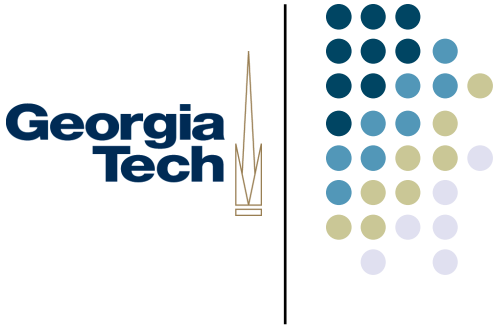
Prototyping

John Stasko

CS 6452

Prototyping Interactive Systems

Learning Objectives

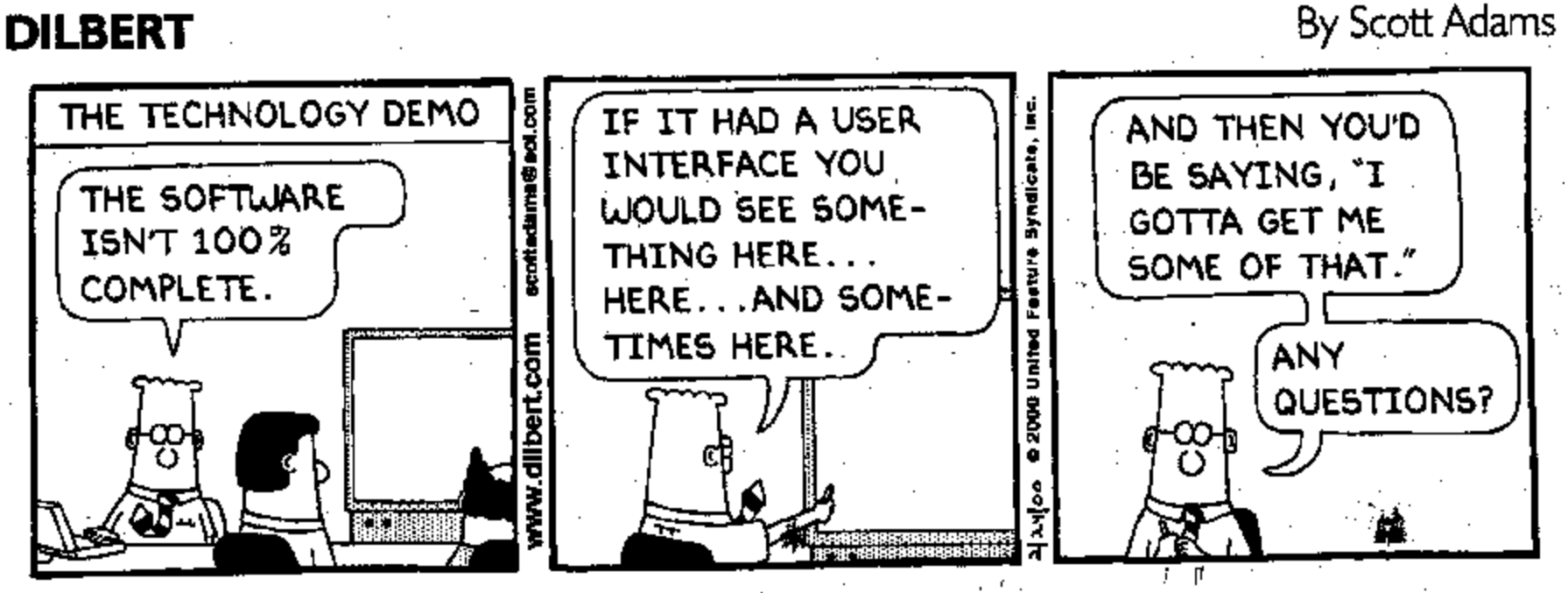


- Goals of prototyping
- Dimensions of prototyping
- Prototyping techniques
- Prototyping tools

What is Prototyping?

- The creation of *artifacts* that can be used to:
  - Assess the utility and usability of a proposed system, through *evaluation*
  - Communicate design alternatives with various stakeholders
    - The "customer"
    - Engineers/builders
    - Management

Well done


DILBERT
By Scott Adams
THE TECHNOLOGY DEMO
THE SOFTWARE ISN'T 100% COMPLETE.
IF IT HAD A USER INTERFACE YOU WOULD SEE SOMETHING HERE... HERE...AND SOMETIMES HERE.
AND THEN YOU'D BE SAYING, "I GOTTA GET ME SOME OF THAT."
ANY QUESTIONS?
www.dilbert.com scottadams@aol.com
© 2000 United Feature Syndicate, Inc.

