constructing multimedia interfaces;
 - + Particularly useful for prototyping interfaces with rich animated feedback
- × PowerPoint: images + links + animation

PROS AND CONS OF STORYBOARDING

× Pros

- + You can draw anything

× Cons: storyboarding is static

- + All you can do is click, not enter text
- + Widgets (scrollbars, list boxes, buttons) are static pictures
- + Testing with users: “Hunt for the hotspot”

FORM BUILDERS

- ✖ HTML pages and forms
 - + Natural if you're building a web application of your web interface
 - + May have low-fidelity look
- ✖ Java GUI builders: Sun NetBeans, Eclipse, Visual Editor, Borland JBuilder
- ✖ Other gui builders
 - + Visual basic, NET windows forms, Mac Interface...

PROS & CONS OF FORM BUILDERS

× Pros

- + Use actual working widgets, not just pictures of them
- + Can hook in some backend if you need it
 - × But then you won't want to throw it away

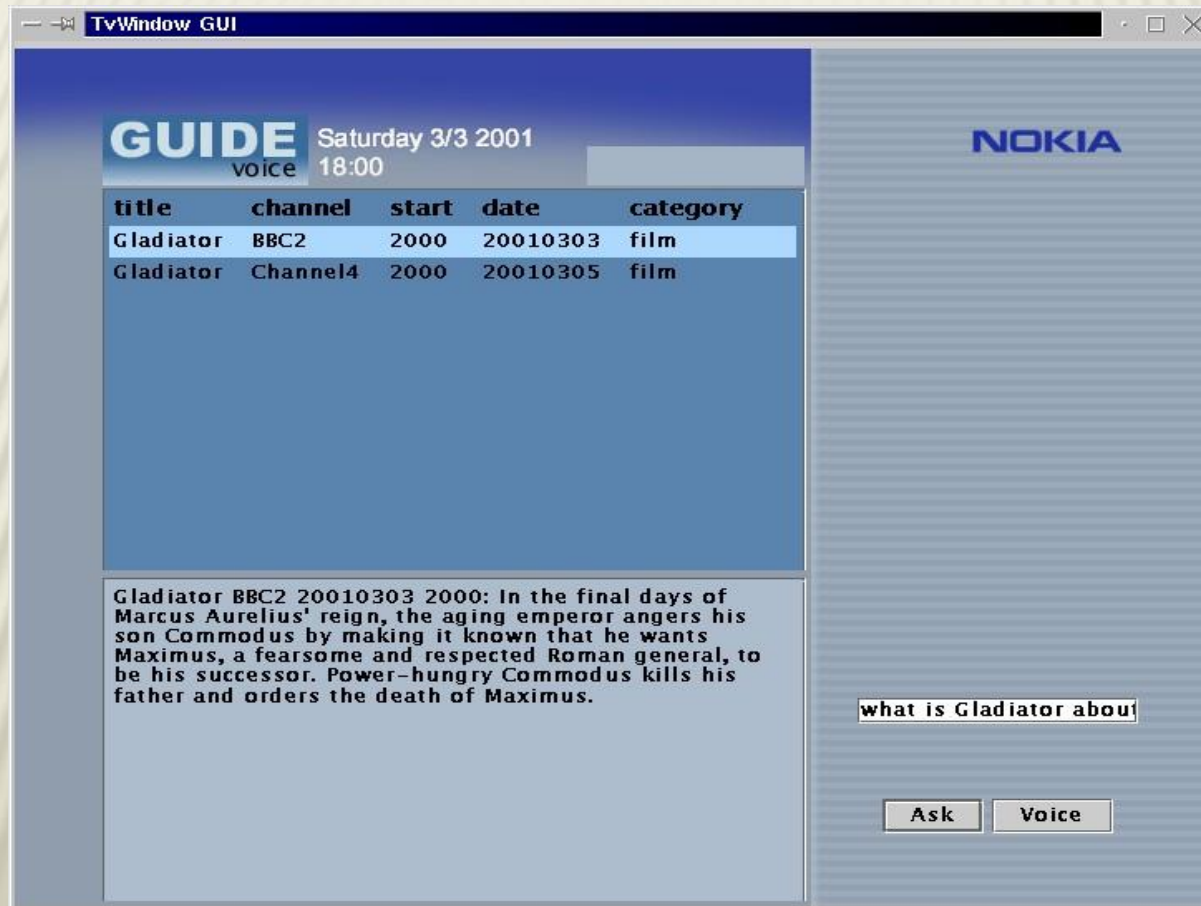
× Cons

- + Limits thinking to standard widgets
- + Useless for rich graphical interfaces

COMPUTER PROTOTYPING: AN EXAMPLE

- ✖ The Nokia TV guide: a dialogue system customized for TV program information to be used in the Nokia Media Terminal
- ✖ Iterative computer prototyping
- ✖ first iteration: prototype able to respond to isolated questions (no context used when interpreting a request)
- ✖ second iteration: enriched interaction, system more 'intelligent'

FIRST PROTOTYPES EXAMPLE



RESPONDING TO QUESTION- MOVIES AFTER 9 PM

GUIDE voice Saturday 3/3 2001 18:00

title	channel	start	date	category
Ryan's Daughter (15)	tcm.turner	2100	20010303	film
Operation Delta Force 4: Deep Fault	Channel5	2100	20010303	film
The Life of Stuff	BBC-knowledge	2230	20010303	film
Legal Eagles	BBC2	2235	20010303	film
Hang 'em High	Carlton	2315	20010303	film

