



UNIVERSITY OF SCIENCE
HO CHI MINH CITY

CTT534 – Thiết Kế Giao Diện

Prototyping

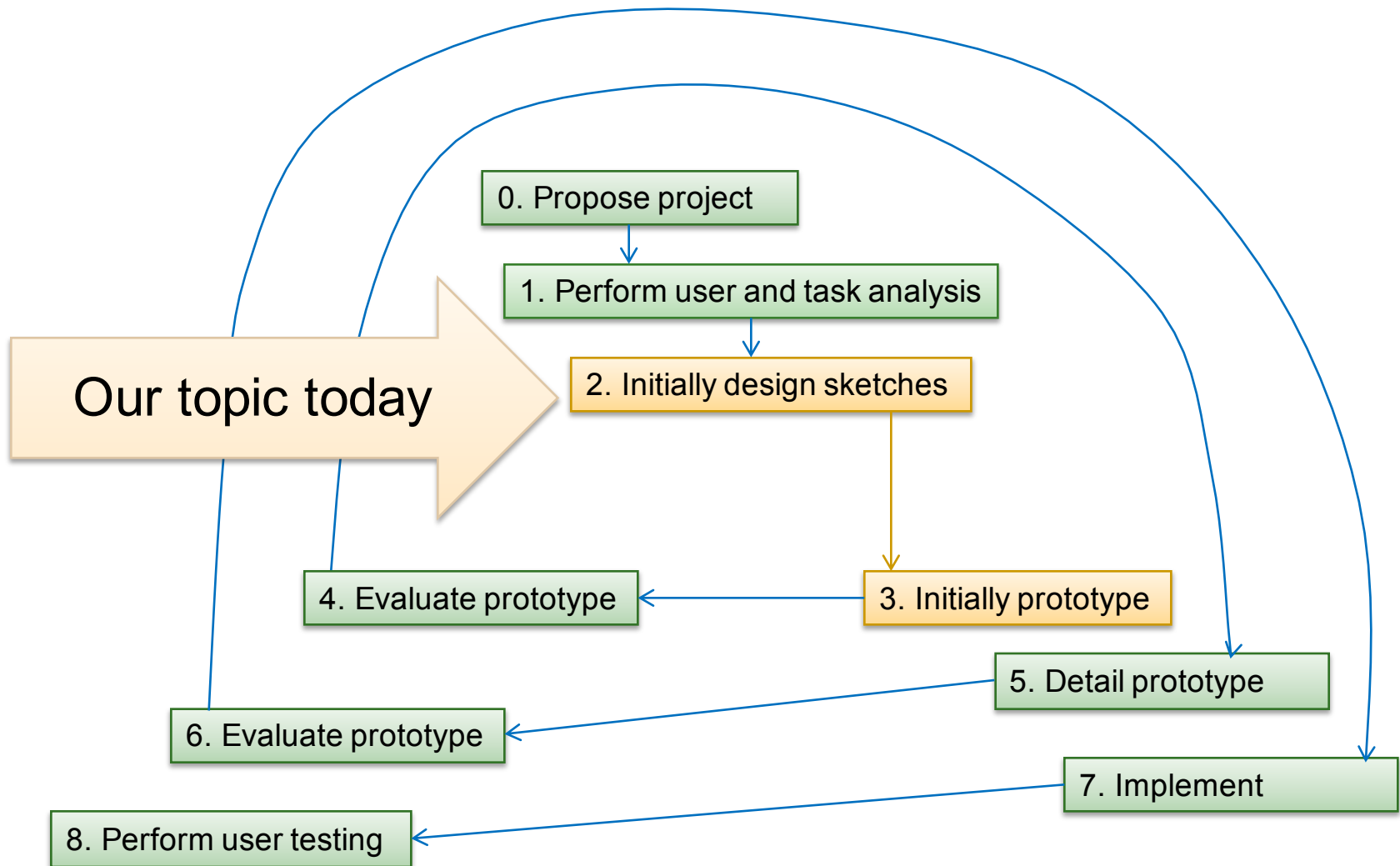
24/3/2016

Adapted from materials of MIT CS
Course 6.813/6.831

Outline

- Prototype
 - Low- and High-fidelity
 - Dimensions of fidelity
- Paper prototype
- Computer prototype
 - Storyboard
 - Form builder
 - Wizard of Oz

Process for Projects in This Class



Design alternatives

- Why we need to express design ideas early?
 - You can't evaluate design until it's created
 - After software is built, changes to the design are difficult
- We want
 - Make it fast
 - Allow lots of flexibility for radically different alternatives
 - Low cost
 - Promote valuable feedback

What is prototype?

