Current Research in HCI

Mobile HCI & (maybe) the Web

Jeffrey Nichols & Maneesh Agrawala
CS 160: User Interfaces

Apple Knowledge Navigator

http://www.youtube.com/watch?v=8mLqJNDWx-8

Returned Today: Pilot Usability Study

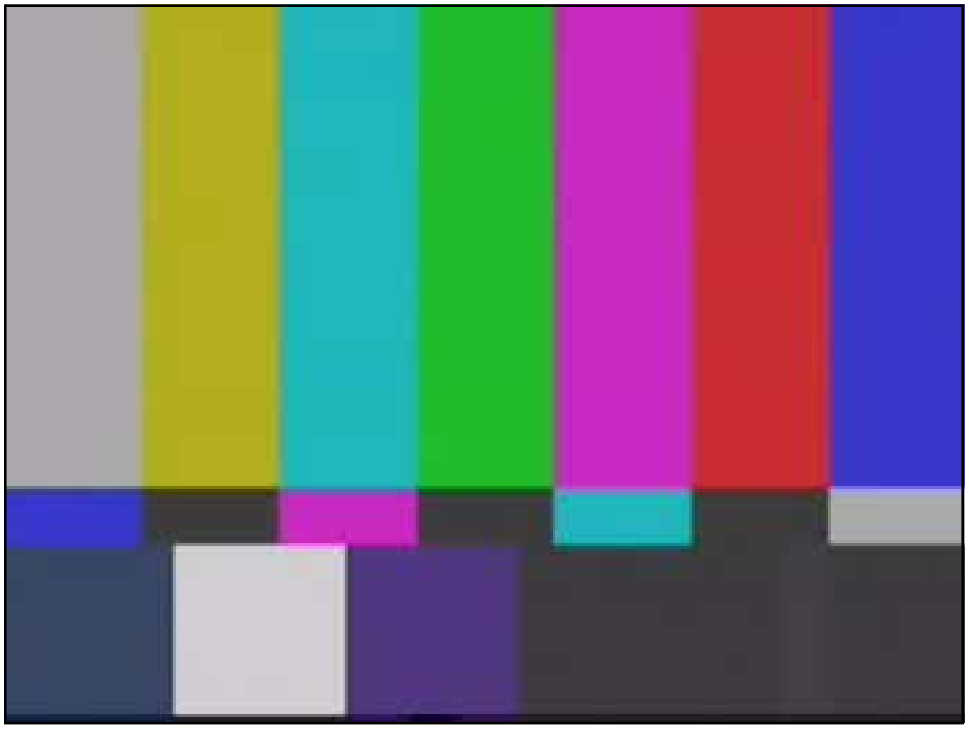
- Feedback seems to have been very useful
- Common issues with writeup
  - No explanation of what counted as errors
  - Averages of quantitative measures missing
  - No discussion of the way quantitative measures might impact design

Upcoming Schedule

Final Presentation and Report (due May 4)

- Revise interface based on pilot study
- Last chance to finish implementation
- Presentations held in Maneesh's office on May 4
  - Thank you for signing up!
- We are planning a project fair for Web May 6
  - 9-10:30am or 10-12am
  - Feel free to put new poster templates on the wiki

Mobile Computing



What's difficult with mobile devices?

- Navigation on the device
- Low resolution display
- Tiny screen
- Occlusion – finger
- Screen gets really oily
- Battery life
- Processing power
- Durability
- Lots of different form factors
- Different carriers
- Available software - # of developers, operating systems, quality of API, # of buttons (form factor)
- Storage size
- Heat

What's difficult with mobile devices?

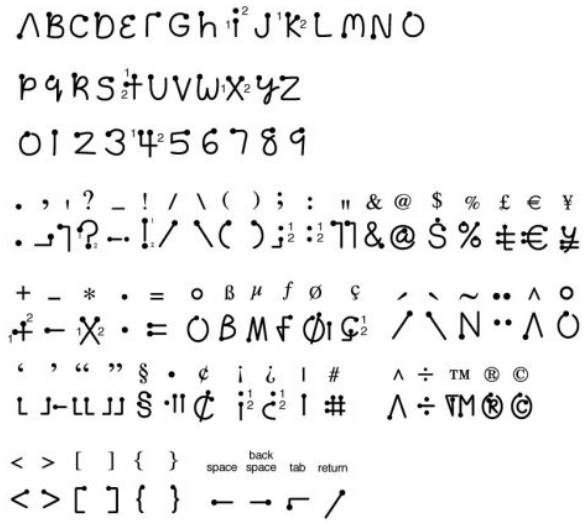
- Text entry
- Hand may hide important details on screen (touch screens only)
- Small screen size
- Interacting with/joining multiple devices
- Creating mobile user interfaces

Difficulty: Text Entry

Still very hard on mobile devices

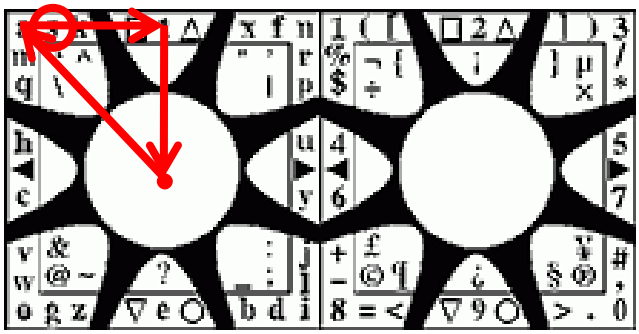
- Keyboards (on-screen and thumb)
- Full hand-writing recognition
- Graffiti
- Quikwriting
- EdgeWrite
- ShapeWriter

Graffiti – Unistroke Text Entry



Quikwriting

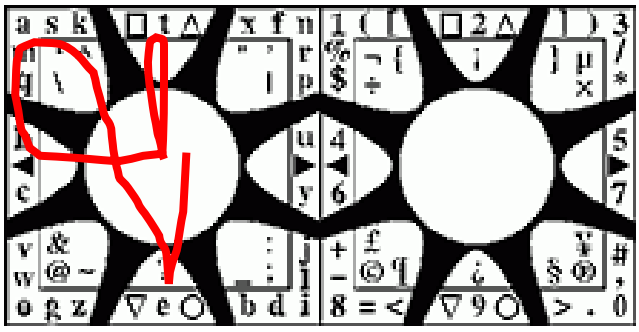
Unistroke text entry, no need to lift pen