Paper Discussion

"Prototyping Debate", J. Rudd, K. Stern, S. Isensee, interactions, Jan. 96

Two Main Goals

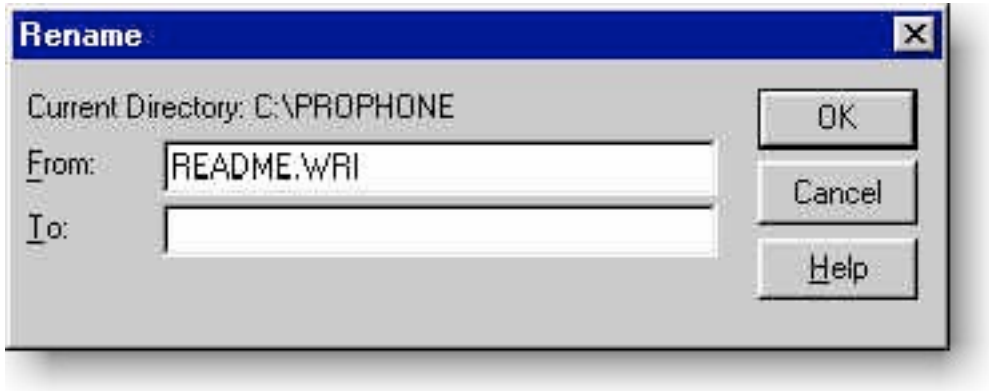
- Ideally, a prototype should
  - … be quick enough to build to allow easy experimentation
  - … have fidelity *appropriate* to demonstrate the desired concepts

Why Prototype?

- In two words: **risk mitigation**
- From an evaluation perspective, allows you to get feedback on designs before there's a huge investment in it
- From a design perspective, allows you to quickly experiment with alternatives, cheaply

An Example

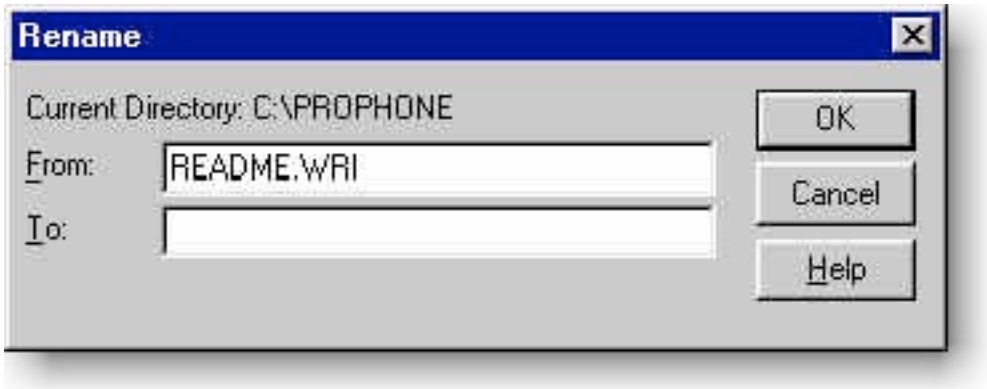
- When interfaces go bad…



- What's wrong with this?

An Example

- When interfaces go bad…



- What's wrong with this?
  - The "From" field is editable, but doesn't do anything!
  - Let's you change the file extension without warning
  - Is modal!
- Could this have been saved by prototyping?

Another Example

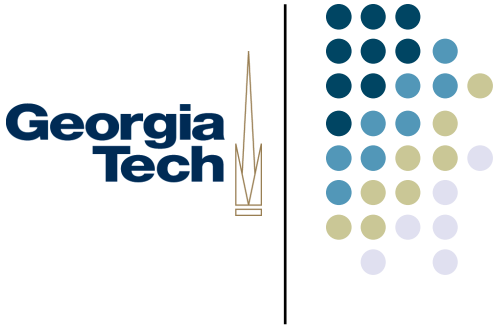
- Not just restricted to applications...

"If you are seated in an exit row and you cannot understand this card or cannot see well enough to follow these instructions, please tell a crew member."

Insert your favorite bad design here

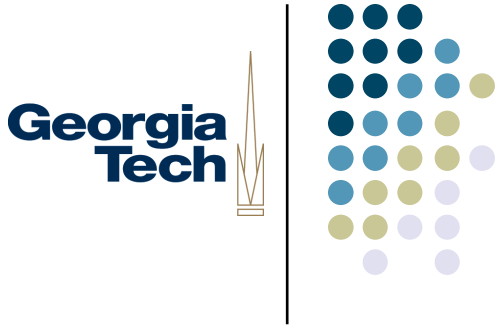
- Might a prototype have helped matters?

