# ACT-R 6.0 Software Updates Summer ‘08 – Summer ‘09

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

# Overview

- Updates and changes over the past year
  - Documentation
  - Extras
  - Environment
  - New functionality
  - Performance
  - Miscellaneous
- Recommend that you update if using the winter ‘08 release [r723]

# Documentation

- Manual Updated
  - More sections related to adding modules
- Reference manual for the Environment added
  - Covers all the existing tools & the new ones
- New unit 5
  - Siegler model is now an example
  - Assignment is modeling learning in a game
    - 1-hit blackjack

# New Extras

- Blending Module
  - Christian's blended retrieval mechanism
  - Requires normal Declarative memory module
  - Works in parallel with it – independent state
- Threaded Cognition
  - Dario and Niels' theory of concurrent multitasking
  - Extends the goal module to hold a set of goals
  - Modifies procedural module to match against that set of goals

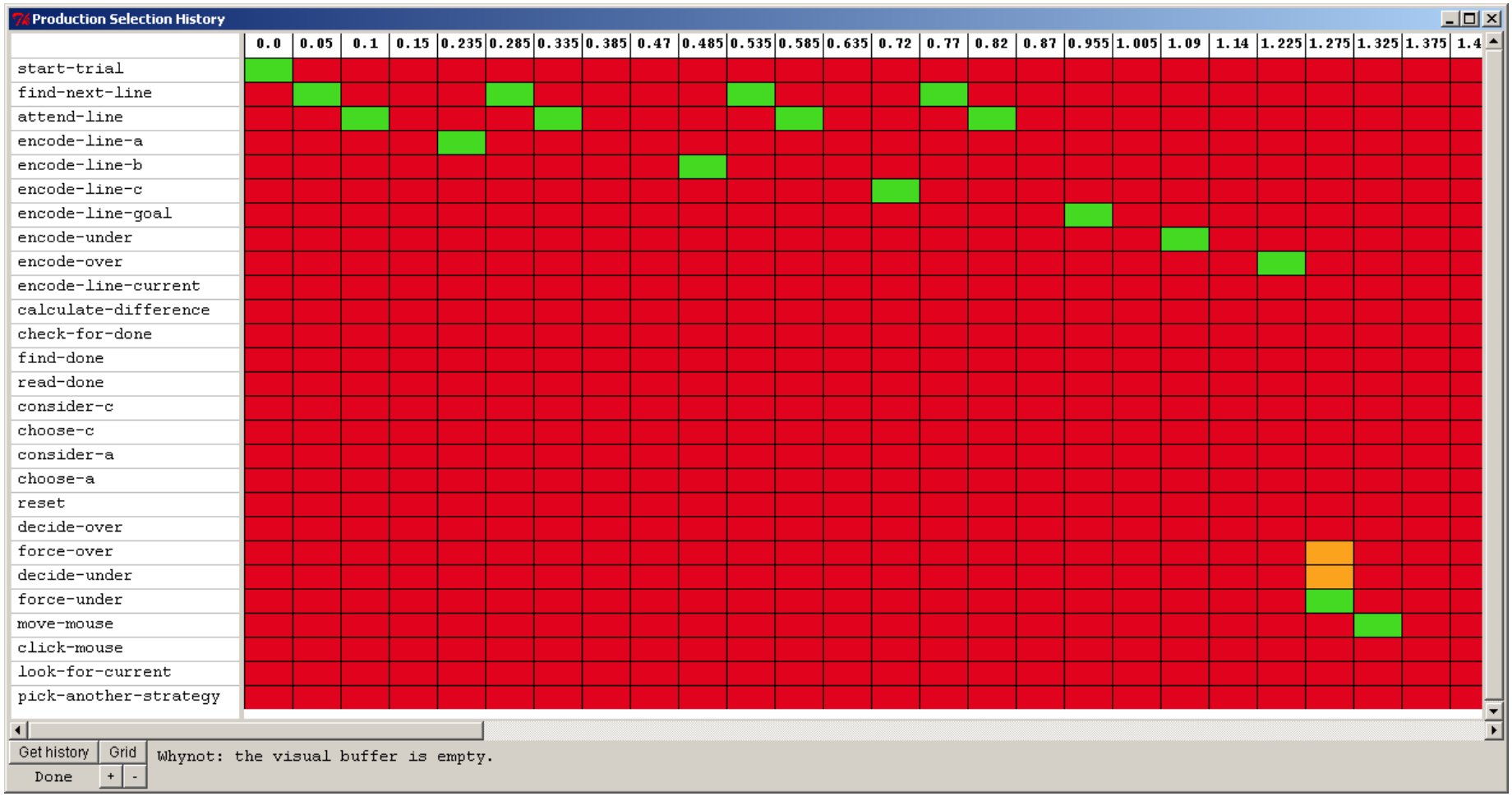
# Environment

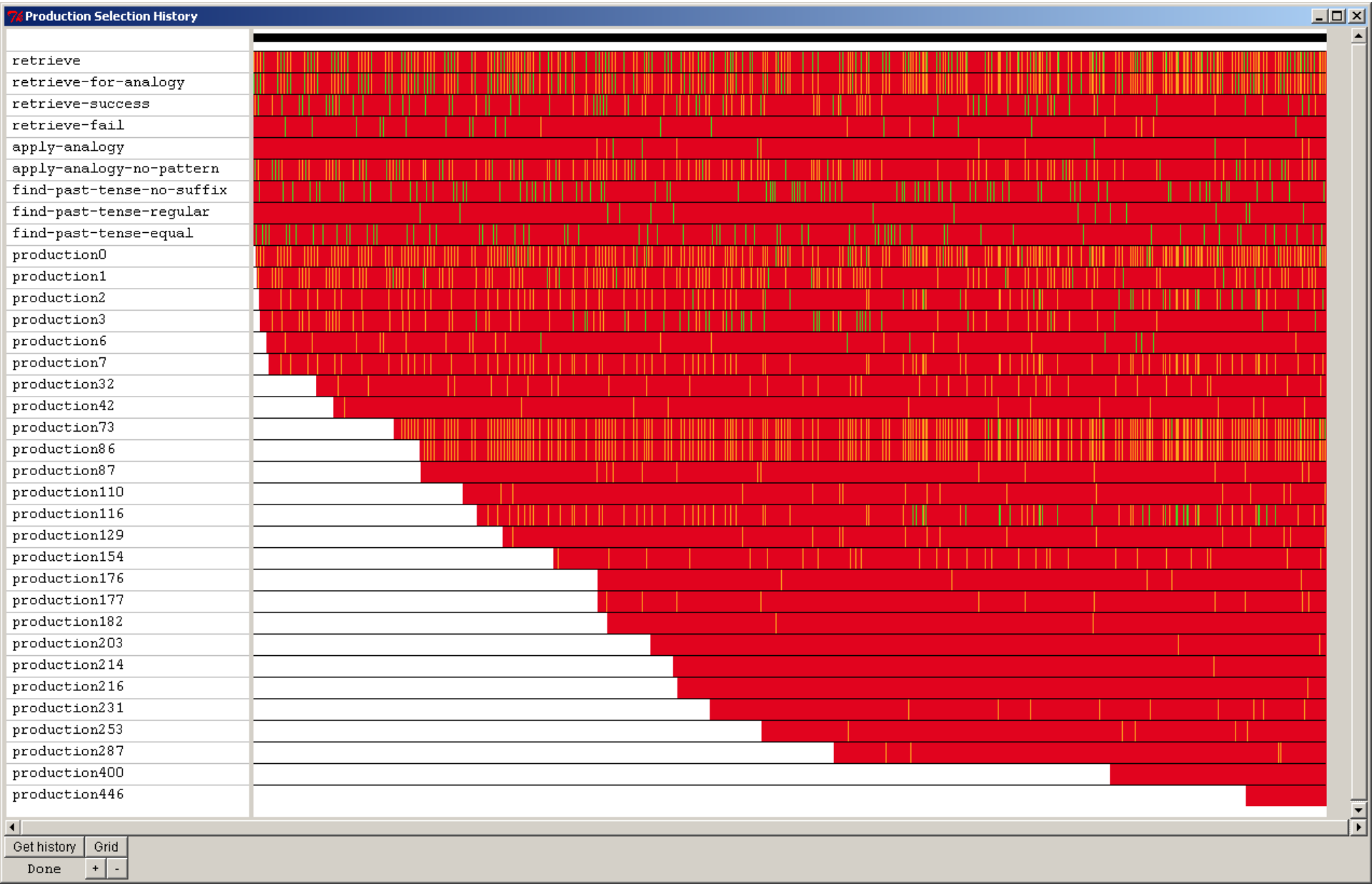
- New command – run-environment
  - Use instead of start-environment
  - Spawns the external app. and makes the connection automatically
  - Works in LispWorks and ACL under Mac OS X and Windows
- Added feature in the graphic traces
  - Clicking on a retrieved chunk or a production name opens the appropriate viewer for that item

