



New tools for modeling for HCI evaluation

Bonnie E. John
Konstantine (Gus) Prevas

ACT-R Workshop 26jul03



Acknowledgements

- ONR
 - Bonnie John, CMU
 - Rick Lewis, University of Michigan
 - Mike Matessa, Roger Remington, Alonso Vera, NASA Ames
 - Ken Koedinger, Gus Prevas, & colleagues, CMU
- General Motors
 - Bonnie John, Gus Prevas, CMU
- Dario Salvucci & Frank Lee, ACT-Simple



New Tools for Modeling for *HCI Evaluation*

- *NOT* tools for *cognitive modeling*
 - Targeted at HCI evaluation models, like GOMS or KLM
- *NOT* tools for *cognitive modelers*
 - UI designers are the target audience
- But you may want to use these tools for your work or teaching, which is why I'm presenting them here



Some History

- NASA-sponsored Masters of HCI project two years ago identified 5 areas that made cognitive modeling hard (3 of which I remember)

- Understanding what a complex model is doing and why

Trace tool prototypes, Summer 2001

- Building and maintaining an understanding of what the modeling community is doing

Community website for Apex, Summer 2001

- Building the “world” that the model interacts with

Dreamweaver-based WorldBuilder for Apex, Summer 2002, Apex Tutorial CogSci 2002



Our user group: UI designers

- Often do not know how to program, nor do they want to
- Do not know how to build a model, nor do they want to (too much like programming???)
- So the cost must be low and the benefit high, for them to adopt predictive cognitive modeling as a UI design tool



UI designers

- Often use html to create an interactive storyboard of their ideas
- Are comfortable using commercially-available WYSIWYG html tools like DreamWeaver or GoLive.



So... knowing our users...

Our goals are to

1. Remove any activity that hints at programming
2. Exploit the use of html tools
3. Quickly produce useful, valid predictive models



1. Remove any activity that hints at programming

- Use drag-and-drop to build the world
- Use demonstration to build the model



2. Exploit the use of html tools

- Extend Macromedia DreamWeaver to include a special widget palette
 - Build any html storyboard with this palette
- When a human demonstrates a task using these widgets, (as UI designers already do to demonstrate their ideas to others), the actions are recorded
- The actions are post-processed to construct a cognitive model



3. Quickly produce useful, valid predictive models

- Focus on models that have been successfully used (though painstakingly constructed) in HCI in the past:
 - GOMS, KLM, CPM-GOMS
 - Inherit their usefulness, validity
 - Predict performance time (initially)
- ACT-Simple is a KLM-like language that compiles into ACT-RPM productions (Salvucci & Lee, 2003)



DreamWeaver Extensions

The screenshot shows the DreamWeaver interface with the 'Recording' palette open. The palette contains various widgets, including a cell phone icon. A red box highlights the palette, and a red arrow points to it from the text 'Dreamweaver palette of widgets to drag and drop to create the mockup in html'. Another red box highlights the cell phone icon, and a red arrow points to it from the text 'html mock-up of an interface. In this case, a cell phone'. The background shows a desktop with various files and folders, including 'link2.html', 'cellphone.html', 'test.html', 'phone_number.html', 'Macintosh HD', 'GM.dmg', 'logictemplate Folder', 'OSXvnc', 'Truth Table demo.html', 'Anon.doc', 'demo.html', 'link1.html', 'WebSim', and 'atm.html'.

Dreamweaver palette of widgets to drag and drop to create the mockup in html

html mock-up of an interface. In this case, a cell phone



Demonstrate a task

Grab File Edit Capture Window Help 58° Fri 10:41 AM

javac

Behavior Recorder for file untitled

File Edit Display Tutor Skills Windows Help

Demonstrate Mode Tutor Mode Pseud

Cellphone

[-1], [8]

state1

[-1], [6]

state2

[-1], [7]

state3

[-1], [5]

Cellphone - Netscape

file:///Macintosh%20HD/Users/kp

As you demo the task in the html mock-up...

the BehaviorRecorder traces the demo.

System says:

User says:

Applet Loaded

atm.html



Export to ACT-Simple

The screenshot shows the **pact.StateFrame.BehaviorRecorder** application window. The **File** menu is open, highlighting the **Export...** option (⌘E). A red arrow points from this option to the text **Export item in the File menu**.