Ken Perlin, UIST 1998

Quikwriting

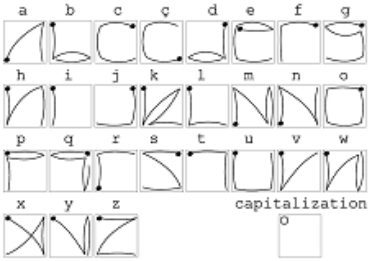
The word “the”



Ken Perlin, UIST 1998

EdgeWrite

- Corner-based text input technique
- Makes use of physical edges and corners to improve input time
- Particularly effective for users with motor impairments
  - Edges provide stability
- Implementable in many different input modalities
  - stylus, joysticks, trackball



Jacob Wobbrock, UIST 2003

ShapeWriter

- Gesture-based technique – draw a line connecting each letter
- Each gesture generates a word
- iPhone app you can try out (search for ShapeWriter)

Per Ola Kristenson & Shumin Zhai, UIST 2004

Difficulty: Hand Obscures Screen

- Can't see what button you're pressing
- Can't see drop target while dragging
- What if you could interact through the back of the device?

Video: http://www.cliftonforlines.com/videos/2007_wigdor_lucidtouch.avi

Daniel Wigdor, UIST 2007

Difficulty: Small Screen Size

- Solution: Make the screen bigger!
- How would you interact with your mobile device if it had an embedded projector?

Xiang Cao, UIST 2007

Multi-user interaction using handheld projectors

Video:
http://www.cliftonforlines.com/videos/2007_cao_multi_user_interaction_handheld.wmv

Difficulty: Interacting Across Multiple Devices

- How does a user specify devices that they would like to be connected?
- What applications make sense?
- How do programmers build services that span multiple devices?

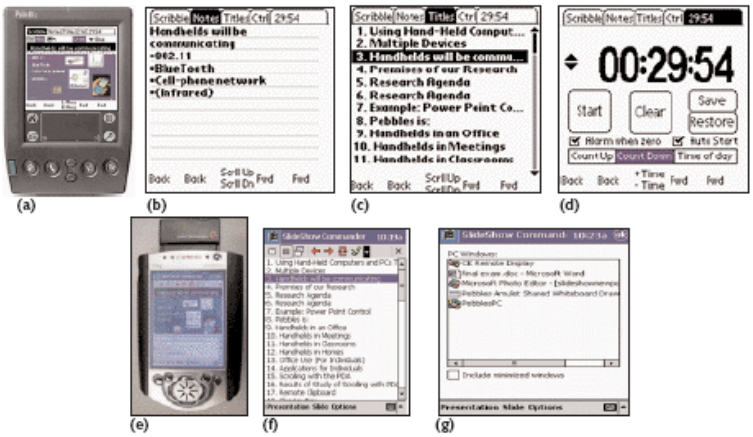
Bumping Objects Together as a Semantically Rich Way of Forming Connections, Hinckley UIST 2003

Pebbles: Using Handhelds and PCs Together

Brad Myers, et al, http://www.pebbles.hcii.cmu.edu/

Pebbles: Mobile Controlling Desktop

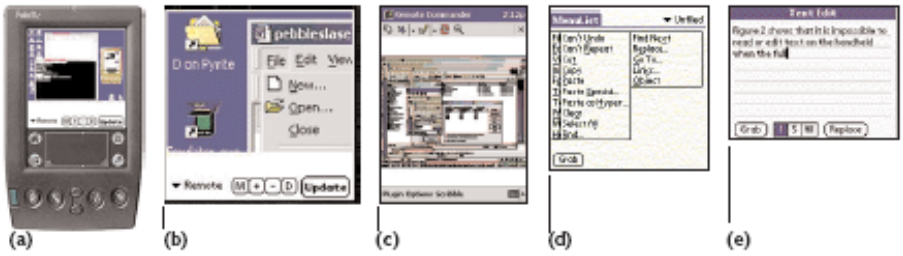
SlideShow Commander



Brad Myers, CACM 2001

Pebbles: Mobile Controlling Desktop

Remote Commander & Semantic Snarfing

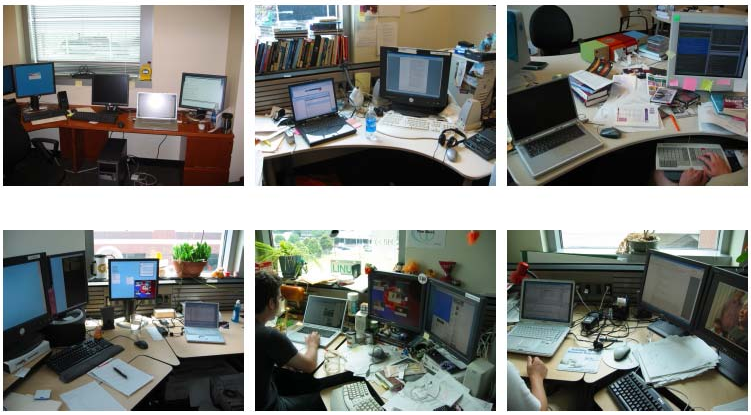


Pebbles: Scrolling

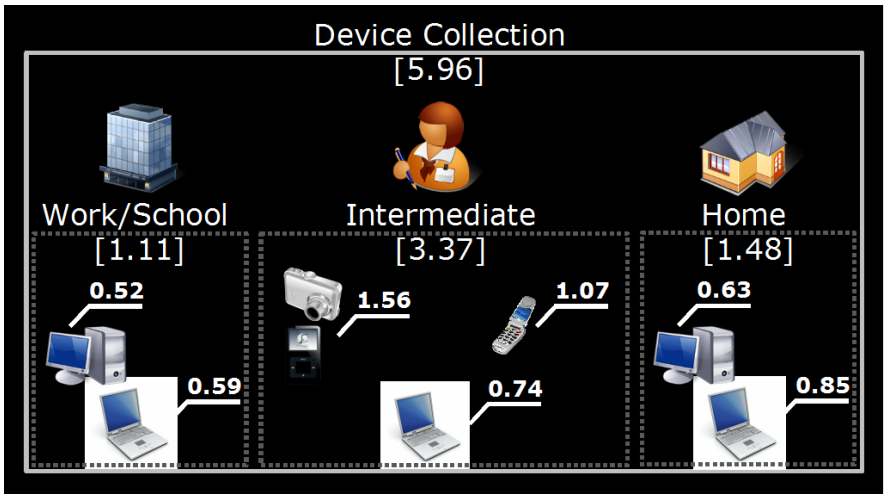
Personal Information Environments

Informal survey:

Count the number of devices that you own or use regularly



Average Device Collection



David Dearman & Jeff Pierce, CHI 2008

Problems with Connecting Devices

- Each computing device assumes that it is the user's only device
  - Some exceptions, but typically these devices assume only one other device
  - Forces device-centric interactions…we want user-centric

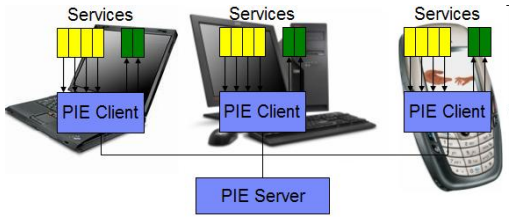
"Working across multiple devices is like collaborating with yourself"

Current Approaches

Approach	Examples	Drawbacks
User-managed	• FTP • Remote control • Sneakernet	• Burden is on user • Doesn't scale well
Logical file system	• Networked storage • Synchronization • Sneakernet	• Poor information / time granularity • No support for coordinating action
Logical computer	• Networked • Sneakernet	• Poor support for heterogeneous devices
Logical collection	• Proximate devices • Smart spaces • Meta-OS	• Promising • Current solutions are problematic

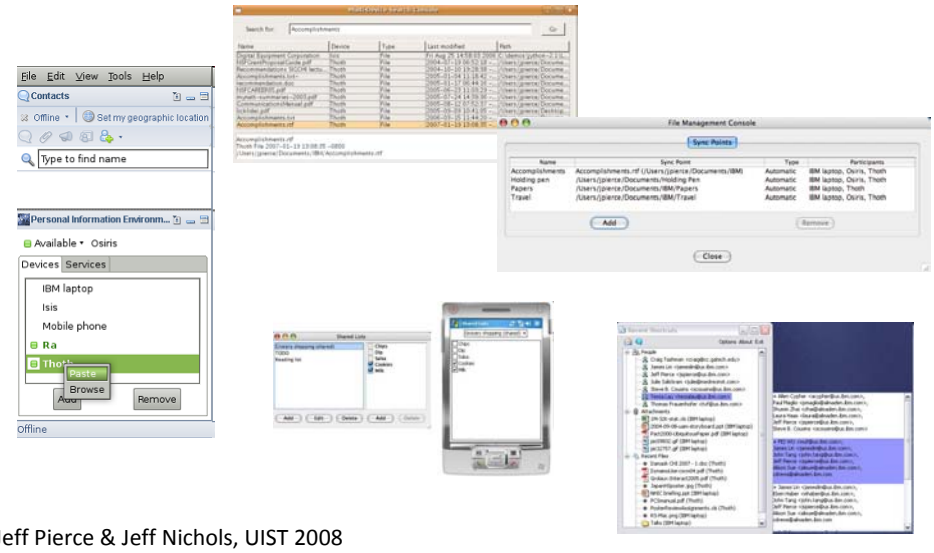
Personal Information Environment

- Logical collection: *your* devices
- Use Instant Messaging architecture as the basis for communication
  - Many benefits: named, flexible, offline support



Jeff Pierce & Jeff Nichols, UIST 2008

PIE: Applications



Jeff Pierce & Jeff Nichols, UIST 2008

Difficulty: Creating Mobile User Interfaces

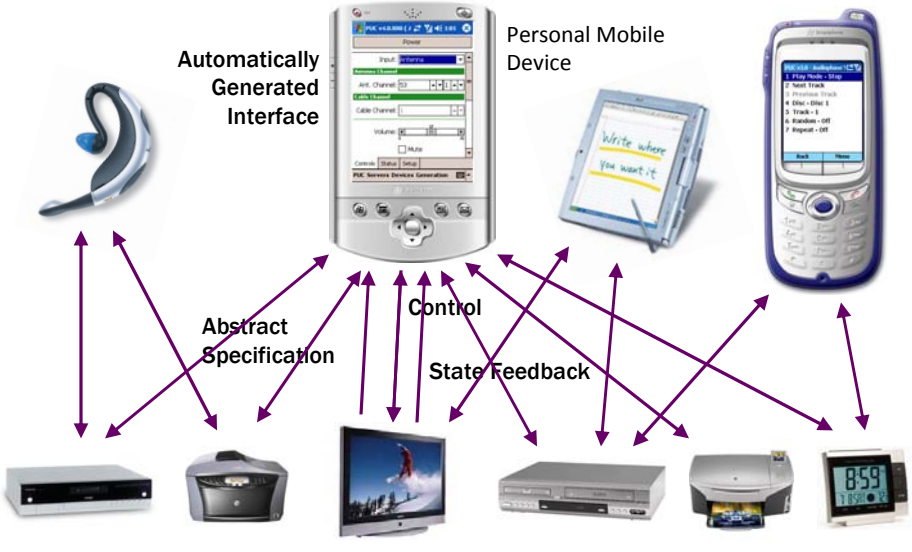
- Requires a different design from desktop applications
- Lots of heterogeneity in mobile devices
  - Many form factors – screen size and input
  - Many operating systems
  - Many user interface styles
- Can we make this any easier?