# Environment (cont.)

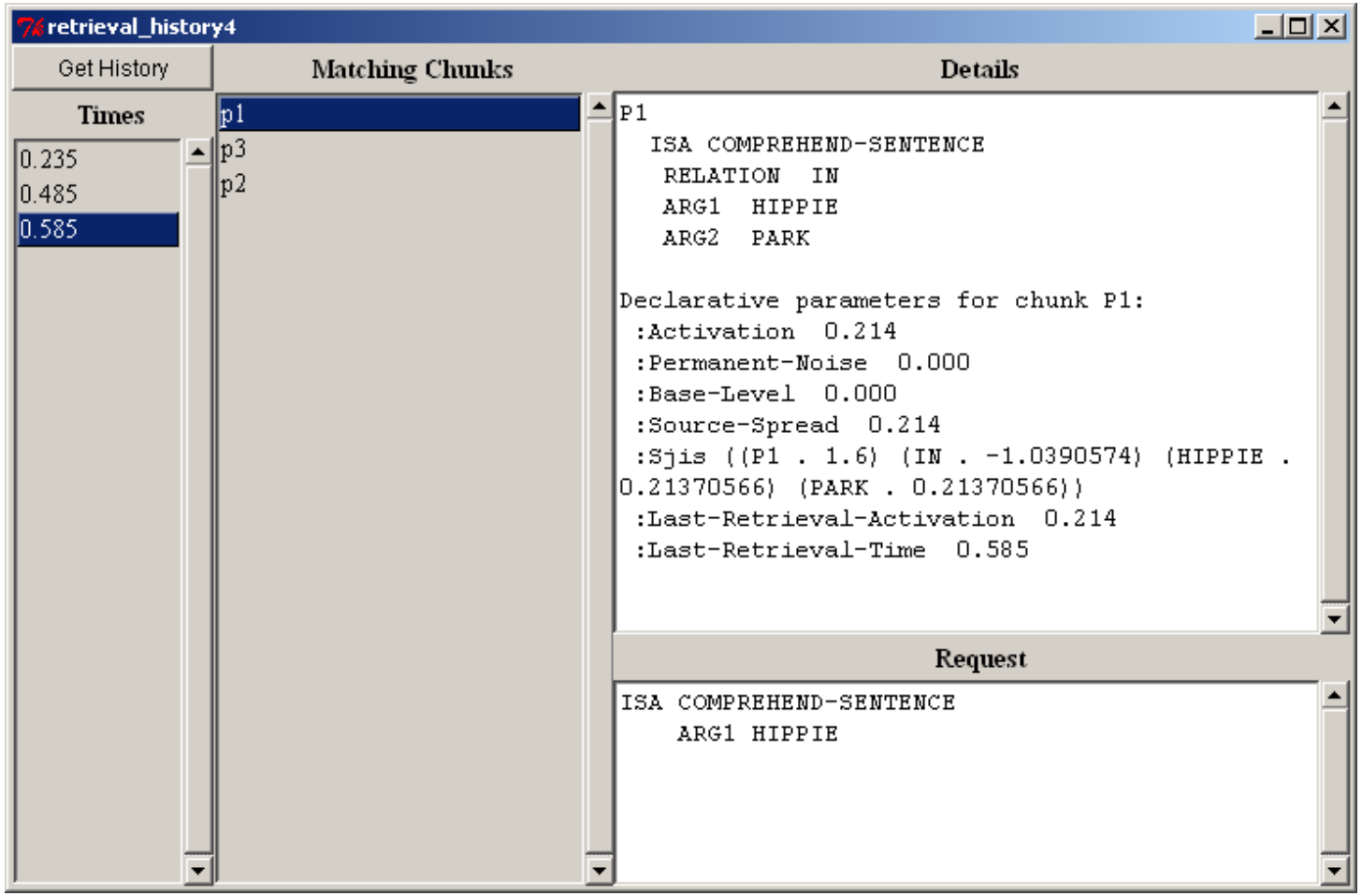
- 3 new tools to display history of events
  - Production selections
  - Retrieval requests
  - Buffer changes
- Not installed by default
  - In extras/history