Kinds of Prototypes



- There are a range of prototyping techniques, for a range of goals
- Ideally:
  - Start with lightweight prototypes to communicate the "big picture"
  - Move to more realistic ones as risk factors are mitigated and you need to communicate about the details

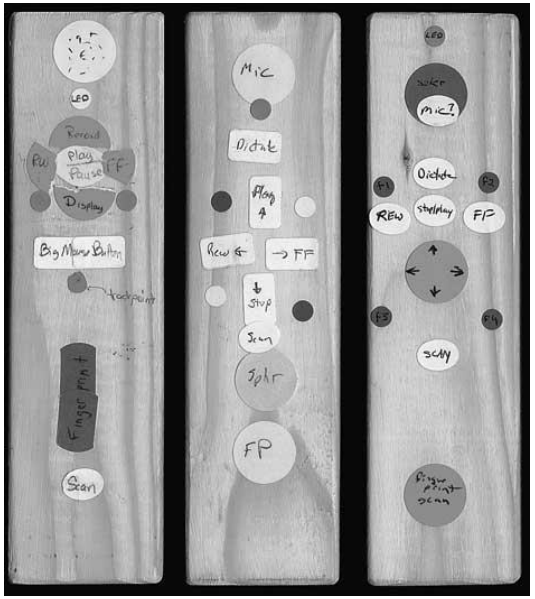
Prototyping Dimensions



- Representation
- Scope
- Fidelity

Prototype Representation

- How is design solution depicted
  - Textual
  - Storyboard/scenario with images and text
  - Video
  - Physical mock-up
  - Wireframe of screenshots
  - …



Prototype Scope

- How much functionality is depicted
  - Vertical
    - Focus on a particular slice of interaction
    - Tends to be more detailed and perhaps closer to functioning (see next point on fidelity)
  - Horizontal
    - Represent all aspects of interaction over lifetime
    - Tends to be less detailed and less functional

Prototype Fidelity

- Extent to which prototype behaves as intended system
  - How executable it is
  - Level of detail in a prototype
    - Low-fidelity: many details missing, maybe "sketchy" appearance
    - High-fidelity: prototype looks like the final system on the surface

Low-fidelity Prototyping

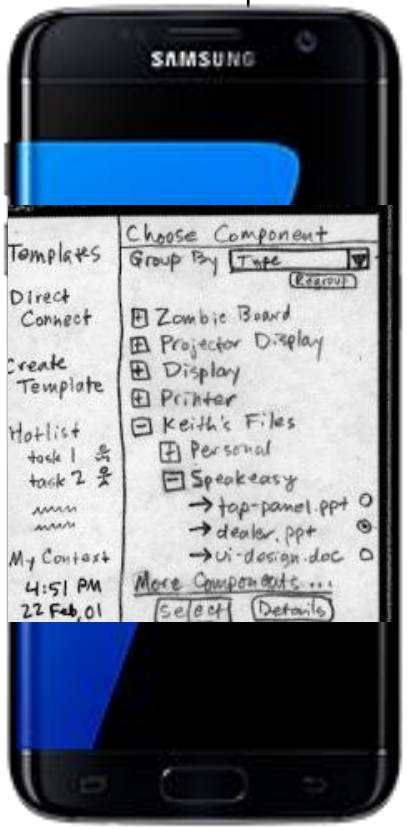
- The lowest of the lo-fi: paper prototyping
  - If you've ever designed a UI, this is probably something you've done informally
  - Capture overall layout
- Storyboards
  - From the film and animation arts
  - Capture *behavior,* not just *appearance*

Low-fidelity Prototyping

- Goal: keep the design/implement/evaluate cycle as tight as possible
- These techniques do it by keeping the implementation phase small