One Solution: Automatically Generate UI

- Describe the user interface abstractly once
- Generate concrete interface for each different device
- Other benefits:
  - Automatically generated interfaces can take into account other properties of user and environment

Personal Universal Controller

Personal Universal Controller



PUC Research Approach

- **Started by exploring appliance user interfaces**
  - Created handheld interfaces for two appliances
  - Found these interfaces to be better than manufacturers' interfaces
  - Analyzed the interfaces for functional information

- **Designed appliance specification language**
  - Support complete functionality of appliances
  - No specific layout information
  - Minimize work needed to write specification

```
<?xml version="1.0" encoding="utf-8" ?>
- <spec name="MediaPlayer" version="PUC/2.0">
  - <labels>
      <label>Media Player</label>
    </labels>
  - <groupings>
    - <group name="Controls" is-a="media-controls">
      - <labels>
          <label>Play Controls</label>
          <label>Play Mode</label>
          <text-to-speech text="Play Mode" recording="play
        </labels>
```

- **Built interface generators**
  - Uses dependency information to infer structure of user interface
  - Smart Templates allow rendering of high-level design conventions

Specification Language

- **Language**
  - Functional information
  - XML-based
- **Elements**
  - State variables & commands
  - Group tree
  - Multiple labels per object
  - Dependency information

```
<?xml version="1.0" encoding="utf-8" ?>
- <spec name="MediaPlayer" version="PUC/2.0">
  - <labels>
      <label>Media Player</label>
    </labels>
  - <groupings>
    - <group name="Controls" is-a="media-controls">
      - <labels>
          <label>Play Controls</label>
          <label>Play Mode</label>
          <text-to-speech text="Play Mode" recording="playmode.au" />
        </labels>
      - <state name="Mode">
        - <type>
          - <enumerated>
              <item-count>3</item-count>
            </enumerated>
          - <valueLabels>
            - <map index="1">
                <label>Stop</label>
              </map>
            - <map index="2">
                <label>Play</label>
              </map>
            - <map index="3">
                <label>Pause</label>
              </map>
            </valueLabels>
          </type>
        - <labels>
            <label>Mode</label>
          </labels>
        </state>
      - <group name="TrackControls">
        - <command name="PrevTrack">
          - <labels>
              <label>Prev</label>
            </labels>
          - <active-if>
              <greater-than state="PList.Selection">0</greater-than>
            </active-if>
```

Full documentation available at:
http://www.pebbles.hcii.cmu.edu/puc/

Jeff Nichols, UIST 2002

Interface Generation

- Graphical Interface Generators
  - Runs on today's handheld devices
  - .NET Compact Framework 2.0
- Speech Interface Generator
  - Masters work of Thomas Harris
  - Built on top of the PUC framework
  - Implemented using Universal Speech Interface (USI) techniques
    [Rosenfeld 2001]

Jeff Nichols, UIST 2002

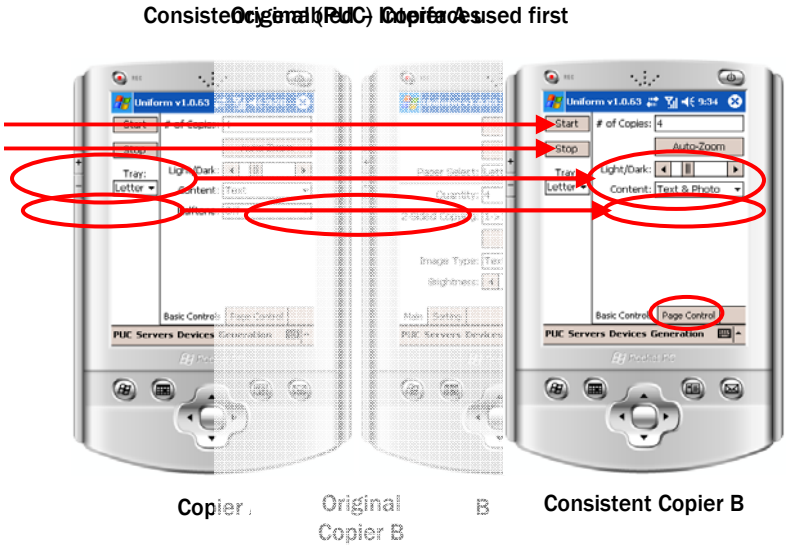
Generating Personally Consistent UI

Original (PUC) Interfaces

Copier A **Copier B**

Jeff Nichols, CHI 2006

Generating Personally Consistent UI



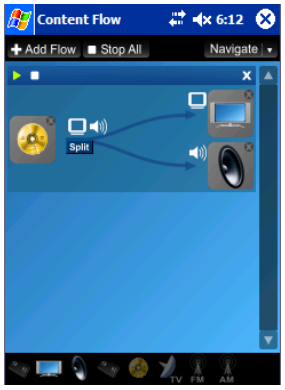
Jeff Nichols, CHI 2006

Multi-Appliance Interfaces

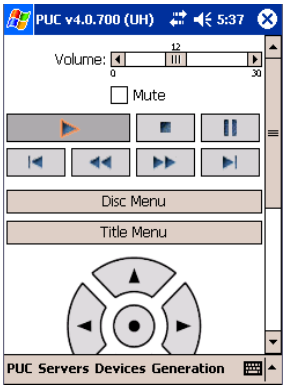
How can we address this problem?

- Generate task-based interfaces that span all appliances
- Need to know how appliances are used together
  - … in a scalable way