# Production history

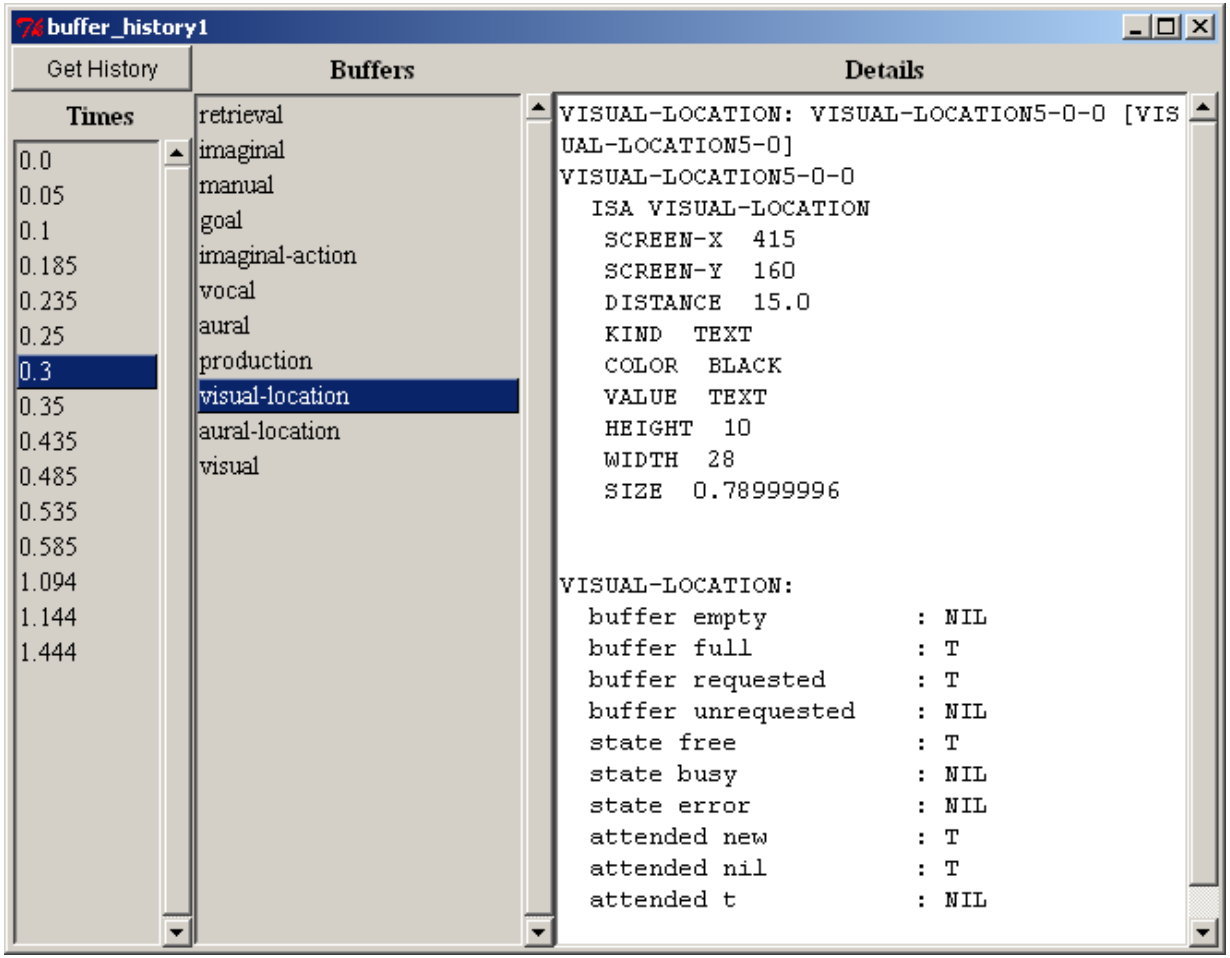



Production Selection History
retrieve
retrieve-for-analogy
retrieve-success
retrieve-fail
apply-analogy
apply-analogy-no-pattern
find-past-tense-no-suffix
find-past-tense-regular
find-past-tense-equal
production0
production1
production2
production3
production6
production7
production32
production42
production73
production86
production87
production110
production116
production129
production154
production176
production177
production182
production203
production214
production216
production231
production253
production287
production400
production446
Get history
Grid
Done
+
-

# Retrieval history



# Buffer history



# New for Declarative module

- New parameter :w-hook
  - Allows one to adjust the $W_{kj}$ values in the spreading activation equation
  - Set to a function like other hooks
    - Passed two parameters
      - buffer name, k
      - slot name, j
  - If it returns a number that overrides the default $W_{kj}$ value

# New for Vision module

- New query for visual buffer
  - Scene-change
- Alternate way for detecting screen changes
  - Not based on theory at this point
    - Modeling convenience
  - More reliable than visual-location buffer stuffing
  - Has a settable change threshold
    - :scene-change-threshold
    - Default is .25

# Scene-change

- The query

```
?visual>
  scene-change t
```

- Will be true when all of these are true
  - there has been a proc-display within :visual-onset-span
  - The change in the visicon was >= to  the threshold
  - The notice has not been explicitly cleared

# Scene-change (cont)

- Change is defined as:

$$Change = \frac{d + n}{o + n}$$

***o***: The number of features in the visicon prior to the update

***d*** : The number of features which have been deleted from the original visicon

***n***: The number of features which are newly added to the visicon by the update

- Can be explicitly cleared with a clear-scene-change request or the existing clear request

```
+visual>
   isa clear-scene-change
```

```
+visual>
   isa clear
```

# “New” chunk name normalizing

- New parameter :dcnn (dynamic chunk name normalizing)
- Works in conjunction with :ncnar
- When both are true (the default values)
  - Chunk names are normalized as the model runs
  - When chunks merge all slots of ALL chunks which have the merged name are updated to the true name
  - Basically how the older versions of ACT-R worked

# More on :dcnn

- Primarily for model debugging
  - Won't see multiple names for one chunk
  - Should not affect the operation of models
- May or may not be faster that normalizing at the end
  - Depends on how much merging occurs, the interrelations among the chunks, and how many chunks the model has
- Does require extra storage to hold the back-links
  - So a larger memory footprint is required to use it
- For best performance :ncnar should still be set to nil
  - Disables all the normalizing

# Simple :dcnn example

```
(add-dm (name isa chunk))

(p start
   ?goal>  buffer empty
  ==>
   +goal> isa goal
   +retrieval> isa chunk)

(p set-up
   =goal> isa goal
   =retrieval> isa chunk
  ==>
   =goal>
    slot =retrieval)

(p report
   =goal>
    isa goal
    slot =val
  ==>
   !output! (the value is =val)
   !stop!)
```

```
CG-USER(12): (sgp :dcnn nil)
(NIL)
CG-USER(13): (run 10)
     0.050   PROCEDURAL   PRODUCTION-FIRED START
     0.050   GOAL         SET-BUFFER-CHUNK GOAL GOAL0
     0.050   DECLARATIVE  SET-BUFFER-CHUNK RETRIEVAL NAME
     0.100   PROCEDURAL   PRODUCTION-FIRED SET-UP
     0.150   PROCEDURAL   PRODUCTION-FIRED REPORT
THE VALUE IS NAME-0
     0.150   ------       BREAK-EVENT Stopped by !stop!

CG-USER(9): (sgp :dcnn t)
(T)
CG-USER(10): (run 10)
     0.050   PROCEDURAL   PRODUCTION-FIRED START
     0.050   GOAL         SET-BUFFER-CHUNK GOAL GOAL0
     0.050   DECLARATIVE  SET-BUFFER-CHUNK RETRIEVAL NAME
     0.100   PROCEDURAL   PRODUCTION-FIRED SET-UP
     0.150   PROCEDURAL   PRODUCTION-FIRED REPORT
THE VALUE IS NAME
     0.150   ------       BREAK-EVENT Stopped by !stop!
```