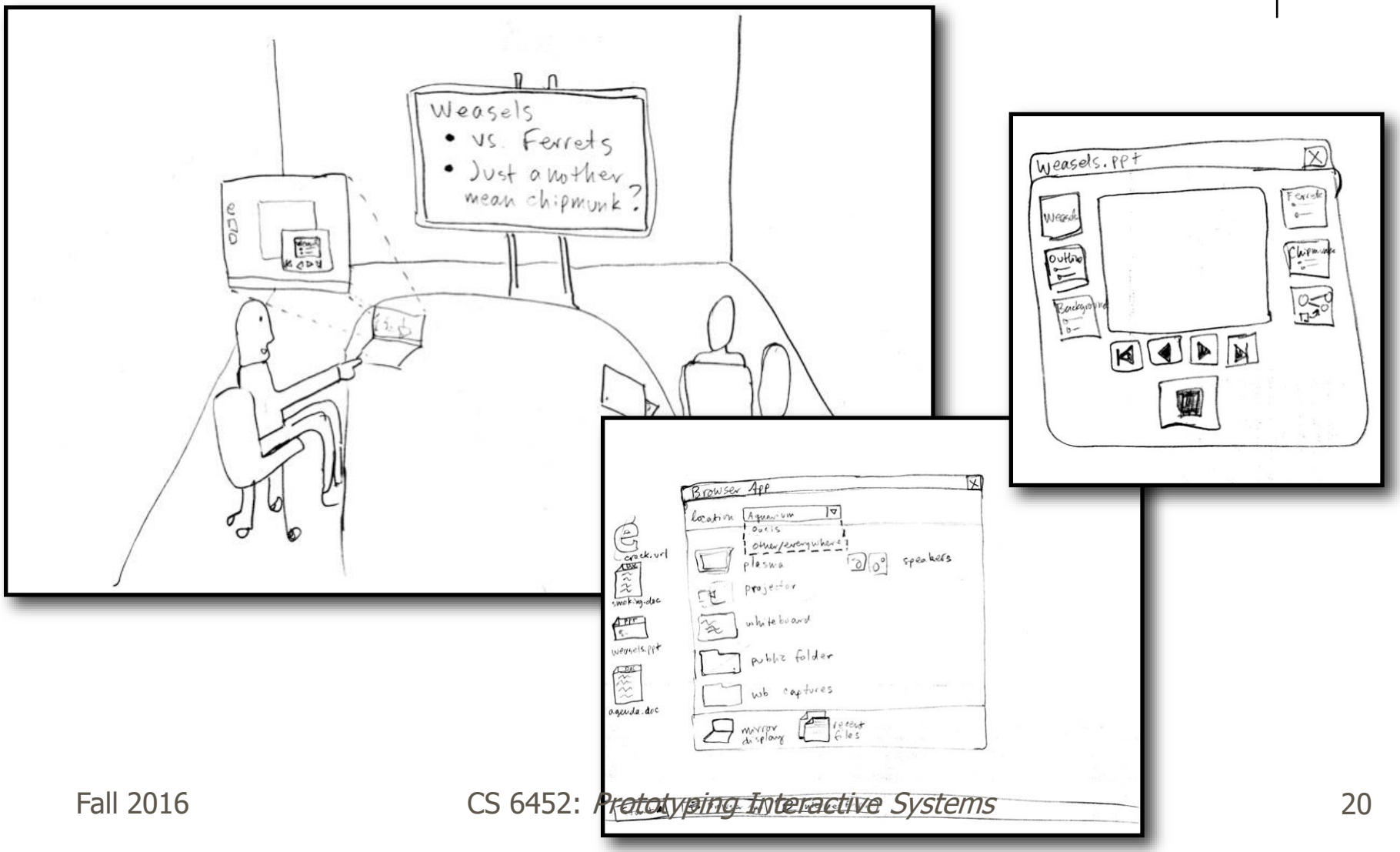
Example: Simple Paper Prototype

1. Get image of phone
2. Cut out screen area
3. Make lots of copies
4. Fill in copies as needed

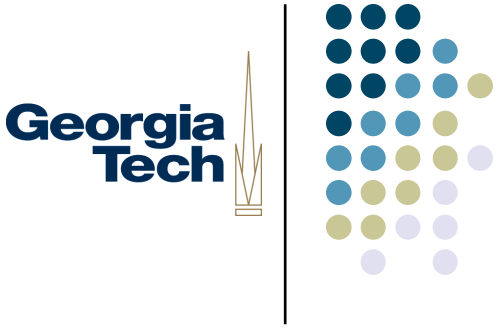


- Can be turned into storyboard
  - Annotate controls with numbers
  - Numbers lead to other sheets

A Few More Examples



Developing Low-fidelity Prototypes



- Basic tools of the trade:
  - Sketch large window areas on paper
  - Put different screen regions (anything that changes) on cards
  - Overlay cards on paper
- The copier is your friend:
  - Can easily produce many design alternatives
- Or, scan in sketches and create interactive PDF, as shown here:
  - http://boxesandarrows.com/pdf-prototypes-mistakenly-disregarded-and-underutilized/

Evaluating Low-fidelity Prototypes

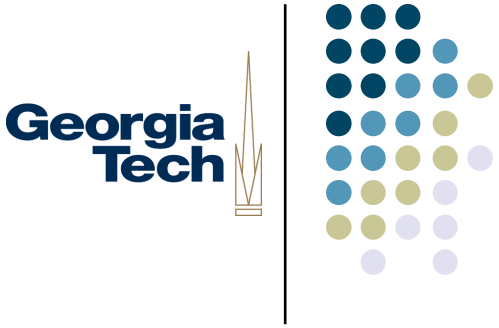
- Evaluation: You can "run" your paper prototype
  - The designer "simulates" the computer in front of a user
  - Need to be ready for any user action (drop-down menus, etc.)