- A prototype is an original type, form, or instance of something serving as a typical example, basis, or standard for other things of the same category
- Examples
 - ❑ Screen shots
 - ❑ Paper drawings
 - ❑ prototype software - referred to as alpha grade, meaning it is the first version to run

Why we need prototypes?

- Experiment with alternative designs
- Provide an early, concrete representation of design ideas
- Provide hands-on experience for all stakeholders (design teams, users, etc.)
- Easier to change or throw away
- Keep the design centered on the user
 - must test and observe ideas with users
- Facilitate iterative design and evaluation
- Reduce the risk of making customers surprise
 - “I won’t believe it until I see it”

Prototype fidelity

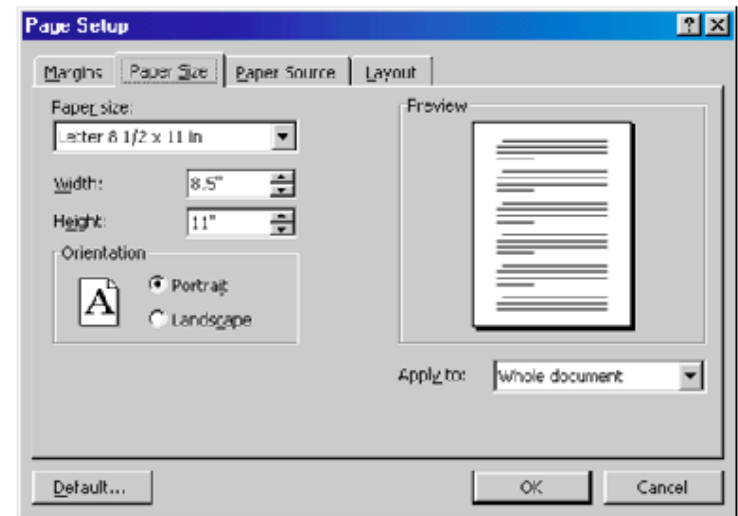
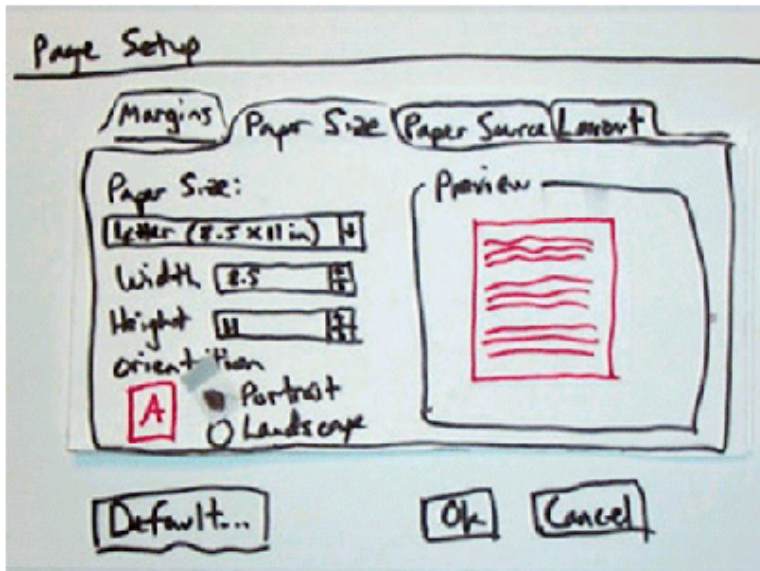
■ Low-fidelity

- ❑ a set of sketches/storyboards providing a static, non-computerized, non-working mockup of the planned product
- ❑ omits details

