“Re-authoring” User Interfaces

Jeffrey Nichols
IBM Research – Almaden

October 1, 2010
University of Rochester
IBM Research – Almaden

- 400+ research employees; 100+ students and post-docs
- Research in Computer Science, Storage Systems, Science and Technology, Services Science
  - User Focused Systems in CS
An increasing variety of interactive devices...
Perceptive Pixel Wall Display

Microsoft Surface

interactive device

About 9,150,000 results (0.50 seconds)
All of these devices need user interfaces...

Where will they come from?
1. Code new user interfaces from scratch

2. Recode existing user interfaces

3. “Re-author” existing user interfaces
Thanks a billion.

Over 1 billion downloads in just nine months.
Only on the App Store. Learn more→

The Billion App Countdown promotion has ended.
Please come back to this page soon to see who won.
Official Rules | Reglas Oficiales→
1. Code new user interfaces from scratch

2. Recode existing user interfaces

3. “Re-author” existing user interfaces
ConCurTaskTrees & TERESA, Paterno et al.

Personal Universal Controller, Nichols et al.

Mobi-D, Puerta et al.

SUPPLE, Gajos et al.
1. Code new user interfaces from scratch
2. Recode existing user interfaces
3. “Re-author” existing user interfaces
“Re-authoring” ...

- may be done by end users
- is done only at the interactive level
- specifies transformations from the original interface to a new interface design
- uses a “remote control” metaphor
Example: AA.com – Flight Tracking
AA.com Flight Tracking: Mobilized

Flight Number:

GO

FLIGHT STATUS - ARRIVED

<table>
<thead>
<tr>
<th>Carrier</th>
<th>City</th>
<th>Date</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMERICAN AIRLINES</td>
<td>LAX Los Angeles,</td>
<td>09/29/2007</td>
<td>2:30 PM</td>
</tr>
<tr>
<td></td>
<td>Arriving</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IAD</td>
<td>09/29/2007</td>
<td>10:20 PM</td>
</tr>
<tr>
<td></td>
<td>Washington,</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Highlight, Nichols et al. IUI 2008, UIST 2008
User: “What is the status for my American Airlines flight?”

System: “What is the flight number?”

User: “144”

System: “Flight Status – Arrived”
AA.com Flight Tracking: Reusing Design

TX2, Bigham et al. UIST 2009
Remote Control Metaphor
144
Benefits

• Working at the interactive level lets authors work with what they can “see.”

• No need to understand underlying code or describe application with complex models

• Possible for end users, extensible by programmers

• If easy enough, allows users to create user interfaces that reflect their own needs and abilities
Drawbacks

- Always running original interface in the background
- Constrained by original design
- How to communicate those constraints to the author?
Highlight:
Mobilizing Existing Web 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
How do end users create applications?

Highlight Designer

- Built using Firefox web browser
- Allows user to demonstrate a “trace” of interaction
- Direct manipulation tools
- Generalization allows creation of mobile apps with complex structure

Nichols & Lau, IUI 2008
Demonstration
Ceiling on End User Authoring

- Complex page manipulations
- Sites with Ajax/dynamic JavaScript
- Adding new functionality
- Mashing up multiple sites
Highlight Server Architecture

- mobile user
- mobile application description
- web server
- proxy server
- proxy browser
- mobile application

Nichols, Hua, Barton, UIST 2008
Mobile Application Descriptions

Requirements
- Support end user authoring environment
- Increase possibilities through programming

Implementation
- Structured JavaScript
- Built on top of standard web APIs (e.g., DOM)
- Set of Highlight API methods to make programming with remote control metaphor easier
Writing Application Descriptions

End-user authoring environment generates JavaScript code in this format
- Programmers can edit this code to modify an app

We have written several apps from scratch
- BackpackIt.com To Do List
- BlueMail
- Fitday.com
Highlight Evaluation

- Breadth of possible applications
- Formative evaluations of authoring (end user and programming)
- Code analysis
Breadth of possible applications

Created applications from a variety of different sites

- AA.com
- Amazon.com
- Google Image Search
- Mapquest.com
- SFGate.com
- Weather.com
- Ebay
- Fitday.com
- Buy.com
- Barnes & Noble.com
- IBM internal directory (BluePages)
- ...
Formative evaluation – End User

Goals
- Will users understand how to interact with the system?
- Can they create applications of their own?