The main window displays a state machine diagram with the following states and transitions:

- state1** → **-1,2-bul** → **state2**
- state2** → **-1,3-bul** → **state3**
- state3** → **-1,send-** → **state4**
- state4** → **Hello, speed** → **state5**

The **Cellphone Demo - Netscape** window shows a mobile phone image and the text "Audio input to system: Hello".

The desktop background is a blue gradient with various icons, including PDF files, folders, and system utilities.



Export to ACT-Simple

Grab File Edit Capture Window Help

Behavior Recorder for file untitled

File Edit Display Tutor Windows Help

☒ Demonstrate Mode ☐ Tutor Mode ☐ Pseudo-Tutor Mode

Cellphone Demo

state1

state2

state3

state4

state5

Cellphone Demo - Netscape

file:///Macintosh%20HD/

to ACT-Simple Commands

Export Behavior

NoCodingDemo

Name	Date Modified
actsimple.lisp	Jul 25, 2003 9:56 AM
BehaviorRecorderApplet.class	Jul 25, 2003 9:56 AM
Cell for Roger	Jul 10, 2003 8:38 PM
cellphoneDemo.html	Jun 26, 2003 9:11 AM
CellphoneDemoACTsimple	Dec 31, 1969 7:53 PM

Name: GellphoneTask1

Format: ACT-Simple Behavior

Cancel Save



ACT-Simple code resulting from demonstration

```
(klm-p klm (klm-goal klm
  (look-at "1-button")
  (move-mouse "1-button")
  (click-mouse)
  (look-at "2-button")
  (move-mouse "2-button")
  (click-mouse)
  (look-at "3-button")
  (move-mouse "3-button")
  (click-mouse)
  (look-at "send-button")
  (move-mouse "send-button")
  (click-mouse)
  (speak "hello")
))
(pm-start-hand-at-mouse)
(pm-set-params :output-speech t)
```



ACT-Simple command set and translation to ACT-RPM

(move-hand [position])

(move-mouse)

(click-mouse)

(press-mouse)

(release-mouse)

(press-key [key])

(speak [string])

(look-at)

(listen [time])

(think [time])

Table 2: Sample translation of (press-key *a*).

```
(p do-task-press-key-102
  =goal>
    isa do-task
    state 1
  =manual-state>
    isa module-state
    modality free
==>
  +manual>
    isa press-key
    key a
  =goal>
    state 2
)
```




Load ACT-Simple into ACT-R and run!

The screenshot shows the ACT-R environment with several windows. The Behavior Recorder window is in the foreground, showing a sequence of actions. The Listener window is open, displaying a list of actions. The procedural1 window is open, showing a list of actions. The ACT-R environment is running on a Macintosh HD, as indicated by the desktop icons and the system menu bar.