Generating Aggregate User Interfaces



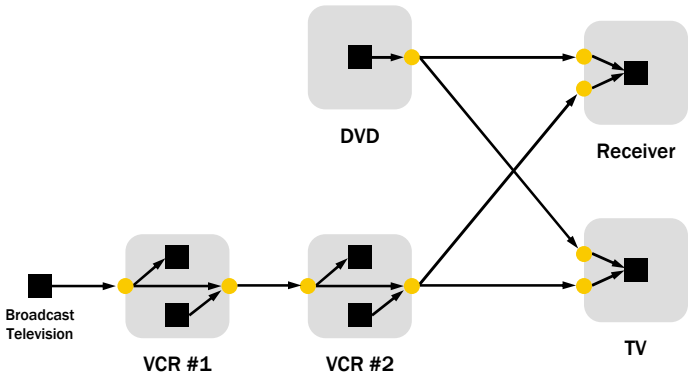
Flow-Based Interface



Aggregate User Interfaces

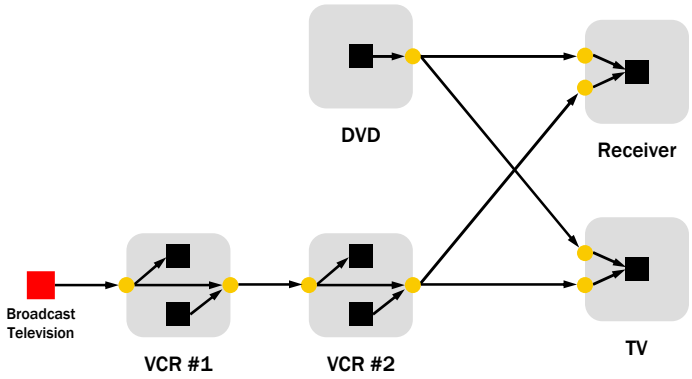
Content Flow Model

Home Theater



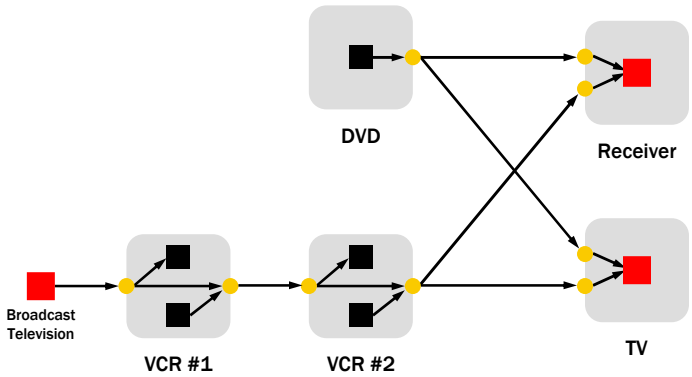
Modeling Tasks

Home Theater: Watching Television



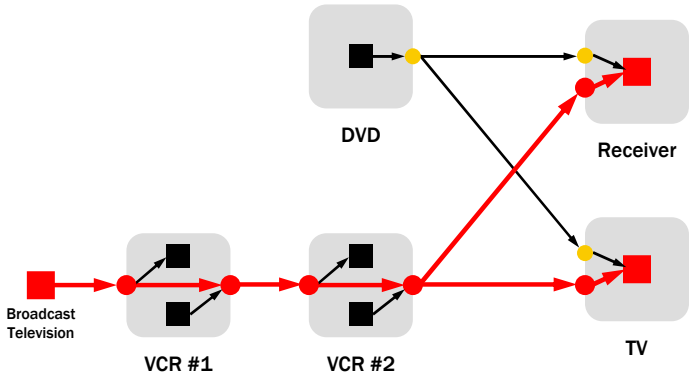
Modeling Tasks

Home Theater: Watching Television



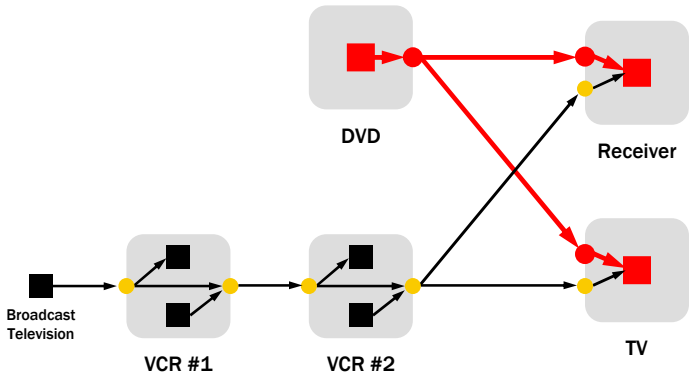
Modeling Tasks

Home Theater: Watching Television



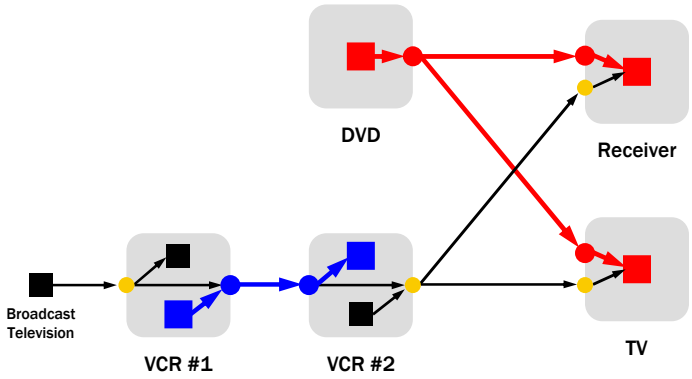
Modeling Tasks

Home Theater: Watch DVD



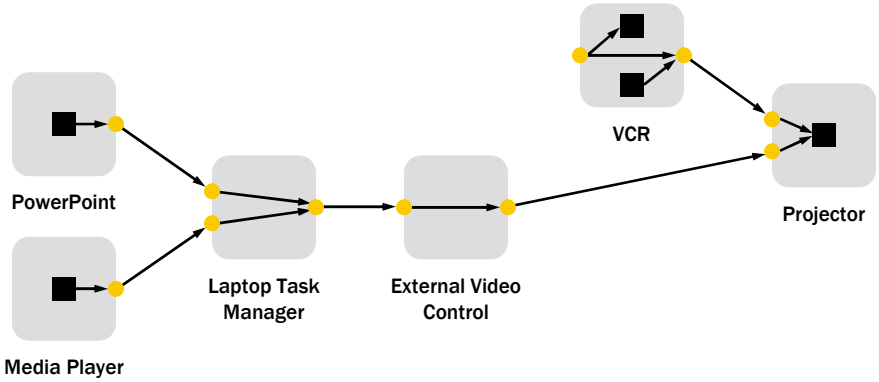
Modeling Tasks

Home Theater: Watch DVD + Copy Tape



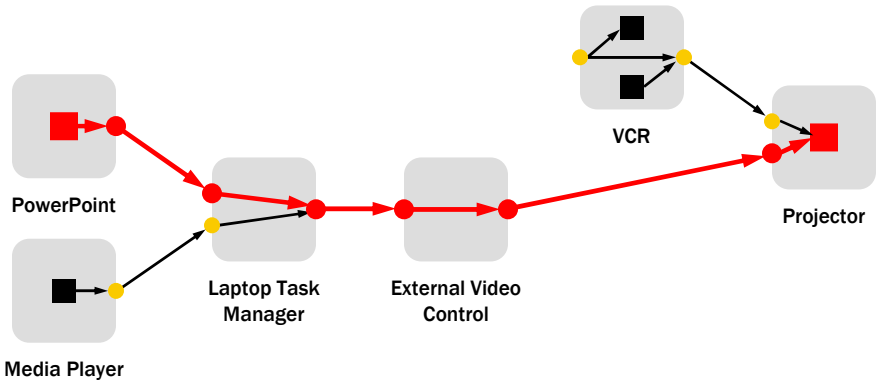
Modeling Tasks

Presentation Room

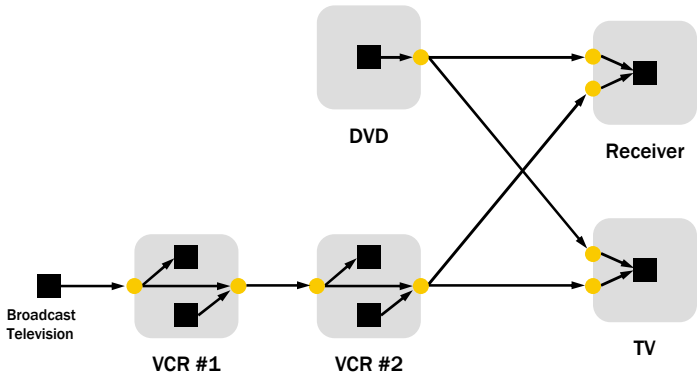


Modeling Tasks

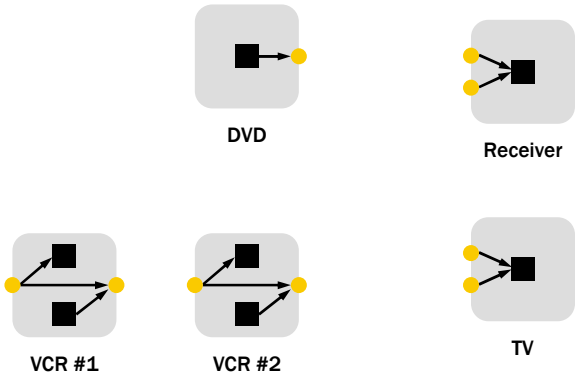
Presentation Room: Presenting PowerPoint



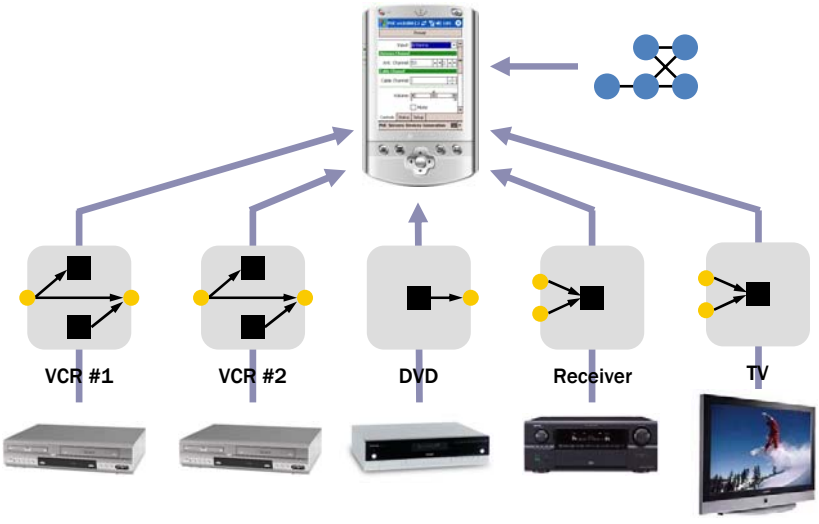
Where does the model come from?



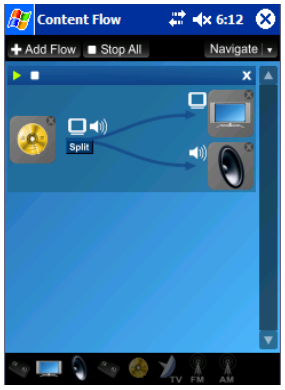
Where does the model come from?



