RICE

ACT-R 5.0 and ACT-R/PM

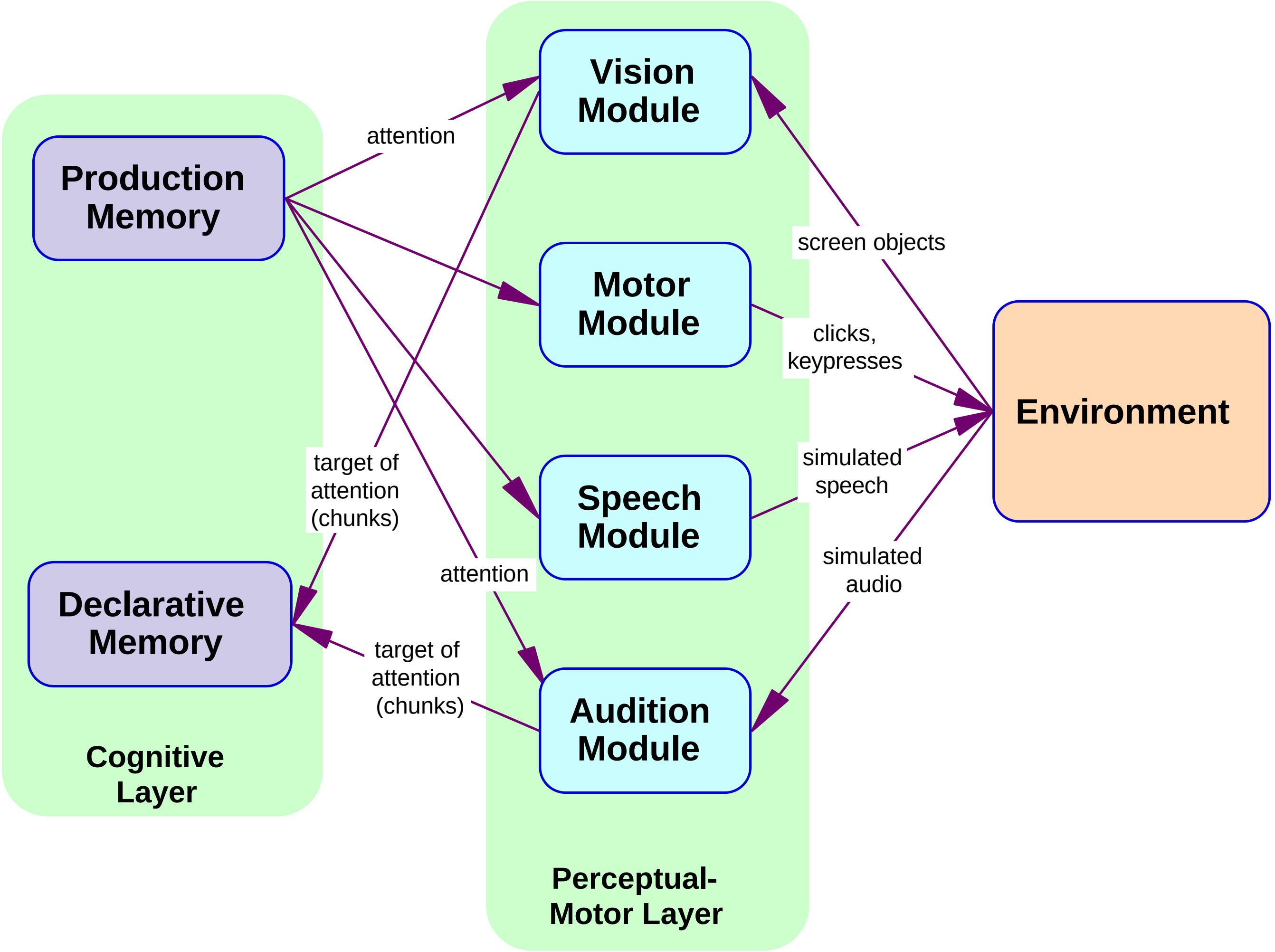
Michael D. Byrne
Department of Psychology
Rice University
Houston, TX 77005
byrne@rice.edu
http://chil.rice.edu

chil
THE COMPUTER-HUMAN
INTERACTION LABORATORY AT RICE UNIVERSITY

Overview

- RPM and 5.0
  - Buffer syntax
  - Cognition parallels
  - Activation sourcing
  - Compatibility issues
- RPM opportunities and future work
  - EMMA
  - Top-down vs. bottom-up attentional control
  - Visual object synthesis
  - Lotsa other stuff