Three subjects from our research lab
- Not regular users of mobile web
- All able to recall instances in which they wished they could access the web in a mobile setting
Procedure

• Brief verbal introduction to the system

• Asked users to create two mobile applications of our choice:
  – Mapquest.com
  – Google Image Search

• Asked users to brainstorm their own mobile app idea and then try to implement it
Results

• All subjects were able to build the first two applications

• Definite learning curve
  – When is human interaction required vs. when will the system do something smart?
  – Users formed a clearer model with use
Results, cont.

Subject-chosen applications:

- SF Chronicle’s Bargain Bites
- Weather.com 10-day Forecast
- www.beatthetrafic.com
Formative evaluation – Programmer

Goals

- Will programmers understand the concepts/API?
- Can they create applications of their own?

Two subjects from our research lab

- Not regular users of mobile web
- Experienced web programmers (JavaScript, DOM, etc.)
Procedure

• 1 hour verbal introduction to the system
• Provided with programmer guide
  – Complete BackPack todo list example with source code
  – API reference
• Provided with choice of authoring/debugging tools
  – Eclipse Web Tools Platform editor, Firebug, Venkman
• Asked users to brainstorm their own mobile app idea and then try to implement it
Results

Subject-chosen applications:

- CBS 5 Traffic Reports
- Facebook Status
Results, cont.

- All subjects were able to build their chosen application
- Typical problems of learning a new API
  - What methods are available? How are they used?
- Conceptual model of remote control not immediately intuitive
- Clip/event method grouping does not match expectations of web programmers
# Mobile App Code Numbers

<table>
<thead>
<tr>
<th>Application Name</th>
<th># of Pagelets</th>
<th>Lines</th>
<th>Average Lines/Pagelet</th>
<th>Average Clip Lines/Pagelet</th>
</tr>
</thead>
<tbody>
<tr>
<td>BluePages*</td>
<td>5</td>
<td>353</td>
<td>69.40</td>
<td>44.80</td>
</tr>
<tr>
<td>Google Image Search*</td>
<td>3</td>
<td>130</td>
<td>41.00</td>
<td>27.00</td>
</tr>
<tr>
<td>AEO Store Locator*</td>
<td>2</td>
<td>81</td>
<td>37.00</td>
<td>27.50</td>
</tr>
<tr>
<td>BlueMail</td>
<td>3</td>
<td>127</td>
<td>36.33</td>
<td>22.00</td>
</tr>
<tr>
<td>AA Flight Tracker*</td>
<td>3</td>
<td>112</td>
<td>35.33</td>
<td>22.33</td>
</tr>
<tr>
<td>Home Depot Store Finder*</td>
<td>2</td>
<td>73</td>
<td>33.00</td>
<td>24.00</td>
</tr>
<tr>
<td>Traffic #2*</td>
<td>1</td>
<td>33</td>
<td>26.00</td>
<td>26.00</td>
</tr>
<tr>
<td>Traffic #1</td>
<td>3</td>
<td>58</td>
<td>18.33</td>
<td>15.67</td>
</tr>
<tr>
<td>Facebook Status</td>
<td>4</td>
<td>59</td>
<td>14.75</td>
<td>7.50</td>
</tr>
<tr>
<td>BackPack To-Do</td>
<td>3</td>
<td>46</td>
<td>14.33</td>
<td>5.67</td>
</tr>
</tbody>
</table>

* Indicates some code automatically generated by our end-user authoring tool
Conclusion

Mobile applications can be created by re-authoring existing web sites

- Demonstrational interface is quite powerful
- Proxy browser/remote control metaphor allows re-authoring of sites with AJAX and dynamic JavaScript
- Evaluation demonstrates end users and programmers can create mobile applications
- Low floor and high ceiling
CoCo:
A Conversational Interface to the Web
The CoCo research vision

- Explore the use of conversational user interfaces to web-based tasks
- *Design* and *build* intelligent agents that:
  - Interact with the web on a user's behalf
  - Converse with the user to clarify meaning
  - Learn new knowledge over time
  - Are personalized for a user's needs
- Goal: improve user productivity and increase access to information technology through simpler interfaces
CoCo does things for you on the web

And tells you the result

user @coco4532 get employee phone number for name Jeffrey W Nichols
2 minutes ago

coco4532 @user 1-408-927-1172
less than 10 seconds ago
Alice: punch out 17 30
CoCo: Extracted this script from your logs:
Go to timecard.com/cocompany, enter your password into the textbox, click Go...
Run it?
Alice: yes
CoCo: I don't know what “password” to use
Alice: punch out 17 30 using alice00 as password
CoCo: I will run your script using params
password=alice00
CoCo: 17:30 Exit
Two paths to determining process:

Automatic

- System finds existing script in database or infers script from web history
- Content is clipped based on heuristics matching original command

Manual ("re-authoring")

- User creates a script in CoScripter
- Specifies parameters as “personal database” values
- Specifies “clip” commands to return information
Re-authoring with CoCo

Current Situation
- Record in CoScripter
- Xpaths generated by user (e.g. Firebug)

Future Work
- What’s a better design?
- Can it be an interactive “teaching”-like process?
Conclusion

• An intelligent conversational assistant for the web
• Performs tasks for users thru a simple textual interface
• Can determine how to execute tasks without explicit teaching, using a database of scripts and the user’s web history
• Can be explicitly taught through a “re-authoring” process, though more refinement work is needed
“Re-authoring” Conclusions
Re-authoring

- Designing on top of the existing interface works
- Highlight and CoCo enable re-authoring the web in different ways
- Re-authoring possible by End Users, and extensible by Programmers
Future Work

• Design of re-authoring tools
• Re-authoring to very different interaction styles
• Partial re-authoring
• Bridge from re-authoring to re-coding
Re-authoring to very different interaction styles
Future Work

- Design of re-authoring tools
- Re-authoring to very different interaction styles
- Partial re-authoring
- Bridge from re-authoring to re-coding
Thanks!

jwnichols@us.ibm.com
http://www.jeffreynichols.com/
IBM Research – Almaden

- 400+ research employees; 100+ students and post-docs
- Research in Computer Science, Storage Systems, Science and Technology, Services Science
  - User Focused Systems in CS
Set of Content Operations

- Clip <item> from browser page into mobile page
- Move item within mobile page
- Delete item from mobile page
- Add new content to mobile page (e.g. label)
Transition Event

- Interaction on the mobile page that advances to next mobile page
- Corresponding interactions to execute in the browser to reach content for next mobile page
- Pointer to next set of content operations
## Breadth and Benefits

<table>
<thead>
<tr>
<th>Description</th>
<th>Interactive Elements</th>
<th>Size (kB)</th>
<th>Percent Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check status of AA flight</td>
<td>736/3</td>
<td>711/3.6</td>
<td>0.5%</td>
</tr>
<tr>
<td>Update Facebook status</td>
<td>217/5</td>
<td>296/0.5</td>
<td>0.2%</td>
</tr>
<tr>
<td>Find nearby Wi-fi hotspot</td>
<td>74/18</td>
<td>1072/2.8</td>
<td>0.3%</td>
</tr>
<tr>
<td>Get weather in my area</td>
<td>486/6</td>
<td>1079/7</td>
<td>0.6%</td>
</tr>
<tr>
<td>Sprint cellphone usage</td>
<td>175/6</td>
<td>739/4.6</td>
<td>0.6%</td>
</tr>
<tr>
<td>Log today’s exercise</td>
<td>128/4</td>
<td>393/0.9</td>
<td>0.2%</td>
</tr>
<tr>
<td>Update Fitday food diary</td>
<td>169/38</td>
<td>145/12.7</td>
<td>8.8%</td>
</tr>
<tr>
<td>Get calories for food</td>
<td>88/16</td>
<td>63/11.5</td>
<td>18.3%</td>
</tr>
<tr>
<td>Real estate in my area</td>
<td>274/35</td>
<td>1036/194.1</td>
<td>18.7%</td>
</tr>
<tr>
<td>Show trip itineraries</td>
<td>77/17</td>
<td>726/42.7</td>
<td>5.9%</td>
</tr>
<tr>
<td>Find Amazon book price</td>
<td>823/4</td>
<td>844/4.1</td>
<td>0.5%</td>
</tr>
</tbody>
</table>
Highlight Programmed Applications
Common Page Structures

- Query/results/detail
- Multiple result pages
- Diverge/converge
- Query/error/result
Link generalization, cont.
Link generalization, cont.

\[\text{HTML[1]/.../TBODY[1]/TR[2]/TD[1]/A[1]}\]

\[\text{HTML[1]/.../TBODY[1]/TR[1]/TD[1]/A[1]}\]
Link generalization, cont.