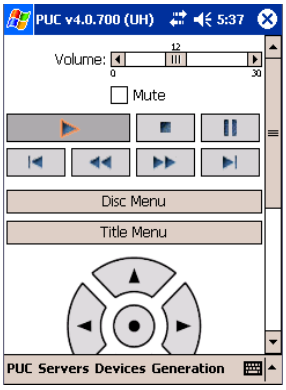
Where does the model come from?



Generating Aggregate User Interfaces



Flow-Based Interface



Aggregate User Interfaces

Personal Universal Controller

How good is it?

- Interfaces are clearly not aesthetically pleasing
- However, it is competing against relatively poor interfaces…

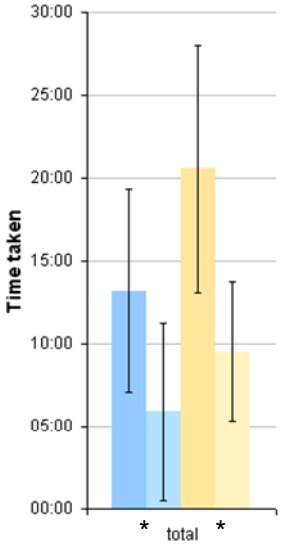
Usability

Evaluation of Generated Interfaces

- Users perform tasks with two all-in-one printer interfaces (HP and Canon)
- Two comparisons
  - Physical interface to PUC interface
  - Without consistency and with consistency
- 48 participants (28 male, 20 female)
  - Divided into 6 groups, 8 per group
  - Recruited through CBDR web page
- Metrics
  - Completion time
  - Failures

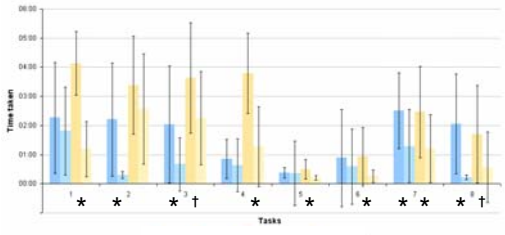
HP printer Canon printer

Usability Results



	#1 - HP	#2 - Canon
Built-In	Physical	Physical
AutoGen	PUC	PUC
Consistent	PUC	Uniform

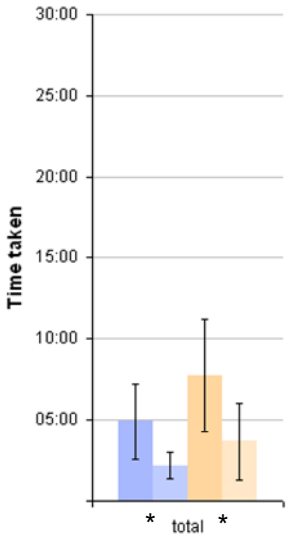
	#1 - Canon	#2 - HP
Built-In	Physical	Physical
AutoGen	PUC	PUC
Consistent	PUC	Uniform