# Performance Overview

- Added a set of test models to measure performance issues
- Bunch of little internal changes
  - Things users shouldn't notice
- Some more noticeable changes
  - Chunks
  - Vision module
  - Procedural module

# Chunk changes

```
(let* ((ht1 (make-hash-table))
       (s1 (hash-table-size ht1))
       (ht2 (make-hash-table :size s1))
       (s2 (hash-table-size ht2)))
    (= s1 s2))
```

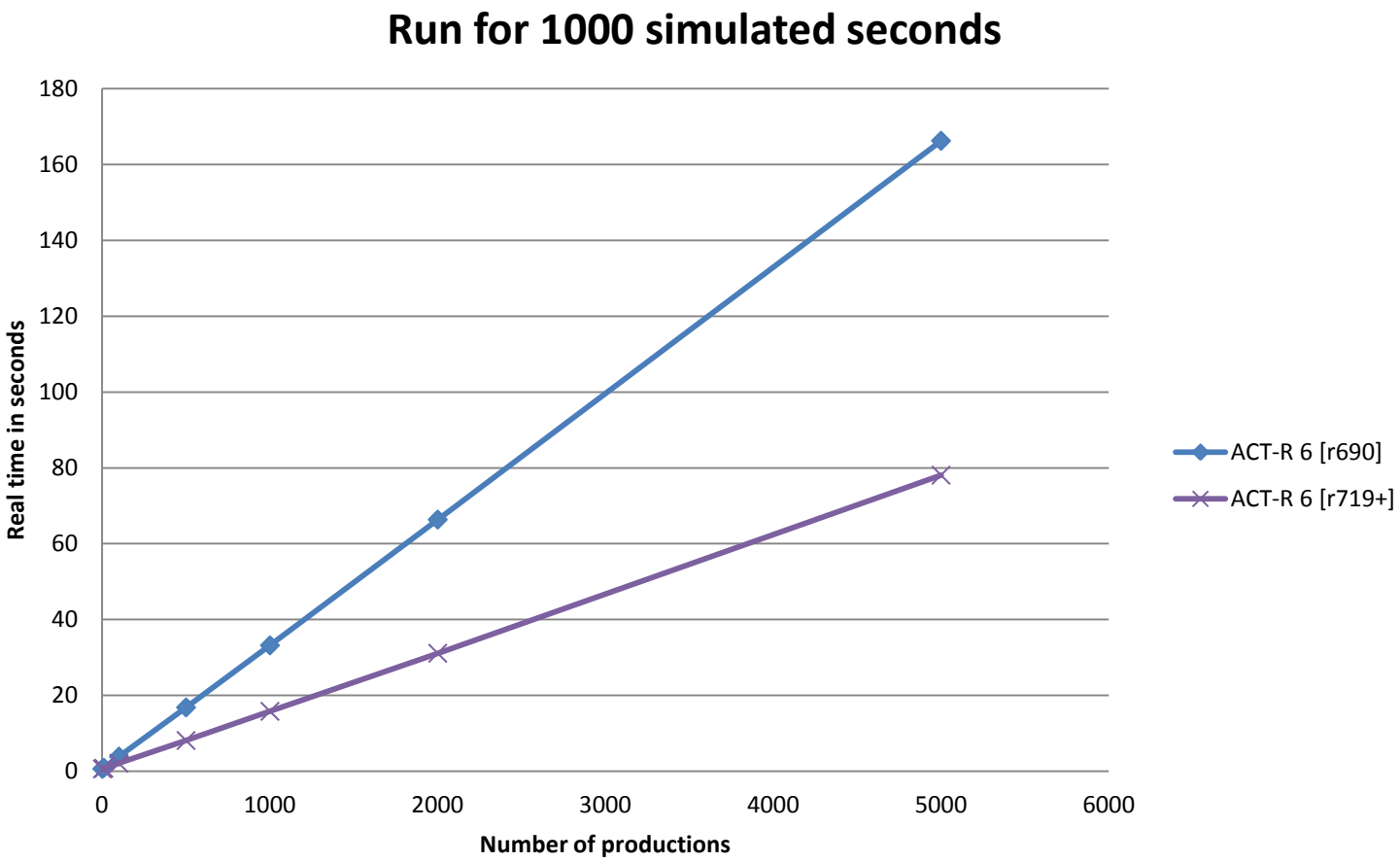
- Fixed how chunks are copied so they don't keep growing in some Lisps
- Changed how declarative module stores fan info
  - Fan-out list is gone now