NOKIA

movies after 9 pm

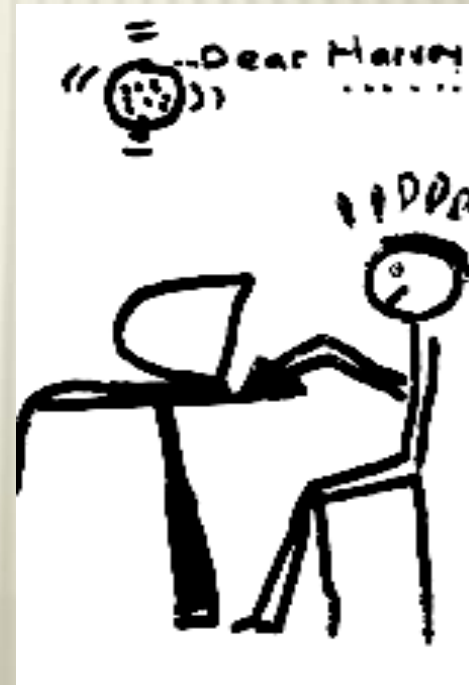
Ask Voice

WIZARD OF OZ

- ✖ A method of testing a system that does not exist
 - + the voice editor, by IBM (1984)



What the user sees



The Wizard

HIGH – FIDELITY PROTOTYPING

× ADVANTAGES

- + Actual functionality
- + Interactive
- + Working navigation
- + Supports interactive usability tests
- + Real “look-feel”
- + Living spec
- + Demo tool

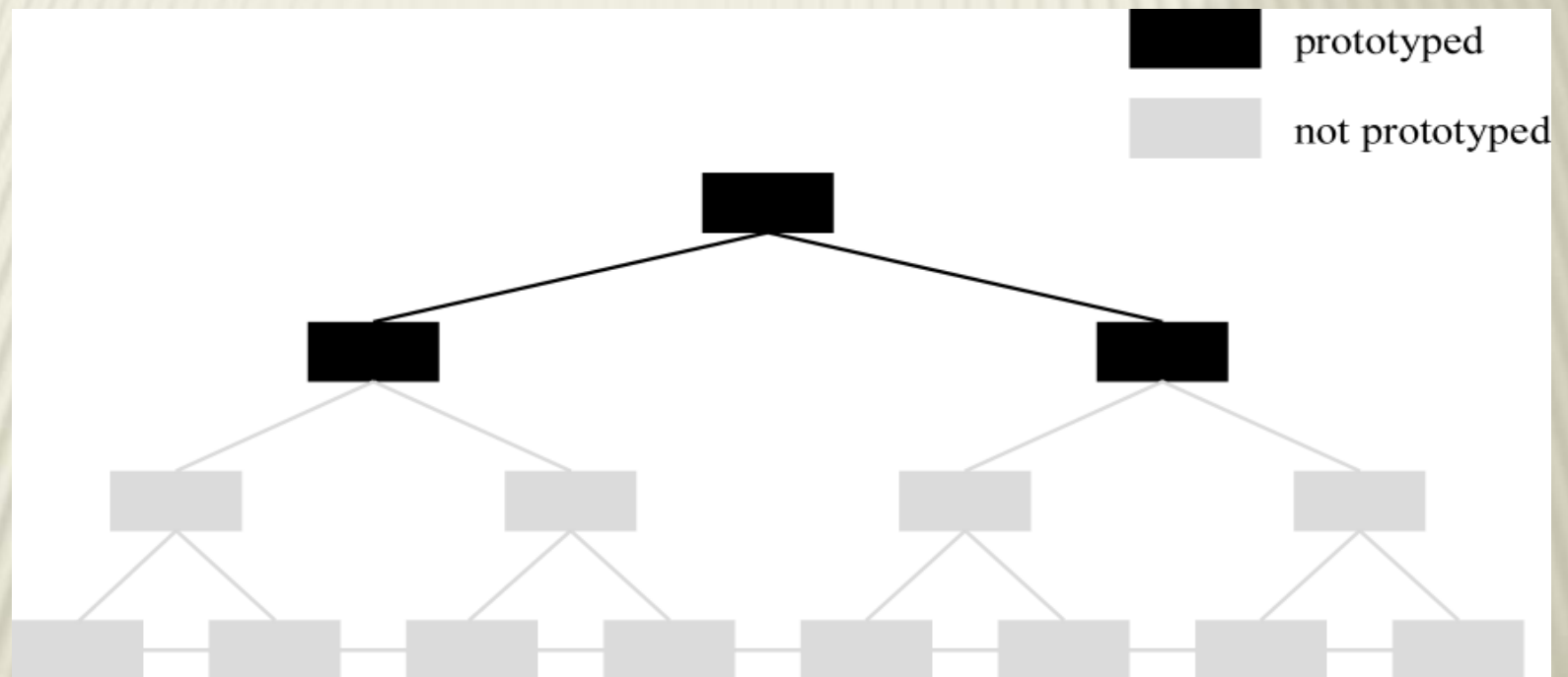
× DISADVANTAGES

- + Expense
- + Time
- + Too polished and expensive for effective req gathering and testing

STRUCTURAL PROTOTYPING

- × Prototyping different parts of the project
 - + Horizontal Prototyping
 - × Navigation
 - + Vertical Prototyping
 - × A task

HORIZONTAL PROTOTYPE



ΛΕΒΙΤΑΓ Γ ΕΒΟΙΘΙΣ Ε