High-fidelity Prototyping

- Once again, a range of practices that give you higher fidelity in exchange for higher implementation time
  - Tool-based approaches
  - GUI builders
  - Code-based approaches
- Downsides:
  - Cost is the obvious one
  - Also - Warp perceptions of the customer: elicit more comments on color, fonts, etc.

    Attending to details can lose the big picture

Tool-based Prototyping

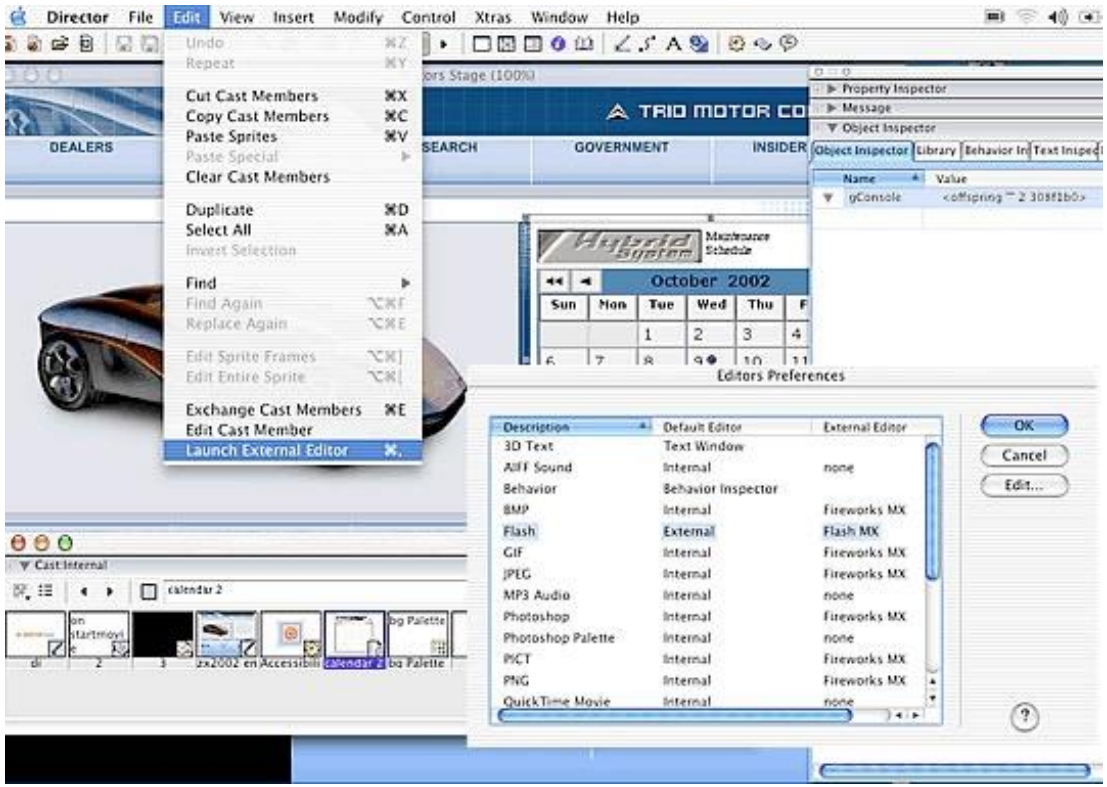


- Examples: Director, Flash, the Web
- Others?
- Pros:
  - Faster than writing code
  - Easier to incorporate changes
  - Often more reliable (hit the back button, rather than program crash)
- Cons:
  - No easy way to transition to a finished product
  - May not allow access to the full range of features available to the finished product (e.g., may not be able to prototype networking, or certain platform-specific features)

Current Tools

- Invision – web, general
- Balsamiq – general, web, mobile, wireframes
- Sketch – general, …
- Proto.io – interactive wireframes
- POP – turn paper into interactive
- Axure – images & scripting
- Visio –
- LucidChart -