■ High-fidelity

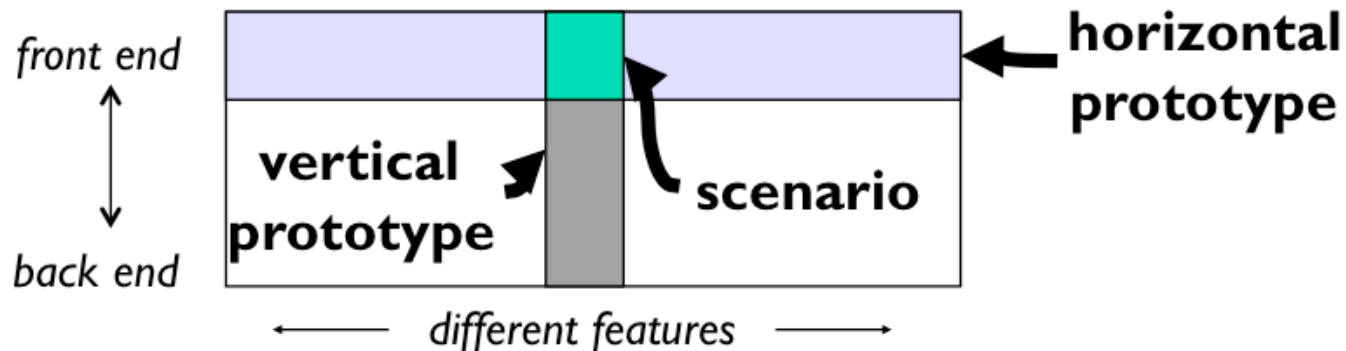
- ❑ a set of screens that provide a dynamic, computerized, working model of the planned product
- ❑ working software

Prototype fidelity (cont'd)



Dimensions of fidelity

- Horizontal
 - Prototypes cover many features but with little detail
- Vertical
 - Prototypes cover few features but with much in detail
- Diagonal
 - Prototypes cover down to a certain level and vertical



Dimensions of fidelity (cont'd)

■ Look

- ❑ Is the appearance and graphic design of the UI
- ❑ Can be sketchy and hand-drawn

■ Feel

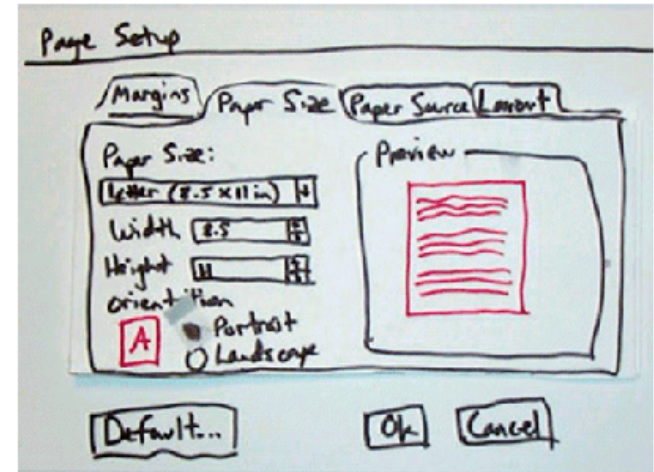
- ❑ Referring to input methods to interact with the UI
- ❑ Pointing and writing (in paper-mockup) is different from mouse and keyboard