- Failures per subject also significantly less for PUC than Physical (Fisher's Exact Test, $p < 0.05$)
  - HP: 1.125 Built-In vs. 0.125 PUC
  - Canon: 2.0 Built-In vs. 0.625 PUC

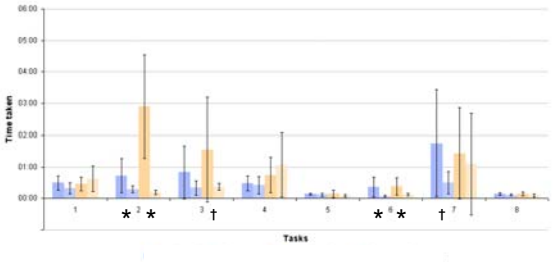
* = $p < 0.05$ † = $p < 0.1$

Consistency Results



	#1 - HP	#2 - Canon
Built-In	Physical	Physical
AutoGen	PUC	PUC
Consistent	PUC	Uniform

	#1 - Canon	#2 - HP
Built-In	Physical	Physical
AutoGen	PUC	PUC
Consistent	PUC	Uniform



- Very few failures in these conditions: 4 total for all tasks performed by 32 subjects (256 tasks, 1.5% failure rate)

* = p < 0.05 † = p < 0.1

PUC Conclusion

- Automatic generation of usable interfaces for end users is possible
  - Automatic design may be viable for some applications
- Automatic generation allows interfaces to be customized to the:
  - user's previous experience
  - user's environment
  - more?

Highlight:
Mobilizing Existing Web Sites

Jeffrey Nichols
IBM Almaden Research Center

Accessing the Mobile Web

- Site designed for mobile use
  - Designed for low-end devices
  - Limited functionality chosen by designer
  - Costly to create
    - Only available for popular, consumer sites

- Normal site through a mobile viewport
  - Most functionality of existing site
  - Greater costs of navigation
    - Many items per page
    - More pages than needed

Previous Work: Transcoding

Traditional proxy server techniques

- Mostly automated
  - www.skweezer.com
  - mobile google search
- Quality of result varies based on site
- Often includes all content of a page

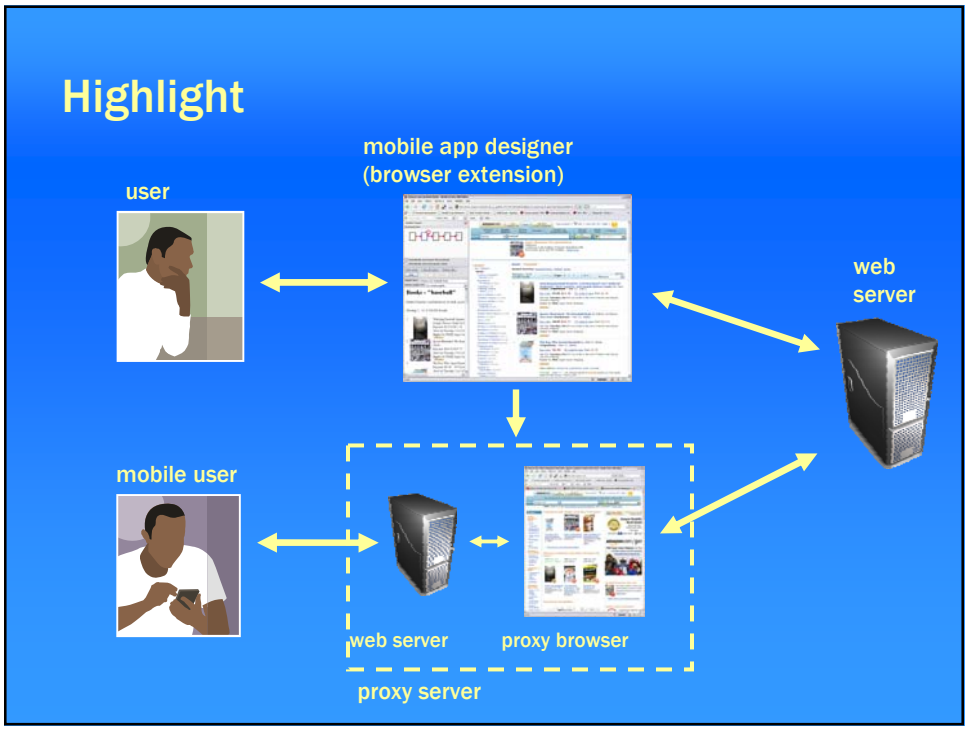
Doesn't work with AJAX/dynamic JavaScript sites