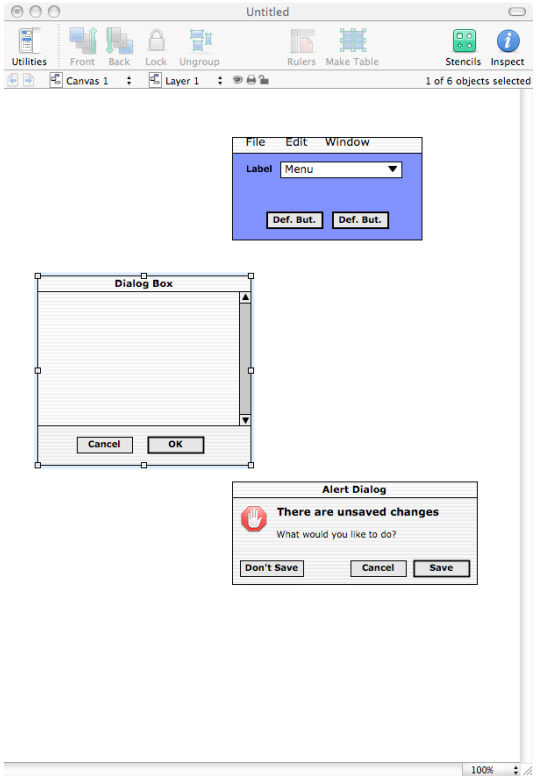
Example: Director



- Timeline editing, palettes of graphical widgets, etc.
- Emits a file that can be executed on any system that has the required runtime engine

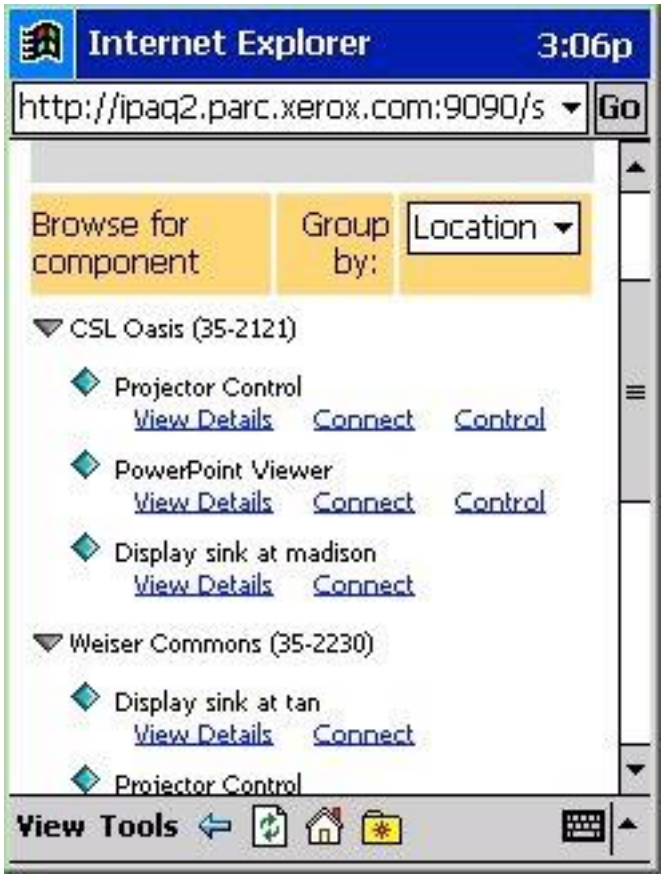
Example: OmniGraffle

- Drag graphics that depict GUI elements onto canvases
- Canvases can be linked
  - Example: Click on element A on canvas 3 goes to canvas 4
- Can emit an interactive set of web pages
- Mac only
  - Visio is at least as powerful, though