Paper prototype - Videos

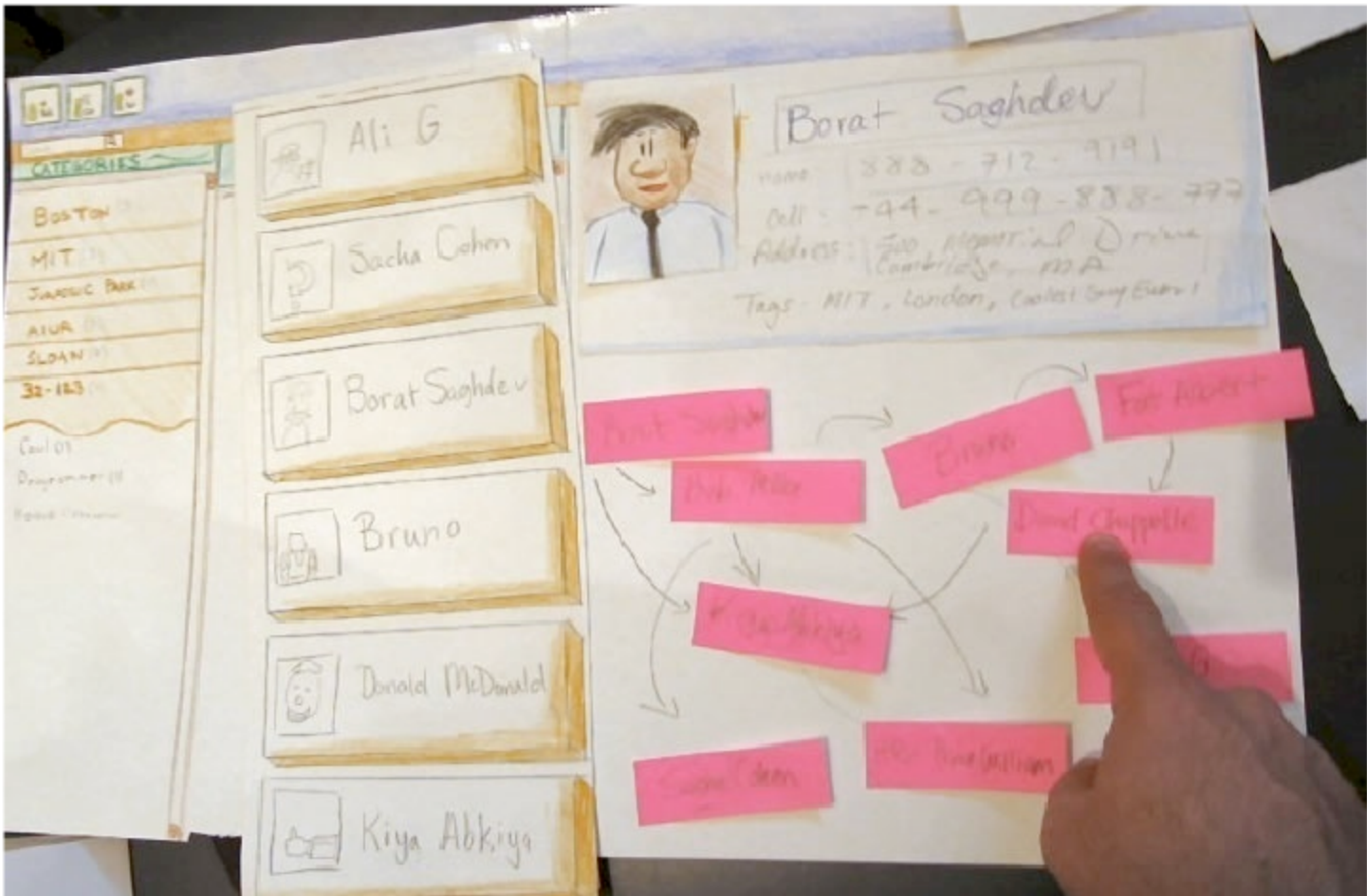
- “Example Usability Test with a Paper Prototype”
- “Hanmail Paper Prototype”
- “Paper prototype usability test”
- Find these videos on Youtube

Paper prototype

- Using paper mockup to represent the UI
 - Sketches of screen appearance
 - Paper pieces show windows, menus, dialog boxes, toolbars
- Interaction is natural
 - Pointing with a finger = mouse click
 - Writing = typing
- A person simulates the computer's operation
 - Putting down and picking up pieces
 - Writing responses on the mockup (screen)
 - Describing effects that are hard to show on paper



Paper prototype (cont'd)



Paper prototype (cont'd)

■ Characteristics

- ❑ Low fidelity in look and feel
- ❑ High fidelity in depth as there is person to simulate the operation

Paper prototype (cont'd)