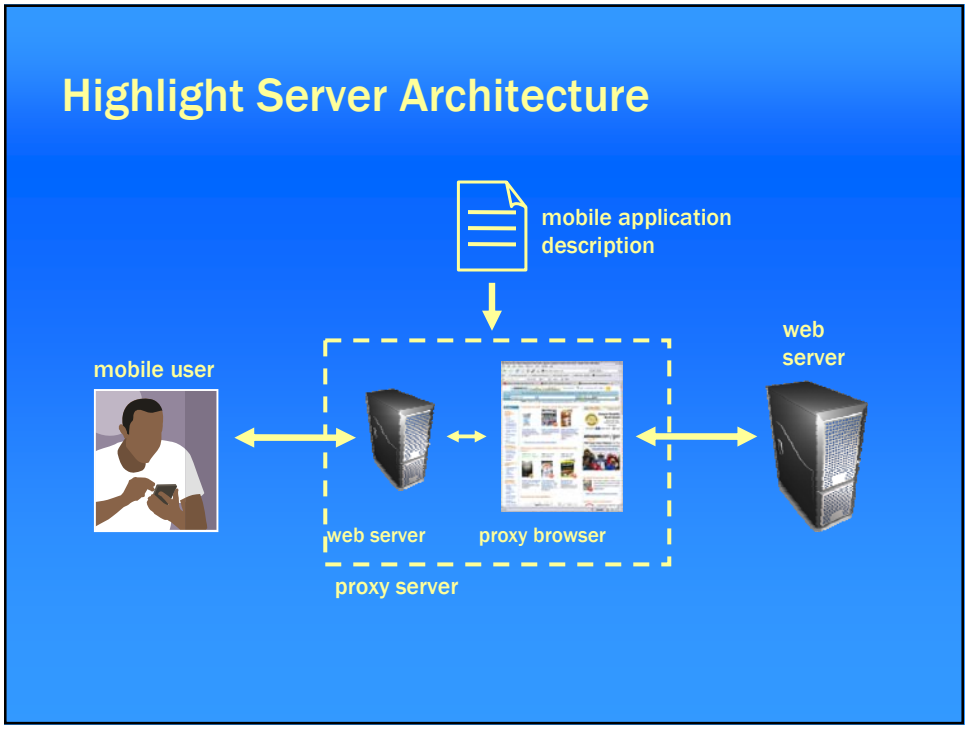
Goals

Allow end users to create their own mobile "applications" for particular tasks

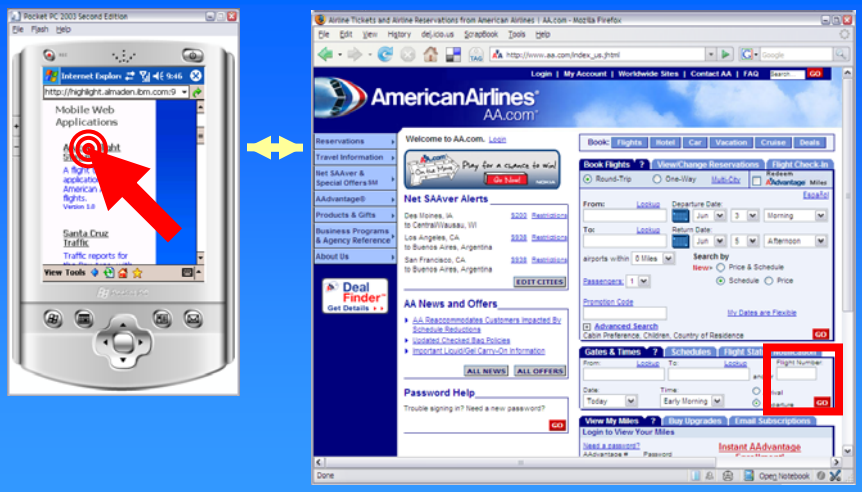
- No programming required
- Possible for any existing site
- All design decisions made by users

Allow programmers to extend capabilities of mobile applications

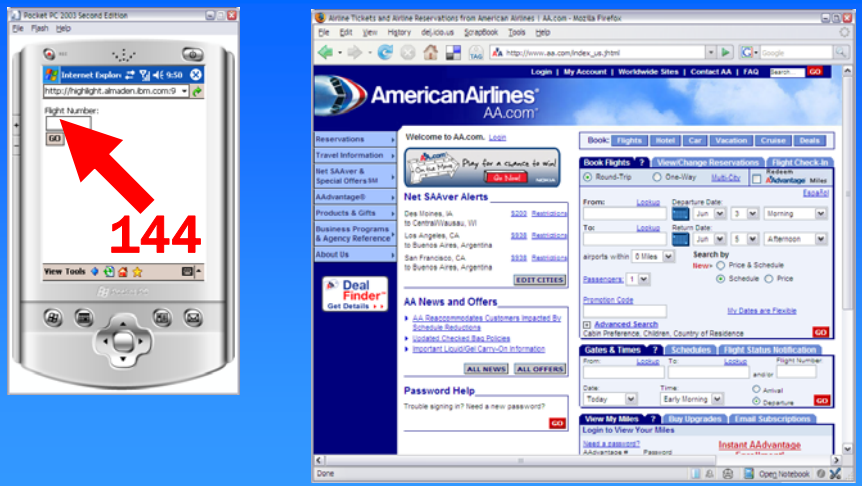

Highlight
mobile app designer
(browser extension)
user
web
server
mobile user
web server
proxy browser
proxy server


Highlight Server Architecture
mobile application
description
web
server
mobile user
web server
proxy browser
proxy server

Remote Control Metaphor

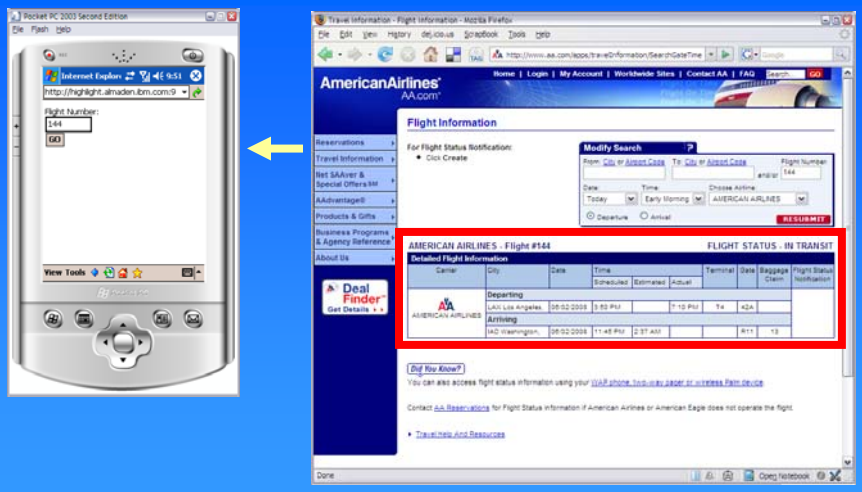


Remote Control Metaphor



Remote Control Metaphor

Remote Control Metaphor



Why use this architecture?

Allows re-authoring of sites with dynamic JavaScript and Ajax

Re-authoring can be done in terms of UI design

- Easier to inspect than code, Web Service descriptions

Other Projects

If time remaining, I will demo…

- CoScripter
- PlayByPlay
- ActionShot

Next Time: No Lecture!!

Group Project Presentations in Maneesh's Office (Soda 635)

9:00-9:15	Group G
9:20-9:35	HCiiight
9:40-9:55	TGTGSH
10:00-10:15	Interface Galactica
10:20-10:35	31337
10:40-10:55	NGRS
11:00-11:15	That's What She Said
11:20-11:35	WAAM!
11:40-11:55	B Pour G
1:00-1:15	PALS
1:20-1:35	Epicware
1:40-1:55	Group O
2:00-2:15	OverDos
2:20-2:35	CalUI
2:40-2:55	Clyk