/HTML[1]/.../TBODY[1]/TR[*]/TD[1]/A[1]
Common Page Structures

- Query/results/detail
- Multiple result pages
- Diverge/converge
- Query/error/result
Mobile Page Re-use
Mobile Page Re-use
Mobile Page Re-use
Mobile Page Re-use
Common Page Structures

- Query/results/detail
- Multiple result pages
- Diverge/converge
- Query/error/result
Common Page Structures

- Query/results/detail
- Multiple result pages
- Diverge/converge
- Query/error/result
Testing for Different Content
Testing for Different Content
Testing for Different Content
Mobile Internet
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
Highlight Designer

Storyboard View

Toolbar

Preview Browser

Main Browser Area
Creating complex structures

Additional traces allow for:
- Branching
- Looping
- Generalization
Some Supported Page Structures

- Query/results/detail
- Multiple result pages
- Diverge/converge
- Query/error/result
Implementation

mobile users

proxy server

web servers
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
Structured JavaScript Format

Based on Storyboard concept

Code divided into chunks corresponding to mobile page (pagelet)

Each pagelet has two methods
- Clip method – selects content from page to clip
- Event method – navigates to next content based on mobile interaction
Application Descriptions

pagelet n-1

clip

pagelet n

clip

event

clip

event

pagelet n+1

clip

server
To-Do List

- go to the store
- more stuff
- eat a pear
- Feed the dog
- Feed the dog
- Make my UIST talk
- Make my UIST talk

Add this item or Close
BackpackIt.com – To Do List

Username:
Password:
(forgot password?)
Sign in

To-Do List

- Feed the dog
- go to the store
- eat a pear
- walk the dog

Add this item or Close

Update To Do Items

View Tools
Deployment Conclusions

• Proxy browser/remote control metaphor powerful for authoring and deployment

• Allows for modification of sites using AJAX or dynamic JavaScript

• Scalability of the server architecture needs to be tested
Empirical study of apps

Compared the Highlight applications with their existing web sites in terms of:

- Number of interactive elements (links, form elements, etc.)
- Size of total pages downloaded (kB)
## Benefits – Interactive Elements

<table>
<thead>
<tr>
<th>Description</th>
<th>Interactive Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check status of AA flight</td>
<td>736</td>
</tr>
<tr>
<td>Update Facebook status</td>
<td>217</td>
</tr>
<tr>
<td>Find nearby Wi-fi hotspot</td>
<td>74</td>
</tr>
<tr>
<td>Get weather in my area</td>
<td>486</td>
</tr>
<tr>
<td>Sprint cellphone usage</td>
<td>175</td>
</tr>
<tr>
<td>Log today’s exercise</td>
<td>128</td>
</tr>
<tr>
<td>Update Fitday food diary</td>
<td>169</td>
</tr>
<tr>
<td>Get calories for food</td>
<td>88</td>
</tr>
<tr>
<td>Real estate in my area</td>
<td>274</td>
</tr>
<tr>
<td>Show trip itineraries</td>
<td>77</td>
</tr>
<tr>
<td>Find Amazon book price</td>
<td>823</td>
</tr>
</tbody>
</table>
## Benefits – Transmission Size

<table>
<thead>
<tr>
<th>Description</th>
<th>Original</th>
<th>Highlight</th>
<th>Percent Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Check status of AA flight</td>
<td>711</td>
<td>3.6</td>
<td>0.5%</td>
</tr>
<tr>
<td>Update Facebook status</td>
<td>296</td>
<td>0.5</td>
<td>0.2%</td>
</tr>
<tr>
<td>Find nearby Wi-fi hotspot</td>
<td>1072</td>
<td>2.8</td>
<td>0.3%</td>
</tr>
<tr>
<td>Get weather in my area</td>
<td>1079</td>
<td>7</td>
<td>0.6%</td>
</tr>
<tr>
<td>Sprint cellphone usage</td>
<td>739</td>
<td>4.6</td>
<td>0.6%</td>
</tr>
<tr>
<td>Log today’s exercise</td>
<td>393</td>
<td>0.9</td>
<td>0.2%</td>
</tr>
<tr>
<td>Update Fitday food diary</td>
<td>145</td>
<td>12.7</td>
<td>8.8%</td>
</tr>
<tr>
<td>Get calories for food</td>
<td>63</td>
<td>11.5</td>
<td>18.3%</td>
</tr>
<tr>
<td>Real estate in my area</td>
<td>1036</td>
<td>194.1</td>
<td>18.7%</td>
</tr>
<tr>
<td>Show trip itineraries</td>
<td>726</td>
<td>42.7</td>
<td>5.9%</td>
</tr>
<tr>
<td>Find Amazon book price</td>
<td>844</td>
<td>4.1</td>
<td>0.5%</td>
</tr>
</tbody>
</table>
CoCo Backup
Mining web history logs