ACT-RPM makes time predictions

ACT-RPM sees widgets and presses buttons

ACT-Simple compiles into ACT-RPM productions



What ACT-RPM “sees” on the html storyboard

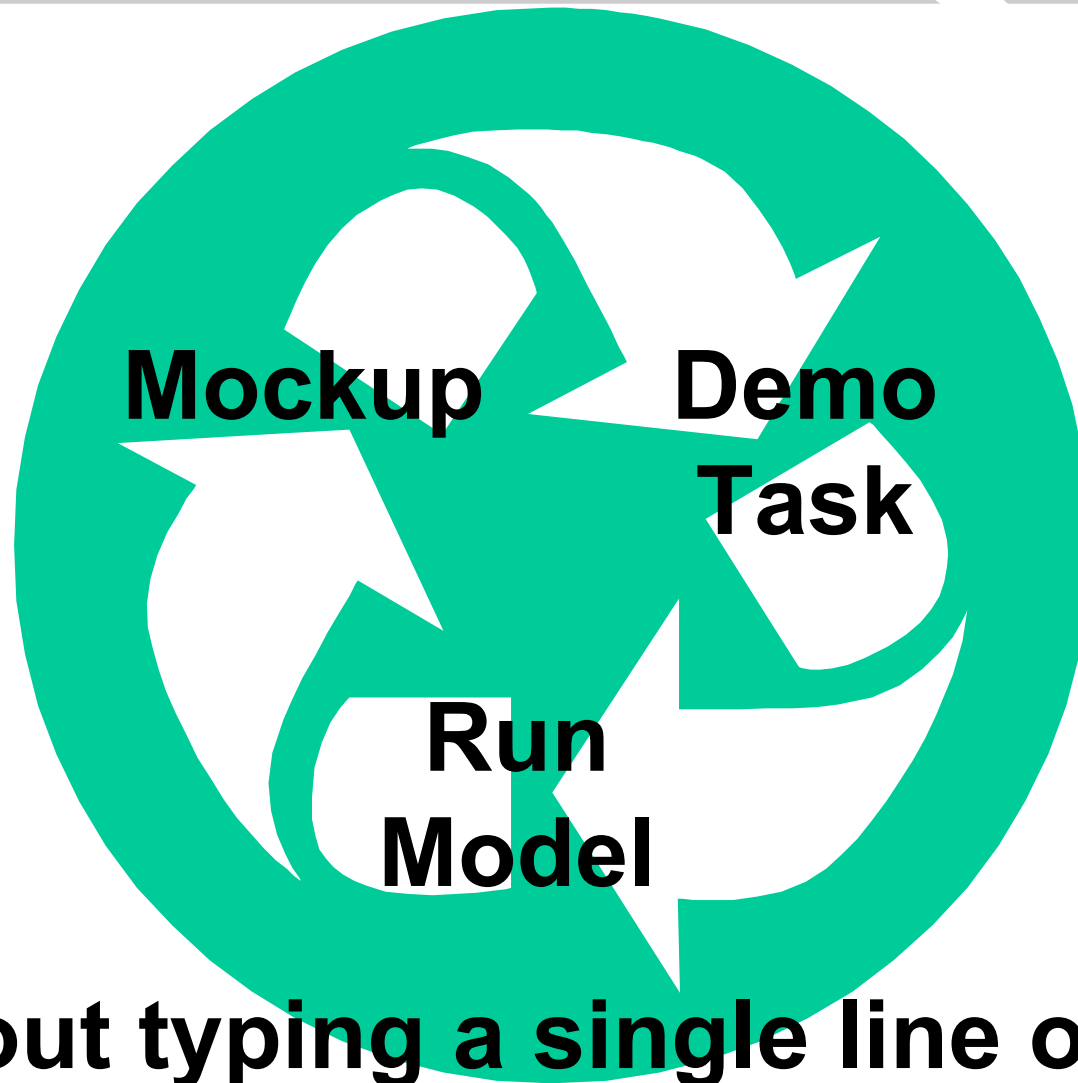
Loc	Att	Kind	Value	Color	ID
(16 16)	NIL	WEB-IMAGE	""	BLACK	WIDGET79
(16 704)	NIL	WEB-BUTTON	"Help"	BLACK	WIDGET80
(41 929)	NIL	WEB-BUTTON	"send-button"	BLACK	WIDGET81
(41 952)	T	WEB-BUTTON	"1-button"	BLACK	WIDGET82
(64 953)	T	WEB-BUTTON	"2-button"	BLACK	WIDGET83
(90 952)	NIL	WEB-BUTTON	"3-button"	BLACK	WIDGET84

Each widget also has a size,
so Fitts’s Law can be calculated for movement times

Thanks to Mike Byrne for helping me get ACT-RPM to tell me these things



Summary: We've Closed the Loop!



Without typing a single line of code



You could open it back up again, if you want a different type of model





What's to come (1)

- Figuring out the relationship between KLM's M operators and ACT-RPM's operations
 - KLM M = 1.35 sec
 - Lot's of ACT-RPM going on in 1.35 sec
 - Look-at
 - Retrieval
 - Verify???
 - Other???
- Automatically placing KLM-like Ms
 - Experience doing this in CRITIQUE (Hudson, John, Knudsen, & Bryne, UIST 99)
 - Use the html widgets to inform the placement



What's to come (2)

- Validating the automatic ACT-Simple models
 - Reproduce published KLM models from the past, e.g., text-editing, CAD, Atropos
- Automatically integrating with Dario Salvucci's driving model
- More widgets, ACT-Simple commands, RPM operators, to simulate other than computer-based devices



What's to come (3)

- Fixing our technical problems
 - Currently exploiting a security hole in Netscape 7.0 for MacOS 9 (ugh)
- Fix our usability problems
 - Widget icons
 - Using too many programs, files, operating systems
 - User testing with UI designers
- Getting it out for others to use



Added bonus

- The same BehaviorRecorder is also a tool for rapid cognitive tutor construction
- So you can tutor on the same tasks you model

...but that's another talk by Ken Koedinger



Want a Demo? See Gus