ACT-R/PM (under 4.0)



5.0 Buffer Syntax

- LHS now consists entirely of testing the state of various "buffers"
  - Goal buffer
  - Retrieval buffer
  - PM state buffers (e.g., motor-state)
  - Visual-location and visual (object) buffers
  - Aural-locaiton and aural (object) buffers
- Goodbye to "time now"!
- Elimination of "!send-command!" syntax
  - Use "+" syntax on RHS to send commands

ACT-R 4.0 vs. 5.0

```
(p look-label
 =goal>
  isa                 do-menu
  target     nil
 =loc>
  isa                 visual-location
  time       now
  screen-x   LOWEST
  attended            nil
 =vis>
  isa                 module-state
  module     :vision
  modality            free
 =mot>
  isa                 module-state
  module     :motor
  modality            free
==>
 !send-command! :VISION move-attention :location
   =loc :scale WORD
 !send-command! :MOTOR move-cursor :loc =loc
)
```

```
(p look-label-5
 =goal>
  isa                 do-menu
  target     nil
 =visual-location>
  isa                 visual-location
 =visual-state>
  isa                 module-state
  modality            free
 =motor-state>
  isa                 module-state
  modality            free
==>
 +visual>
  isa                 visual-object
  screen-pos          =visual-location
 +manual>
   isa                move-cursor
   loc                =visual-location
)
```

Ramifications

- Cleaner syntax (yay!)
  - More consistent
  - No way to confuse RPM calls and retreivals
- Issues
  - Restricts motor flexibility
    - Each command is a chunk type, therefore fixed # of arguments
    - The PREPARE command takes a variable number of arguments
  - No parallel to “time now” LHS test on visual-location
    - Under 5.0, can only request an action on a buffer in the RHS
    - LHS is only for tests of a buffer

Two Productions

```
(p look-label-5
   =goal>
     isa                     do-menu
     target    nil
   =visual-location>
     isa                     visual-location
   =visual-state>
     isa                     module-state
     modality                free
   =motor-state>
     isa                     module-state
     modality                free
==>
   +visual>
     isa                     visual-object
     screen-pos              =visual-location
   +manual>
      isa                    move-cursor
      loc                    =visual-location
)
```

```
(p find-label-5
   =goal>
     isa                     do-menu
     target    nil
==>
   +visual-location>
     isa                     visual-location
       screen-x              lowest
       attended              nil
)
```

Visual-location Testing

- Thus, the “find-and-shift” idiom has to be split across two productions
  - This affects timing—old shift time was 185 ms (one 50 ms production, one 135 ms shift)
    - An extra production required at each step
    - Attention shift latency dropped to 50 ms (why not 85?)
  - This affects state control
    - Both of those productions will match, so now we need to be more restrictive with conditions
- The (current) solution: “buffer stuffing”
  - Visual-locations automatically “stuffed” into the =visual-location> buffer
  - Default is newest & furthest left (lowest screen-x)

Buffer-stuffing Issues

- This creates one other problem:
  - Display updates cause an implicit attention shift to the currently-attended location (the “blink” problem)
  - Not consistent with buffer stuffing
- Is the improvement in syntax worth breaking the idiom?
- Discussion: We could make the =visual-location> and =aural-location> buffers “instant” buffers
  - That is, not requiring RHS call-out
  - Breaks parallel syntax (bad)
  - Fixes timing issue and blink issue (good)
  - Improves code-level compatibility with 4.0 (good)
  - Would models be easier or harder to understand?

Cognition-PM Parallels

- 5.0 makes the declarative memory system and the visual/audio systems look very much alike
  - Set up a request for information on the RHS
  - Get it back in a buffer
  - Asynchronous
- But for PM requests, it is possible for a production to check whether a request is in progress
  - For example, by testing the =visual-state>
- So, should there be a =retrieval-state> ?
- Note that it is possible to set up a retrieval and harvest it in two productions, but vision/audio requires three

Activation Sources

- Under 4.0, the slots of the currently attended visual object (and the currently attended sound) were activation sources
- This enabled ACT-R to rapidly answer questions like “what color is the thing you’re looking at?”
  - color slot of object was activation source
  - Thus, it is retrieved very quickly
  - Should properties of attended object be highly accessible?
- This has been removed for 5.0
  - ?