■ Advantages

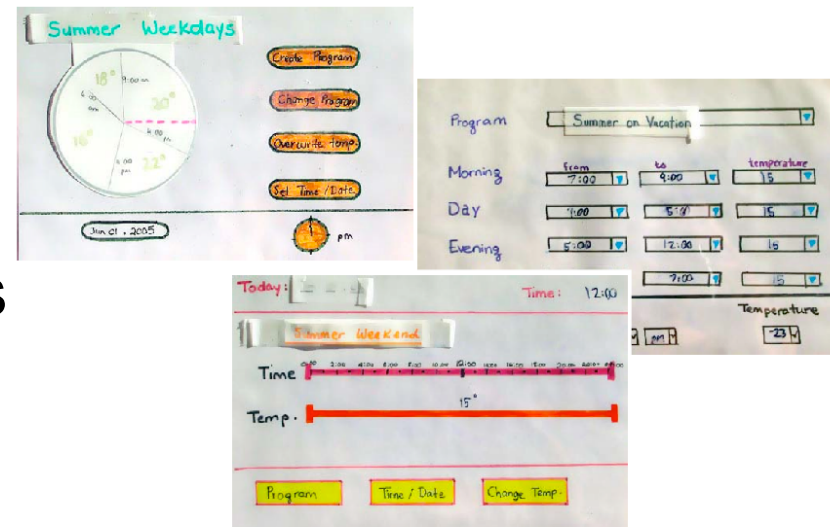
- ❑ Faster to build
- ❑ Easier to change
- ❑ Cheap
- ❑ Focuses attention on big picture
 - Designers don't waste time on details
 - Attract ideas from customers
- ❑ Non-programmers can help
- ❑ Convenient
 - You can prototype on the bus
 - You can utilize your time efficiently

Tools for paper prototyping

- White poster board
 - Used as background or window frame
- Index cards
- Re-stickable glue
- White correction type
- Photocopier
 - For making copies
- Pens, markers, scissors, type
- Etc.

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



What can paper prototypes help?

- It helps better understanding of
 - ❑ conceptual model
 - Do users understand the UI?
 - ❑ functionality of the system
 - What features are missing in the UI?
 - ❑ navigation and task flow
 - Do users understand the navigation of the UI?
 - ❑ terminology
 - Are terms and levels understood?
 - ❑ screen content and layout
 - What are there in the UI?

What it does not help?

- Showing “look”: color, font, whitespace, etc.
- Demonstrating “feel”: efficiency issues
 - Interactions are in low fidelity (not real)
- Measuring response time
- Demonstrating animation and high-level of interaction
 - Actions like drag and drop, drawing, etc.

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

What can computer prototypes help?

- Everything from paper prototypes, plus
- Better and higher-fidelity **look**
 - Screen layout
 - Colors, fonts, icons, etc.
 - Choices of controls
- Interactive feedback
- Efficient issues
 - Controls are big enough?
 - Whitespace?
 - Distance between controls?

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

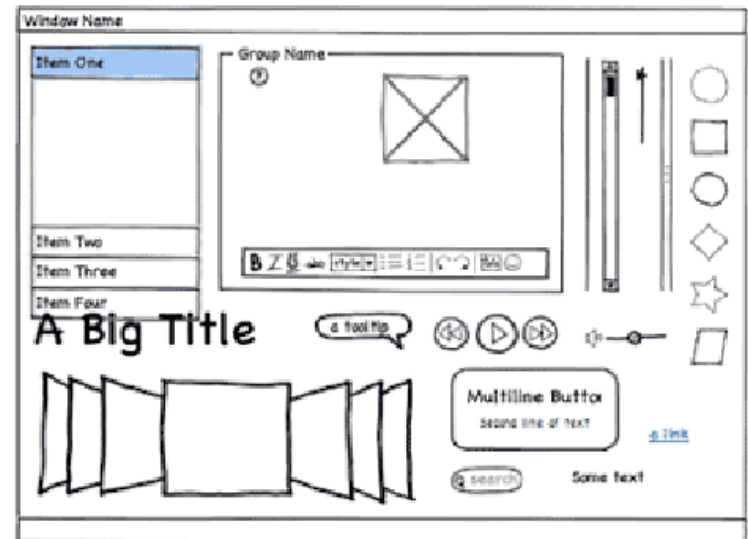
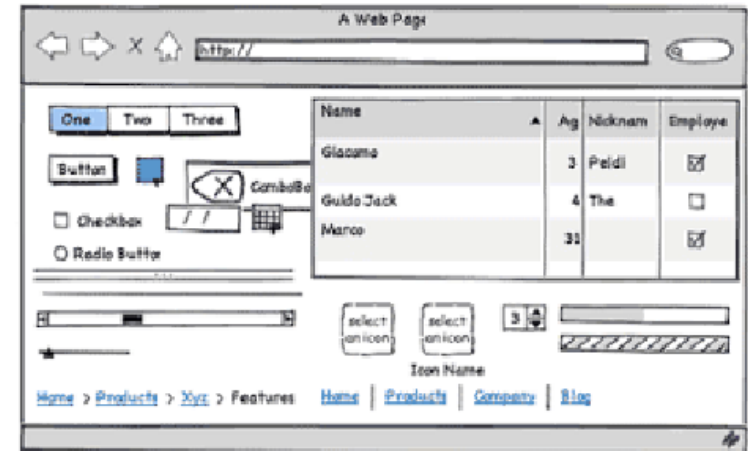
Computer prototyping techniques