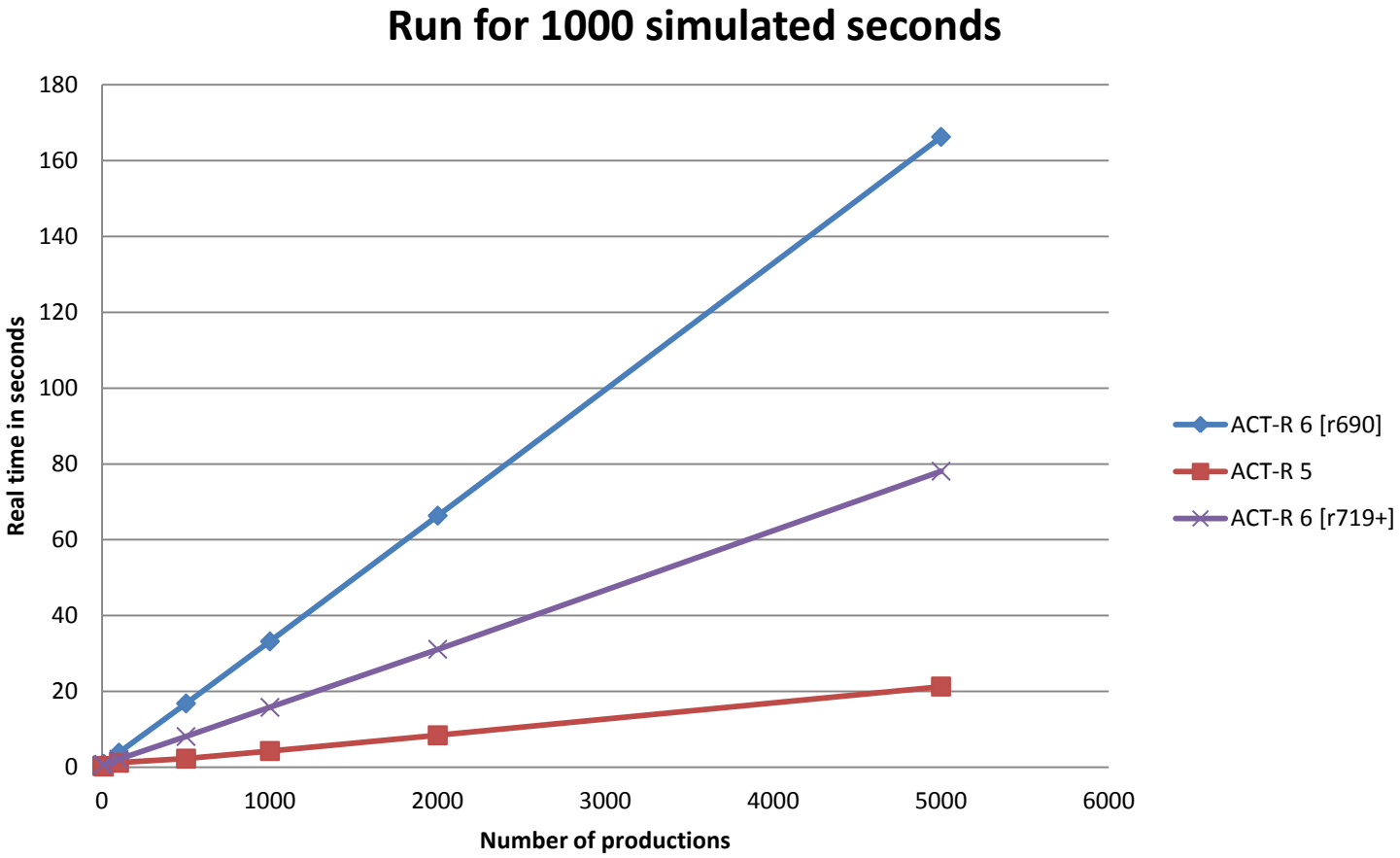
# Vision module & device interface

- Fewer chunks created
  - The virtual devices reuse chunks across proc-display calls
- Deletes chunks when not needed
  - The virtual devices delete their chunks when items are removed from the display
- New parameter :delete-visicon-chunks
  - If true vision module will delete unneeded internal chunks
  - Defaults to t
  - May need to set to nil to work with some extensions (EMMA)

# Procedural module

- Production matching
  - Easy target
  - Sizeable component of most model run times
- Two initial changes
  - Internal production representation
  - Custom buffer matching code
- Tested with a simple model (one of the performance test models)
  - Lots of productions
  - Each tests goal buffer type and single slot
  - Only one matches the chunk in the buffer


Run for 1000 simulated seconds
Real time in seconds
180
160
140
120
100
80
60
40
20
0
0
1000
2000
3000
4000
5000
6000
Number of productions
ACT-R 6 [r690]
ACT-R 6 [r719+]