Example: Web Prototyping

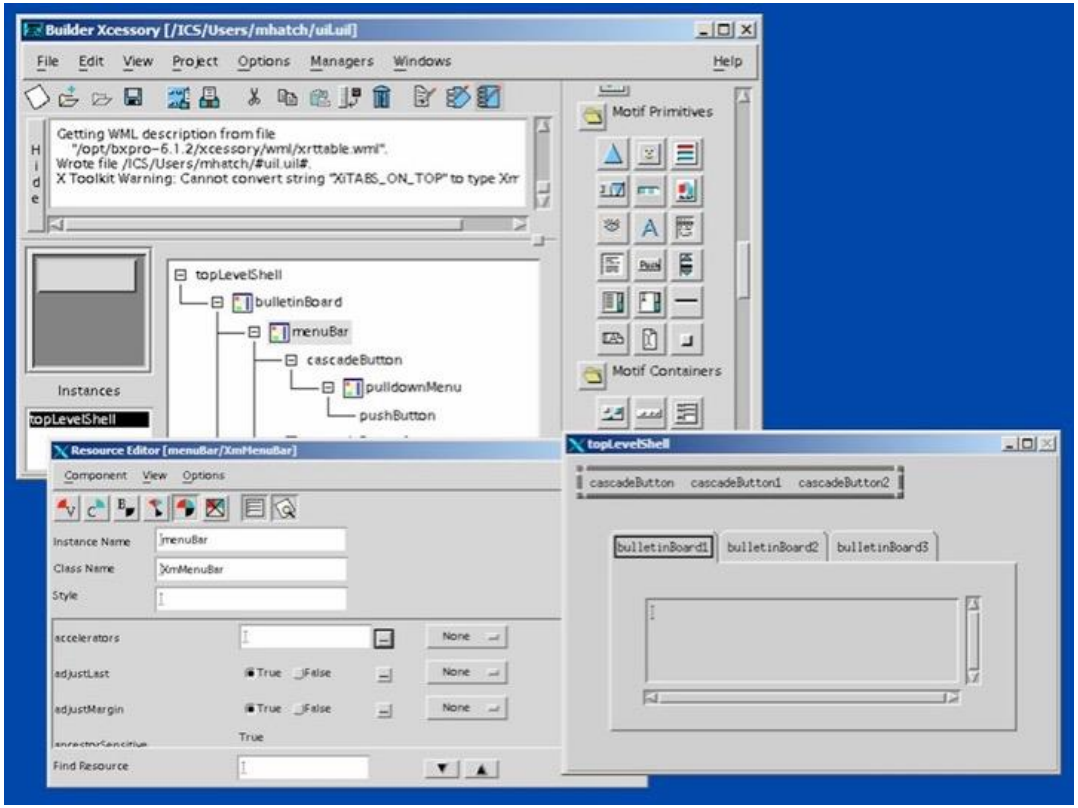
- Web-based version of lo-fi prototype shown earlier
- "Controls" simply link to another page
- Allows fine-tuning of text, graphic size, after behavior has been tested on paper
- Can be done by hand or by web development tools



GUI Builders

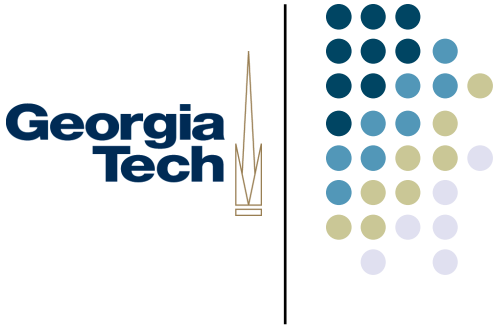
- A special class of tool for creating GUI systems
  - Drag-and-drop "widgets" from a palette
  - Emit code that you then edit: fill in the blanks
  - Most pro dev environments have them (Xcode, VS, etc)
- Pros:
  - Facilitate reasonably good transition to the final product
  - What you get looks exactly like what the finished product will look like
- Cons:
  - Still have to know a lot about programming
  - *AND* have to know about programming peculiarities in the GUI builder itself (can be very opaque)

Example: BX Pro



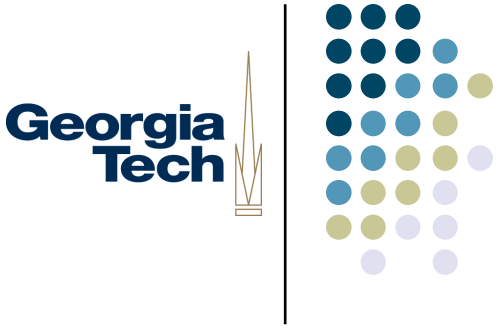
- Drag and drop graphical "widgets" onto a screen canvas
- Set properties of widgets
- Fill in C++ code for behavior

Code-based Prototypes



- Our main focus in this class
- Many approaches:
  - Production languages (Java, C++, JavaScript,etc.)
  - Scripting languages (Python, Visual Basic, AppleScript, TCL)
- There is often a fuzzy line between code and the use of tools
  - Can often "drop down" to code to augment behavior

Code-based Prototypes



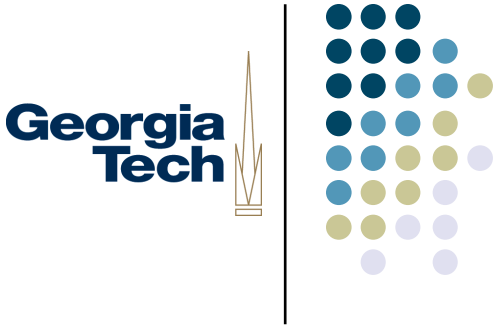
- Our main focus in this class

- Pros:
  - Very high fidelity
  - True interactivity
  - Good transition to final system
- Cons:
  - Cost, learning curve

