ACT-R Software Updates

2020 ACT-R Workshop

Dan Bothell

Carnegie Mellon University

Overview

- Bug Fixes
- Changes that could affect existing models/code
- Performance Improvements
- Additions and changes that don't affect existing models/code
- Pending issues and the Future

Bug fixes

- Production creation
  - Ordering and circularity issues with multiple !mv-bind! conditions or actions
- Production Compilation
  - Eliminates 'duplicate' !safe-eval! and !safe-bind! in composed productions
  - Better tests for matching productions
    - Those that have !safe-eval! and !safe-bind!
    - Ordering issues in original productions
- Motor module tracks both current hand position and the position after the action completes
  - Fixes an issue with subsequent action getting the right features when actions started with preparation free
- Setting both :v and :cmdt to the same filename works as expected now

Changes (procedural module)

- The :lhst actions are only scheduled if both :lhst and :v are true

- Production compilation now allows !safe-bind! on the LHS

- Internals of the compilation-type interface changed
  - If you created a custom type it'll need to be updated

Changes (motor)

- Press-key action now checks that finger is in home row
  - Prints a warning, but still performs potentially wrong action

- All the move/point hand actions just warn when hand at target
  - Previously some would go through all the stages for a no-op

- New requests created with extend-manual-requests only available to models created after the call
  - Previously a reset would be sufficient

Changes (vision)

- Line features now have the orientation in the value slot
  - Both the location and object chunks

Changes (chunk commands)

- Extend-chunks added a :merge-values-fn option
  - Passed the current values for the chunks instead of the names

- Overwrite-buffer-chunk requested parameter is now optional instead of keyword to be consistent with set-buffer-chunk

Changes (external interface)

- The RPC add action has an additional optional parameter
  - If the command adds a Lisp-side macro now that macro can have the parameters use the extended string encoding or not before sending them

- The AGI no longer sends a "clear" message to any visible virtual handlers

Changes (other)

- Scheduling commands can now be given nil as the action
  - Useful when only the details need to be displayed in the trace
  - The "dummy" actions of show-buffer-copy, temporal-clear, no-visual-object-found, and find-loc-failure no longer needed

- In the full version, Lisp function objects (lambda and #'<name>) are no longer valid where it may be externally accessible
  - Scheduled actions, hook functions, module interface functions, etc
  - Still allowed in the Lisp-only (single-threaded) mode

Change (motor actions)

- The trace output for preparation-complete, initiation-complete, and finish-movement now include the time of the request

```
0.485    MOTOR           PRESS-KEY KEY V
0.735    MOTOR           PREPARATION-COMPLETE 0.485
0.785    MOTOR           INITIATION-COMPLETE 0.485
0.885    KEYBOARD        output-key DEMO2 v
1.035    MOTOR           FINISH-MOVEMENT 0.485
```

Changes (tutorial)