Backward Compatibility Issues

- How many RPM models based on 4.0 will break under 5.0?
  - In principle, very few: “time now” could just be translated to a buffer test
  - However, find-and-shift idiom will have some trouble being translated
- Implementation
  - 5.0 makes a lot of under-the-hood changes that render it not backward-compatible at the code level
  - Maintaining one version of RPM is taxing enough, I don’t know about maintaining two
  - Should all future development of RPM assume ACT-R 5.0?

EMMA

- I keep saying that if people have ideas about extending RPM, by all means bounce it off me and we'll see how it goes
  - This has finally happened!
- Dario Salvucci's Eye Movements and Model of Attention (EMMA) extension to the Vision Module
  - Separates attention shifts and eye movements
  - Now part of the RPM 2.0 release (<u>should</u> work with 5.0 but I'm not sure yet)
  - Dario wrote the original, and Dario and I hammered out a new version
  - Still some unresolved issues

Bottom-up vs. Top-down Control of Attention

- Attentional control in RPM (under 4.0) is entirely top-down
- Buffer stuffing gives some modicum of bottom-up attentional control
- How should this work?
  - Current literature on top-down vs. bottom-up control is mixed
  - Best guess seems to be that top-down settings override bottom-up when present, but there isn’t always top-down
  - Something like the Wolfe model might work, except that isn’t fleshed-out enough to implement
  - I have a grad student working on this

What is the identity of the green LETTER?

7 5

9

2

G 8

T

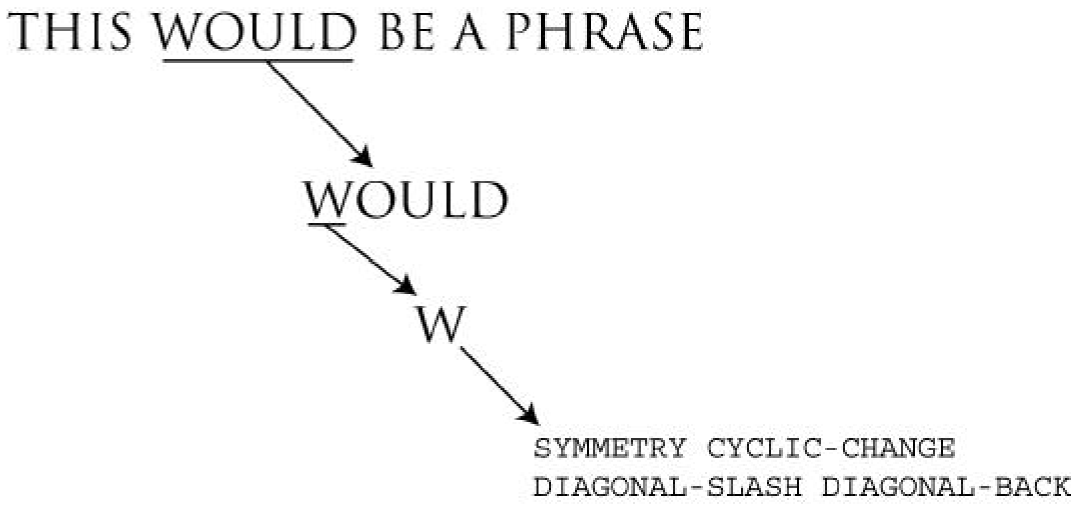
3

4 3

3

6 1

Visual Object Synthesis



THIS WOULD BE A PHRASE

WOULD

W

SYMMETRY CYCLIC-CHANGE
DIAGONAL-SLASH DIAGONAL-BACK

- Scales are all defined for text (phrase, word, letter) but not for other objects
- How should it work more generally?
- Use angle-based scale?

Other Issues We Could Discuss

- Number of finsts
- Targeting nothing still isn’t really worked out
- Visual guidance constraints on aimed movements are not really enforced
  - Should they be?
  - If so, how?
- Movement noise
- Spatial cognition