- 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

Computer prototyping techniques (cont'd)

■ Storyboarding tools

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



Computer prototyping techniques (cont'd)

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

Computer prototyping techniques (cont'd)

■ Form builders

- ❑ FlexBuilder
- ❑ Silverlight
- ❑ Visual Basic
- ❑ C++ Builder
- ❑ Visual C#
- ❑ Qt Designer

Computer prototyping techniques (cont'd)

■ 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

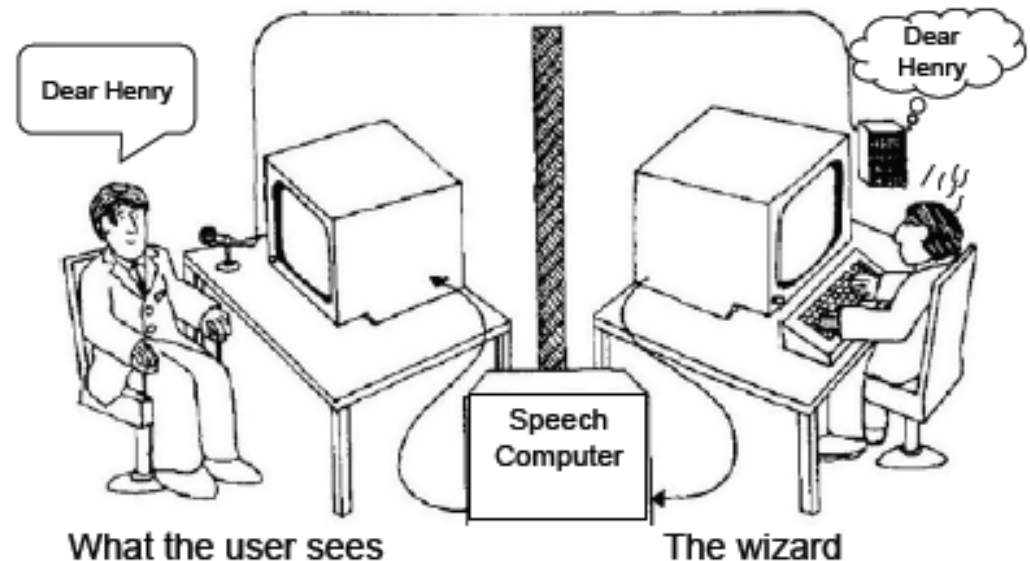
Computer prototyping techniques (cont'd)

■ 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)

Computer prototyping techniques (cont'd)

- Wizard of Oz
 - Faking the interaction



- Issues
 - Wizard has to be mechanical (pretending to be non-human)
 - Worry about both UIs: for wizard and users

Summary

- Prototype
 - Low- and High-fidelity
 - Dimensions of fidelity
- Paper prototype
- Computer prototype
 - Storyboard
 - Form builder
 - Wizard of Oz