- All the unit models use a consistent representation of chunks for numbers and counting facts

```
(chunk-type number number next aural-rep visual-rep vocal-rep)

(one isa number number one next two aural-rep 1 visual-rep "1" vocal-rep "one")
(two isa number number two next three aural-rep 2 visual-rep "2" vocal-rep "two")
```
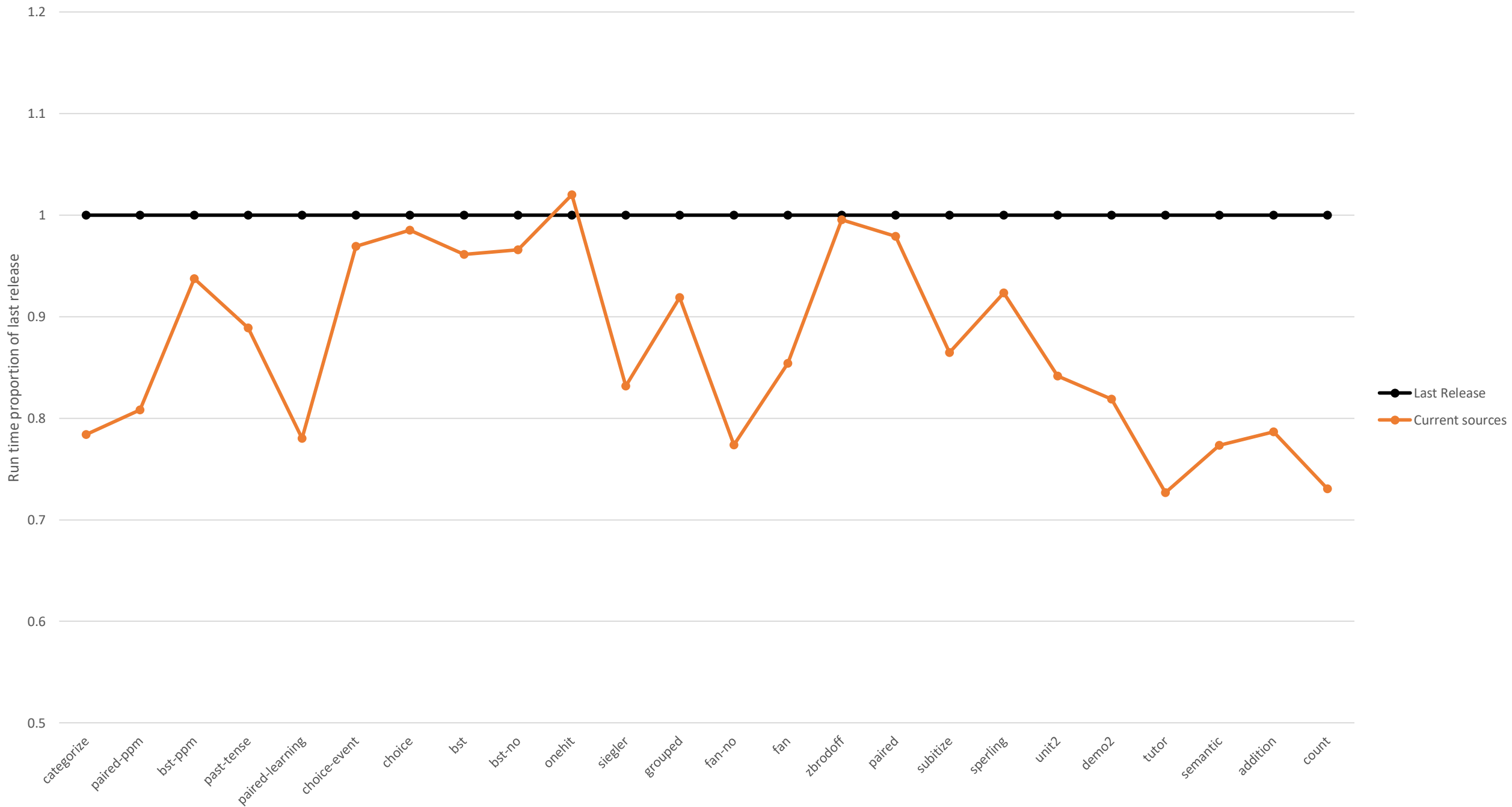
Performance

- Modules can create chunk-types and chunks at creation time instead of reset time
  - All the provided modules now do so
- Determination of whether an event's trace should be displayed simplified
- Procedural module stores productions by buffer usage for matching
  - Only test the appropriate group
- Production compilation improvements
  - Finding if a matching production exists
    - Tables to search to reduce tests
  - Stores info at production creation time instead of lookup at every composition