- Leverage a recorded history of everything a user has done on the web via ActionShot/CRH

<table>
<thead>
<tr>
<th>Task</th>
<th>Time</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>go to &quot;google.com&quot;</td>
<td>03:31</td>
<td>google.com</td>
</tr>
<tr>
<td>click the &quot;Shopping&quot; link</td>
<td>03:32</td>
<td><a href="http://www.google.com/">http://www.google.com/</a></td>
</tr>
<tr>
<td>enter &quot;Biere DualFusion&quot; into the &quot;Search Prod...&quot;</td>
<td>03:34</td>
<td><a href="http://www.google.com/prdh...">http://www.google.com/prdh...</a></td>
</tr>
<tr>
<td>click the &quot;Search&quot; button</td>
<td>03:34</td>
<td><a href="http://www.google.com/prdh...">http://www.google.com/prdh...</a></td>
</tr>
<tr>
<td>go to &quot;google.com&quot;</td>
<td>03:35</td>
<td>google.com</td>
</tr>
<tr>
<td>enter &quot;how to flip a video upside down&quot; into the &quot;Search Prod...&quot;</td>
<td>03:35</td>
<td><a href="http://www.google.com/">http://www.google.com/</a></td>
</tr>
<tr>
<td>go to &quot;amazon.com&quot;</td>
<td>03:35</td>
<td>amazon.com</td>
</tr>
<tr>
<td>enter &quot;iPad&quot; into the second &quot;Search for&quot; textbox</td>
<td>03:35</td>
<td><a href="http://www.amazon.com/">http://www.amazon.com/</a></td>
</tr>
<tr>
<td>click the second &quot;Go&quot; button</td>
<td>03:35</td>
<td><a href="http://www.amazon.com/">http://www.amazon.com/</a></td>
</tr>
</tbody>
</table>

- Segment stream based on:
  - Time between actions (5 mins)
  - Switching websites

- Generate a set of segments, one for each task
Finding the right script

- CoCo searches CoScripter scripts and ActionShot segments to find one that satisfies the query
- Search uses vector-space model and TF/IDF to rank possible scripts
- Interact with user to supply missing parameters
- Confirm that this script is the one the user wants
Generating a response

- How does CoCo decide what information to return to the user?
  - Manual: user provides explicit `clip` command
    - User has full control over which content is returned
    - Enables authoring of complete CoCo experience
  - Automatic: system guesses which portion of web page to clip and return
    - Requires no additional authoring, e.g. for ActionShot scripts
    - Relies on heuristics to select a page region for clipping
Automatic clipping

- Given a web page, which region best represents the result of the script?
- Use geometric heuristics\(^1\) to identify candidate regions on the page
- Use word overlap to rank regions relative to the user's query
- Return highest-ranked region

\(^1\)[Mahmud et al, CSurf: A Context-Driven Non-Visual Web Browser, WWW 2007]
User study: will users trust automation?

- Would users trust a system such as CoCo to automate their web tasks?
  - Compare to existing automation systems: iMacros, Automation Anywhere, CoScripter

- Designed experiment to compare understandability of textual (script) vs visual (screenshot) feedback
  - 1219 Mechanical Turkers completed study

- Hypothesis: the reduced information in the textual representation does not hamper user understanding of automation effects
Example task

Goal
Check the status of American Airlines Flight 279

Users saw one or more of these three visualizations

They were asked whether the visualization(s) they saw accomplished the goal

We computed how many people answered the question correctly for each condition

1. Static screenshots (like video, no animation)
   * go to "aa.com"
   * enter "279" into the "Flight Number:" textbox
   * click the fifth "GO" button
User study results

Except for condition #4, all conditions had similar success rates

Conclusion: text-only (#1) is comparable to existing automation approaches with visual feedback
Summary

• We present CoCo, an intelligent conversational assistant for the web
• CoCo performs tasks for users thru a simple textual interface
• An automatic clipping algorithm guesses what response to return to the user as a result of execution
• A user study shows that users understand CoCo automation as well as traditional visual automation
Future work

- Teach CoCo how to do a new task that it hasn't seen before in scripts or web history
- Design an authoring environment specifically to teach CoCo how to do tasks on the web
- Improve the clipping algorithm by leveraging more implicit user feedback such as mouse hovering
- Model user preferences and become more personalized over time