ACT-R 6.0 Software Updates Summer ‘09 – Summer ‘10

Dan Bothell

Carnegie Mellon University

db30@andrew.cmu.edu

Overview

- Last summer’s release was r765 and this summer’s release is r875 (currently at r891 in archive)

- Lots of minor updates, bug fixes, tweaks, etc
    - Check the commit log and file headers for details

- Discuss the notable changes and additions

Updates for external reasons

- Changes necessary for newer versions of Lisps
  - ACL 8.2/9
  - LispWorks 6
  - RMCL 5.2.1

- Built new Mac environment application
  - Universal binaries

New hooks

:utility-offsets & :activation-offsets

- Allow for multiple functions to be added
- The normal calculations are done first
- All offsets on the list called
- All numeric return values are added to the calculation

New module in extras

- Lebiere & Best short-term inhibition of activation

$$B_i = \log\sum_{j=1}^{n} t_j^{-d} \boxed{- \log\left(1+\left(\frac{t_n}{t_s}\right)^{-d_s}\right)}$$

- Discussed at last year's Workshop at CogSci
- Paper, presentation, and example model also included
  - extras/base-level-inhibition

Procedural Partial Matching

- Buffer slot tests can mismatch and still allow a production to be selected
- Equality tests of slot values can match to similar contents in the buffer chunk’s slot
  - Uses the same similarity values as DM
  - Must have a similarity greater than the maximum difference
- It is enabled by setting the :ppm parameter to a number (defaults to nil)

PPM cont.

- If there is a mismatch the utility of the production for conflict resolution is adjusted

$$Utility_i + \ ppm \times \sum_m similarity(m, v)$$

- Alternatively there's a hook function which can be set with :ppm-hook that allows for a custom utility adjustment when mismatched

Whynot and ppm

- For debugging whynot indicates a partial match and the details

```
Production STATE1 partially matches the current state:
(P STATE1
   =GOAL>
       ISA GOAL
       STATE [STATE1, STATEA, -0.1]
 ==>
   =GOAL>
       STATE STATEB
)
```

New threaded cognition module

- Rewrote threaded cognition as a new goal module
  - No changes to any other module’s code
  - using a new type of buffer
- Performs the same as the original in test models
- Both currently available in extras/threads
  - New-threads and legacy-threads

Why a new type of buffer

- Threaded cognition couldn’t be implemented cleanly before
  - Required changes to buffer and procedural code
  - Difficult to maintain
  - Potential issues with other extensions
- Designed a general mechanism sufficient to implement it
- Should be useful in other contexts
- Detailed in reference manual

Multi-buffer

- A multi-buffer allows the module to maintain a set of chunks (the buffer set) which may be placed into the buffer without copying
- Module must maintain ownership of those chunks
- A chunk cleared from the buffer is removed from the buffer set
  - Use of overwrite-buffer-chunk and the new erase-buffer command recommended
- Still only one chunk “in the buffer” at a time

Searchable buffer

- A multi-buffer for which the procedural module will attempt to find a matching chunk from the buffer set during production matching
  - A multi-buffer does not have to be searchable
- Several constraints on the searching to avoid issues
  - Multiple instantiations
  - Size/complexity of the search

Search buffer match for a production

- Each search buffer may only be used once in a production
- Search takes place after non-search bindings
- All search buffers searched in “parallel”
- Stop at first match found for each buffer
  - Order can be specified by the module
- Any tests among search buffer bindings performed
  - No backtracking – any inconsistency means the production fails to match

Search buffers and conflict resolution

- After conflict set determined the production with highest utility selected
- Module may provide a preference given the chunks which were found
  - Specify offsets to the utility based on matched chunk
- As part of production selection all necessary found chunks are put into their buffers
  - Search buffers which were not required for the selected production are unchanged

New tool in extras

- Save the current declarative and procedural components in a model file
- Added in r878
  - So only available via subversion right now
  - extras/save-model
- The command is save-chunks-and-productions
  - One required parameter – file name
  - Optional parameter whether or not to adjust chunk parameters to time zero (defaults to t)

```
(save-chunks-and-productions "saved-model")
```

What gets saved

- General parameters related to declarative and procedural
  - :esc, :er, :md, :rt, :le, :ms, :mp, :pas, :mas, :ans, :blc, :lf, :bll, :ol, :iu, :ul, :alpha, :ut, :nu, :egs, :epl, :tt, :dat, :ppm
- Chunk-types
- Declarative memory chunks
- Appropriate chunk parameters
  - :creation-time, :reference-count, :reference-list, :similarities
- Productions (both original and compiled)
  - Marks the compiled productions so utility learning treats them correctly
- Production parameters as appropriate
  - :at, :u, :reward
- Model name, current date, and the current :seed parameter saved in comments

What isn’t saved

- Module states
- Buffer contents
- General parameters not on the saved list
- Chunks not in declarative memory
- Events on the queue
- Lisp code
- Chunk Sji settings
  - Recomputed at load time by add-dm
  - Specific add-sji settings are not recorded
- Chunk parameters not on the saved list
- Production parameters not on the saved list

```
;;; Saved version of model PAIRED at run time 1600.0 on 2010/7/2 15:23:16

(clear-all)

(define-model PAIRED-saved

(sgp
:RT -1.7
:ANS 0.5
:LF 0.4
:BLL 0.5
:ESC T
:IU 10
:UL T
:EGS 0.1
:EPL T
)

;;; (sgp :seed (25403176208 13448))
```

```
(chunk-type TASK
  STATE
  STEP
)

...

(add-dm
(
GOAL-0
  ISA TASK
   STATE  NEW-TRIAL
   STEP  RETRIEVING-OPERATOR

)

...
)

(sdp GOAL-0
  :CREATION-TIME -1594.63
  :REFERENCE-COUNT 160
)

...
```

```
(P PRODUCTION0
  "RETRIEVE-OPERATOR & READ-ARG1 - OP1"
   =GOAL>
       ISA TASK
       STATE START
       STEP READY
   =VISUAL-LOCATION>
       ISA VISUAL-LOCATION
   ?VISUAL>
       STATE FREE
 ==>
   =GOAL>
       STATE STIMULUS-READ
       STEP ARG1
   +VISUAL>
       ISA MOVE-ATTENTION
       SCREEN-POS =VISUAL-LOCATION
   +IMAGINAL>
       ISA ARGS
       ARG1 FILL
)

...
(spp RETRIEVE-OPERATOR :u 14.090138 :at 0.05)

...

(setf (production-user-created 'PRODUCTION0) nil)
...
)
```