Performance

Performance
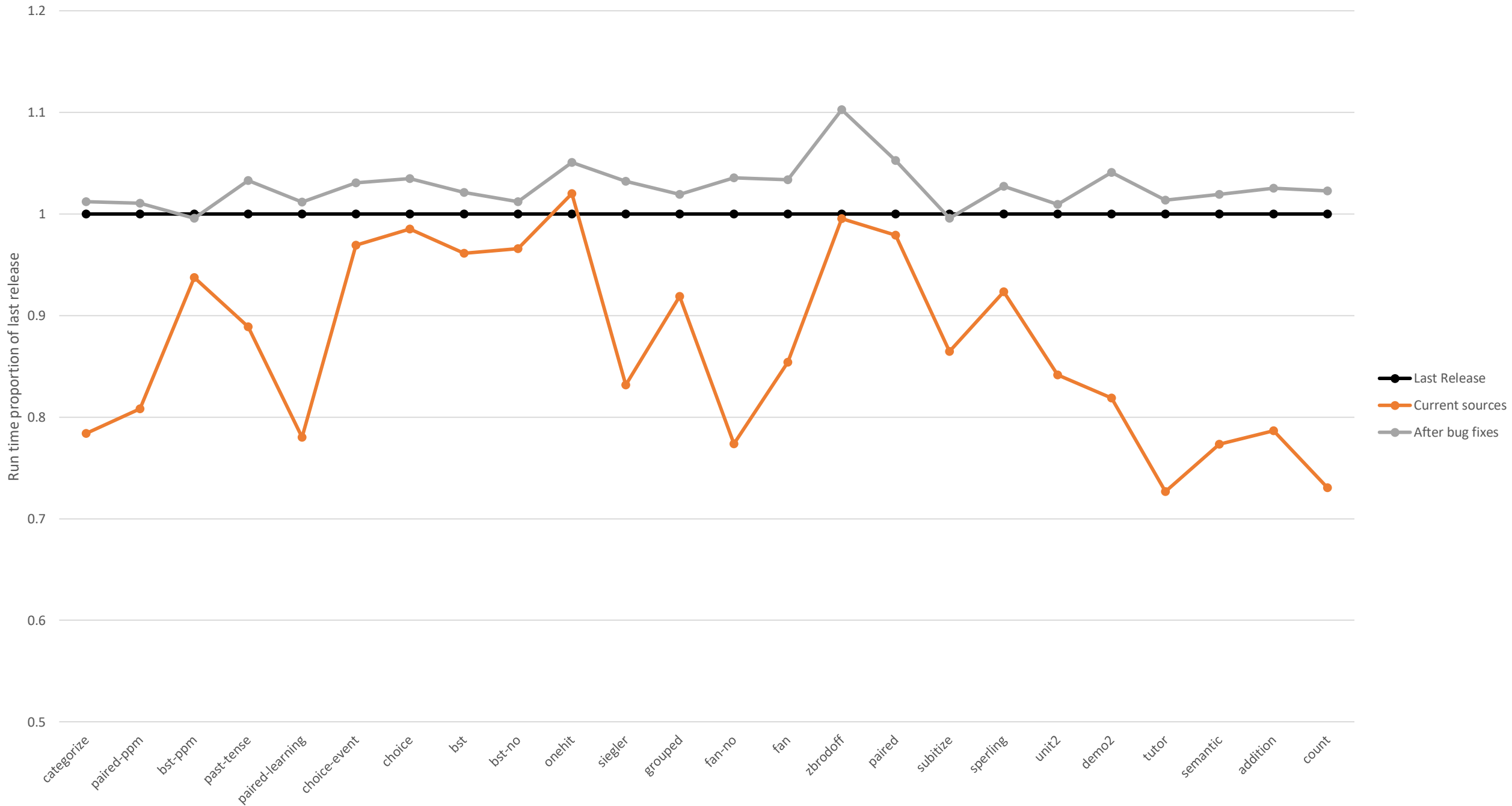
Additions (parameters)

- :add-production-hook
  - Called when productions are defined and passed the production name

- :ignore-buffers
  - Set to a list of buffer names
  - DM doesn't store chunks from those buffers

- Hook parameters that have lists of commands can now have specific items removed instead of only removing everything
  - Pass a list with :remove and the command name

  ```
  (sgp :conflict-set-hook (:remove some-hook-fn))
  ```

Additions (other)

- Most chunk and chunk-type commands now have remote versions

- The file recompile-act-r.lisp
  - Loads things as load-act-r.lisp does, but forces things to be recompiled

- The feature switch :dont-compile-actr
  - Uses existing compiled files only

- examples/creating-modules/external/simple_declarative.py
  - Replacement for declarative module sufficient to run unit 1 models

Issues

- Remote access to chunk parameters during merging
  - The chunks are locked during the merge and when parameters are accessed

- Full ACT-R performance under SBCL on (some?) Macs

  Evaluation took:
    0.457 seconds of real time
    0.003262 seconds of total run time (0.001190 user, 0.002072 system)
    0.66% CPU
    1,645,054,546 processor cycles
    32,768 bytes consed

- PPM and production compilation

```
(define-model ppl-ordering
   (sgp :esc t :ppm 1)
   (chunk-type test slot1 slot2)
   (define-chunks c1 c2)
   (set-similarities (c1 c2 -.9))

   (set-buffer-chunk 'goal
          (car (define-chunks (slot1 c1))))
   (set-buffer-chunk 'imaginal (car (define-chunks (slot2
c2))))
```

```
(p p1                          (p p2
  =goal>                         =imaginal>
    slot1 =a                       slot2 =a
  =imaginal>                       slot2 =c
    slot2 =a                     =goal>
    slot2 =c                       slot1 =a
  ==>                            ==>
    +retrieval>                    +retrieval>
    slot1 =a                       slot1 =a
    slot2 =c)                      slot2 =c))
```

> (whynot)

Production P1 partially matches the current state:
(P P1
    =GOAL>
       SLOT1 C1
    =IMAGINAL>
       SLOT2 [C1, C2, -0.9]
       SLOT2 C2
 ==>
    +RETRIEVAL>
       SLOT1 C1
       SLOT2 C2
)

Production P2 partially matches the current state:
(P P2
    =IMAGINAL>
       SLOT2 C2
       SLOT2 C2
    =GOAL>
       SLOT1 [C2, C1, -0.9]
 ==>
    +RETRIEVAL>
       SLOT1 C2
       SLOT2 C2
)

Future

- Apple…