Evaluating Hi-Fi Prototypes

- Some hi-fi prototypes are hi-fi-*enough* that standard HCI-style analyses work fine
- But what if you don't have all the necessary behaviors implemented?
- Answer: ***fake it!***

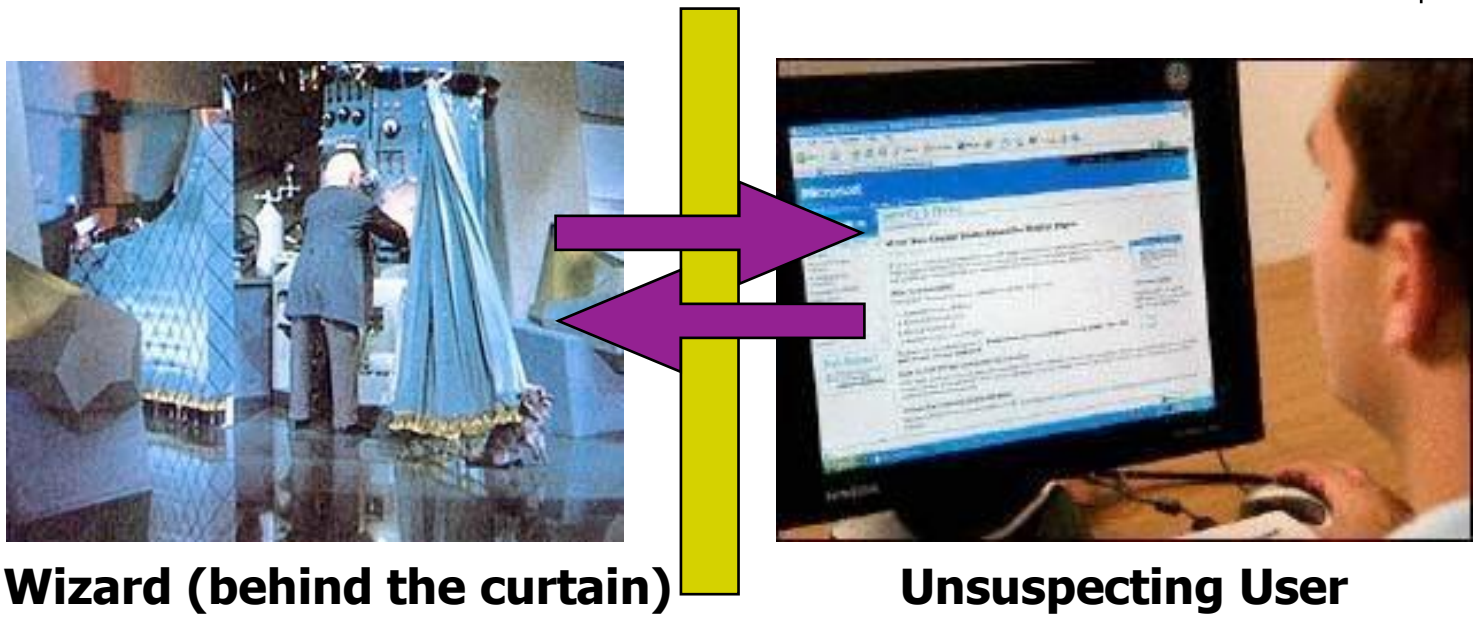
Evaluating Hi-Fi Prototypes



- *Wizard of Oz* technique
  - You are the person "behind the curtain"
  - Provide simulation of missing implementation details as necessary
  - Especially important for features that are hard to implement

    E.g., speech or handwriting recognition, activity sensing, intelligent interfaces, etc.

Example: WoZ



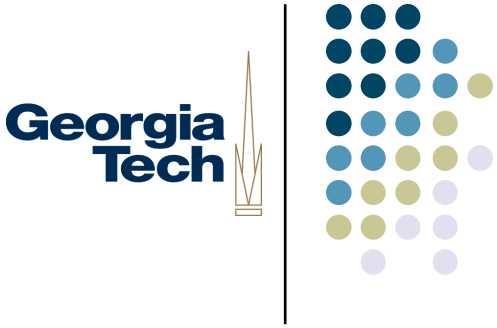
Wizard (behind the curtain) **Unsuspecting User**

- Wizard watches human input and explicitly controls the computer

Learning Objectives

- Goals of prototyping
- Dimensions of prototyping
- Prototyping techniques
- Prototyping tools

Next Time



- Back to Python
  - Reading and working with CSV, JSON datafiles