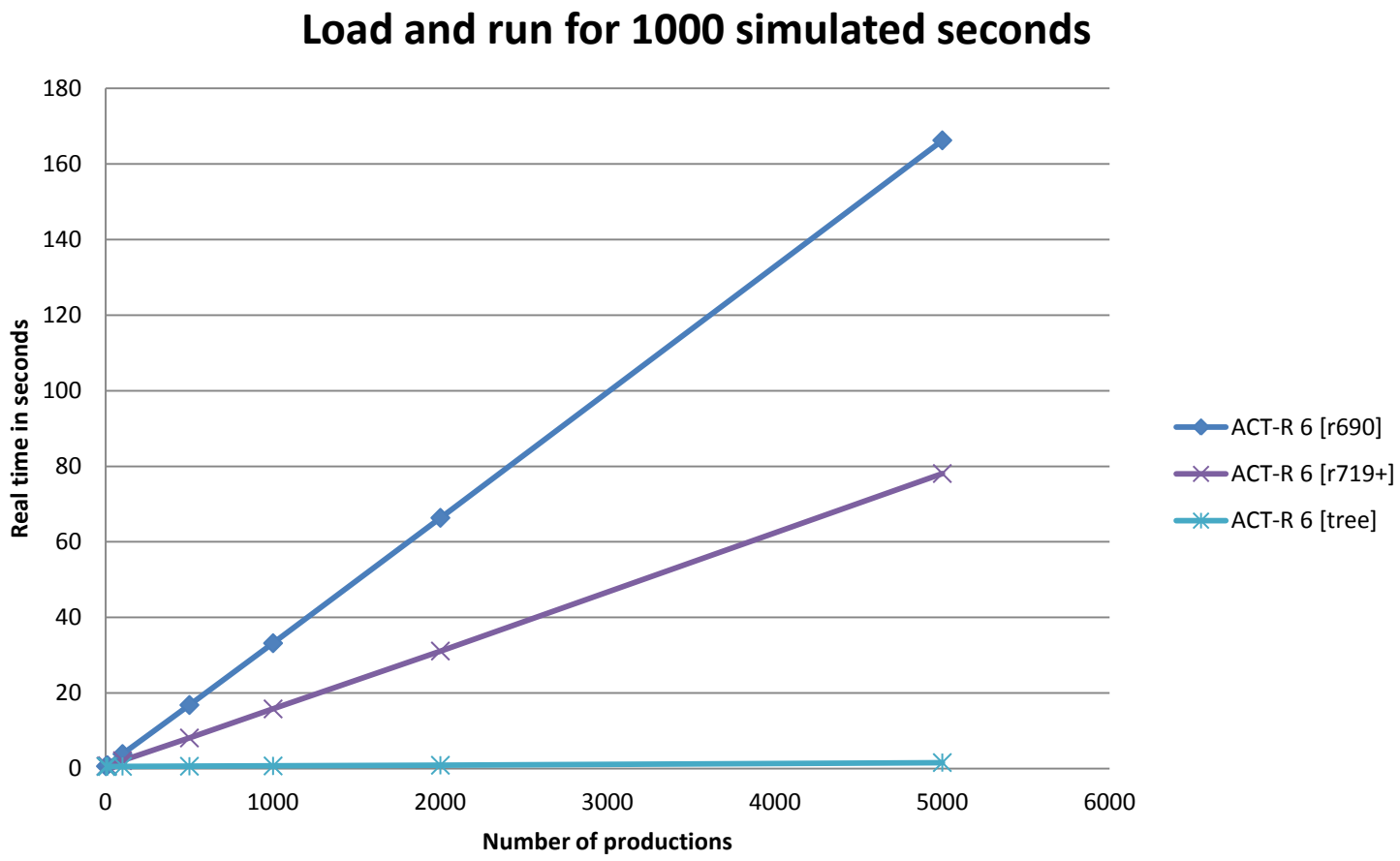

Run for 1000 simulated seconds
Real time in seconds
180
160
140
120
100
80
60
40
20
0
0
1000
2000
3000
4000
5000
6000
Number of productions
ACT-R 6 [r690]
ACT-R 5
ACT-R 6 [r719+]

