ACT-R 6.0 Updates
Summer ‘10 – Summer ‘12

Dan Bothell
Carnegie Mellon University
db30@andrew.cmu.edu
Overview

• 2010 at ICCM described changes up through r891
• Current release version is r1227 (last commit is r1252)

• Lots of minor updates, bug fixes, tweaks, etc
  • Check the commit log for info.

• Discuss the notable changes and additions
  – Mostly software issues related to “using” ACT-R
  – Still nothing that points to a 7.0 in the near future
First some brief data

• 445 subscribers to the mailing list
• Software downloads
  – ACT-R 6.0: 6387
  – ACT-R 5.0: 1252
  – ACT-R 4.0: 175

• User questions/problems/bug reports
  – CMU 83
  – Other 97
Changes

• Instructional materials
• Cognitive components
• New commands
• Environment tools
• AGI
• Other
Instructional Materials

• Added a unit 8 to the tutorial
  – PPM and dynamic pattern matching
• Odd tutorial units have an additional debugging text and model
  – The model doesn’t work
    • Typical issues for the section
  – Text walks through how to find and fix the issues
Cognitive Components

• Temporal
• Motor
• Audio
• Procedural
• Vision
• Declarative
Temporal

• It’s now a default component
Motor

• Mouse movement improved
  – Uses a minimum jerk movement profile
  – :incremental-mouse-moves can take a number as an argument to specify granularity (in seconds)
Audio

• Ability to detect the end of a sound and “current” sounds

• When a sound which has not yet finished is attended the audio module will automatically update the audio-event chunk in the aural-location buffer with the offset and duration when that sound stops
  – Added a duration slot to the audio-event chunk-type for convenience

• Added a new request parameter, :finished, for aural-location buffer and a corresponding query, finished.
  – It works similar to attended
Procedural

- There is now a utility learning trace which shows the changes when a reward occurs: \texttt{ult}
- There is a new ‘type’ of buffer for compilation purposes: \texttt{imaginal}
  - Differs from goal since there’s a cost to the action
  - Queries are important, like the perceptual
- Production compilation trace includes more detail as to why it can’t compose two productions
- \texttt{!eval!} And \texttt{!bind!} operations always occur in the production’s order if possible
- Production warnings now read top-down
Warning: No production defined for (INITIALIZE-ADDITION =GOAL> ADD ARG1 =NUM1 ARG2 =NUM2 SUM NIL ==> =GOAL SUM =NUM1 COUNT 0 +RETRIEVAL> ISA COUNT-ORDER FIRST =NUM1). |

Warning: Invalid syntax in =GOAL> condition. |

Warning: No production defined for (INITIALIZE-ADDITION =GOAL> ISA ADD COUNT =NUM ARG2 =NUM2 SUM =ANSWER ==> =GOAL> ISA ADD COUNT NIL). |

Warning: Invalid buffer modification (=GOAL> ISA ADD COUNT NIL). |

--- end of warnings for undefined production INITIALIZE-ADDITION --- |

Warning: No production defined for (TERMINATE-ADDITION =GOAL> ISA ADD COUNT =NUM ARG2 =NUM2 SUM =ANSWER ==> =GOAL> ISA ADD COUNT NIL). |

Warning: Invalid buffer modification (=GOAL> ISA ADD COUNT NIL). |

--- end of warnings for undefined production TERMINATE-ADDITION --- |

Warning: First element to define-chunk-spec isn't the symbol ISA. (ADD ARG1 =NUM1 ARG2 =NUM2 SUM NIL) |

Warning: Invalid syntax in =GOAL> condition. |

Warning: First element to define-chunk-spec isn't the symbol ISA. (INITIALIZE-ADDITION =GOAL> ADD ARG1 =NUM1 ARG2 =NUM2 SUM NIL ==> =GOAL SUM =NUM1 COUNT 0 +RETRIEVAL> ISA COUNT-ORDER FIRST =NUM1). |

Warning: Invalid buffer modification (=GOAL> ISA ADD COUNT NIL). |

Warning: No production defined for (TERMINATE-ADDITION =GOAL> ISA ADD COUNT =NUM ARG2 =NUM2 SUM =ANSWER ==> =GOAL> ISA ADD COUNT NIL). |
Vision

• Add-word-characters command provides an easy way to get the default devices’ text parsing to group other things with words

• The :nearest request parameter can now take the values clockwise and counterclockwise which are computed relative to an arbitrary “center”
  – Specified with the request parameter :center
  – Initialized with the set-visual-center-point function
    • Automatically set for experiment windows to width/2,height/2
Declarative

• :act can now be set to t, medium, or low
• New parameter :sact (save activation trace)
  – Saves the details instead of printing them
  – Can be accessed later with print-activation-trace and print-chunk-activation-trace
• New request parameter :mp-value
  – Temporarily override the value of :mp for this request
  – Provides a way to guarantee a perfect match when needed
New commands

• SSP
  – Like sgp except for “system” parameters – things that don’t belong to a model/module
    • Default environment port and host for example

• With-parameters
  – Temporarily set parameters for a block of code

(with-parameters (:v t :act t)
  (run 10))
ACT-R Environment

• The history tools are now included by default
  – Buffer, production, and retrieval
• The retrieval history tool now includes the whole activation trace for the chunks
• There is an option that allows the environment to work with multiple models simultaneously
• New tools
  – Parameter viewer
  – Production Graph
Production Graph tool

• Provides a graph of the production firing history

• Works essentially the same as the other history and trace tools
  – Open first or set the enable param. :save-p-history
  – Get the display after the model runs

• Results can be saved as an image or as text in DOT language
All Transitions

- First production
- Matched and selected
- Matched but not selected
- Last production
- Unmatched productions
Frequencies

- Link thickness varies with frequency
  - Thicker more often
Utilities

- Graphs based on when rewards given
- Red production provided a reward
- Blue lines are utilities before (top) and after (bottom)
AGI

• Now supports opening multiple windows simultaneously
  – Doesn’t automatically close when opening a new one
• Each model has only one focus ring displayed
• Open-exp-window now takes a symbol or string as the “title”
• All the commands have an optional or keyword parameter for the window to use
  – If there’s only one open window in the current model can be omitted
  – Otherwise it can be referenced by object or title/name
• One non-backward compatible change: add-line-to-exp-window
  – Previously had an optional color now it’s a keyword :color
Other

• Time is now recorded using an integer count of milliseconds internally
• Most of the initial load warnings have been eliminated
• Windows standalone now uses CCL and doesn’t have the “remote listener” interface
• New directory called user-loads
  – Gets processed last
  – Will never have any files from the distribution in it
*act-r-enabled-p*

- No longer defined in the ACT-R sources
- Didn’t really do much at all
  - Determined whether get-time returned ACT-R’s time or get-internal-real-time
  - Some minor GUI tweaks in the devices checked it
- Almost every model set it
- Tutor models now take an optional parameter to the experiment function to indicate person or model
- Get-time now takes an optional parameter which if provided as nil uses get-internal-real-time otherwise it returns mp-time
  - Doesn’t change for running a model, but not backward compatible for running people
R.I.P. MCL?

• Is anybody still using it
• The multiple window update was not added to the MCL device
• Plan to move it to extras and stop updating it