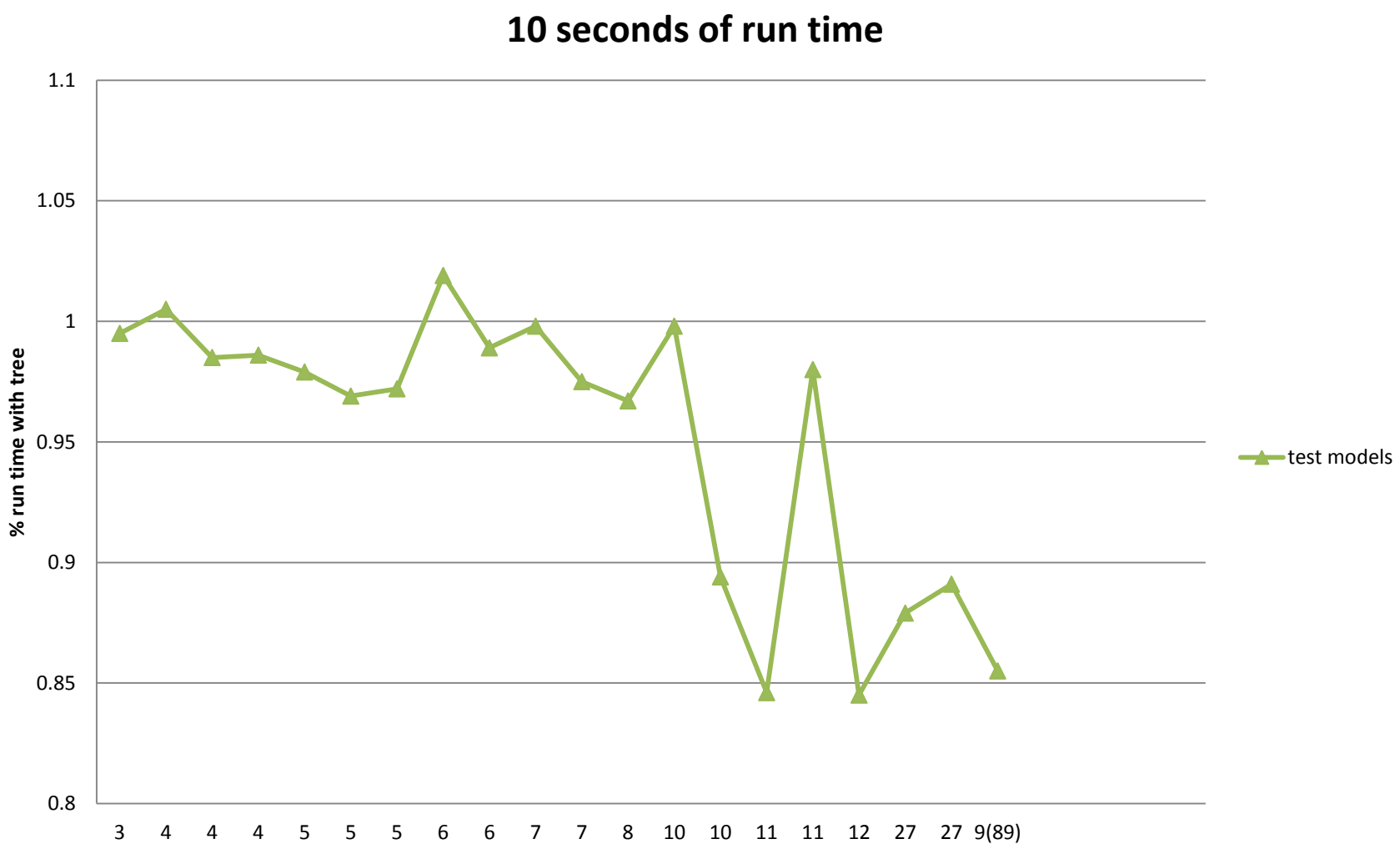
# Can it do better?

- Try a bigger change
  - Algorithmic instead of just improvements
- Why not use RETE?
  - Doesn't really fit our situation
    - No search required in matching
    - We have a fairly small and volatile set of items to match
- Added a simple decision tree

# Decision tree

- Nodes represent the conditions (basic tests)
  - Branches for possible values
- Leaves are a set of productions
  - Which may need further testing
- Use the ID3 algorithm to build it
  - Add condition which has the most information gain
  - Heuristic favors smaller depth trees
  - Add a cut-off if the info. gain is consistently negative
- Happens at load time
  - Does not need to rebuild on a reset


Load and run for 1000 simulated seconds
Real time in seconds
180
160
140
120
100
80
60
40
20
0
0
1000
2000
3000
4000
5000
6000
Number of productions
ACT-R 6 [r690]
ACT-R 6 [r719+]
ACT-R 6 [tree]


10 seconds of run time
% run time with tree
1.1
1.05
1
0.95
0.9
0.85
0.8
3 4 4 4 5 5 5 6 6 7 7 8 10 10 11 11 12 27 27 9(89)
test models

# Notes for the procedural tree

- Not enabled by default
  - need to set the :use-tree parameter to t
- Considerations
  - Time to build the tree
    - Reloading can be more costly
  - Space to hold the tree
    - Trading off space for time savings
- More useful for models with lots of productions
- Works for tutorial and test models
  - Could benefit from more user testing

# Miscellaneous (1)

- Added appropriate tests to work with RMCL
  - Updated MCL that works on Intel Macs through Rosetta

- New parameter :short-copy-names
  - Defaults to nil
  - If set to t then copies of copies don't append a new -0 and just increment #

    ***visual-location3-1*** instead of ***visual-location3-0-0***

# Miscellaneous (2)

- P* doesn't verify slot names in modification requests
  - Allows a module to extend chunk-types "on the fly" if needed in addition to the procedural module's ability to extend them

```
(p* test
   =buffer>
     isa some-type
   ...
 ==>
   ...
   =buffer>
     =slot-name =value)
```

```
(p* test
   =buffer>
     isa some-type
   ...
 ==>
   ...
   +buffer>
     =slot-name =value)
```