Towards persistent cognition

Niels Taatgen

Challenge

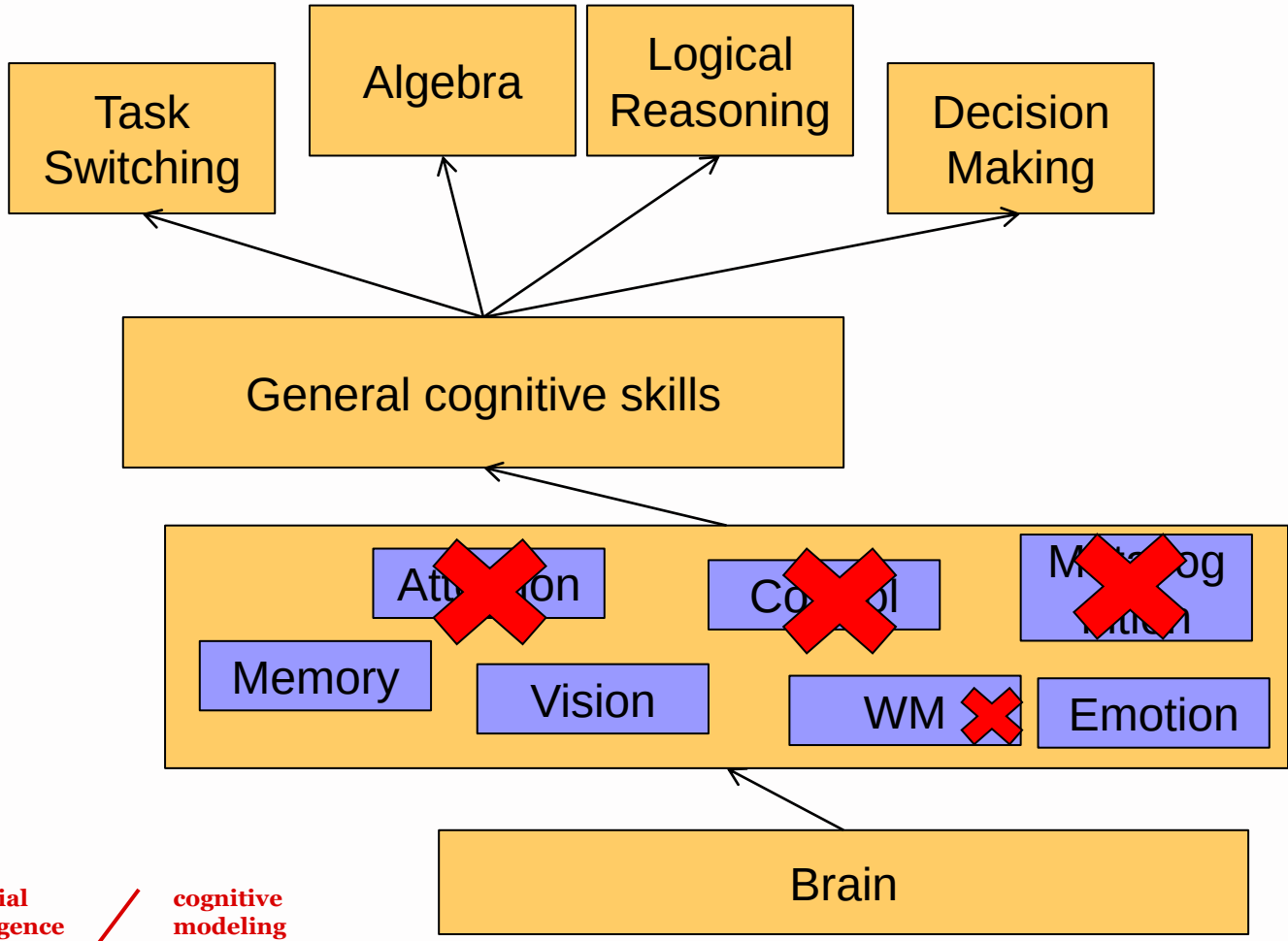
- How to get from models of specific tasks to cognition in general?
  - Larger time scale
  - Goal selection and organization
  - Transfer, general skills
  - Language and semantics

Alternative worldview

Fast learning

Slow learning

No learning
(or very
very slow)



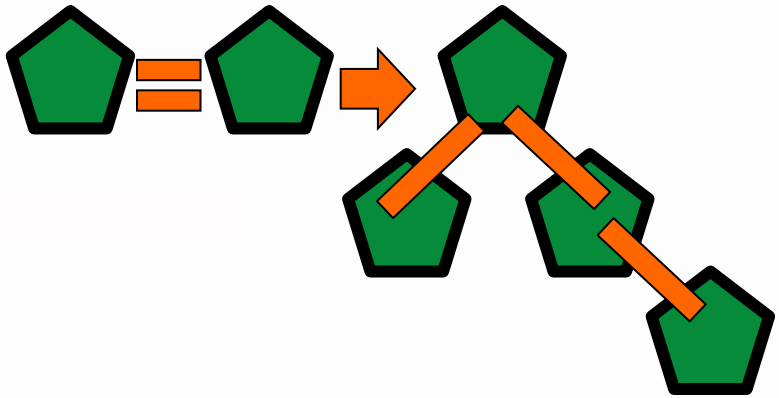
Always return to Newell

Time Units	System	World theory
Hours	Task	
10 min	Task	Rational Band
minutes	Task	
10 sec	Unit Task	
1 sec	Operations	Cognitive Band
100 msec	Deliberate act	
10 msec	Neural circuit	Biological Band

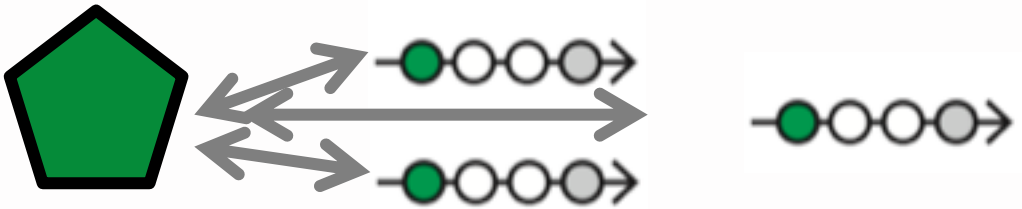
Production Rules

Productions span 4 levels

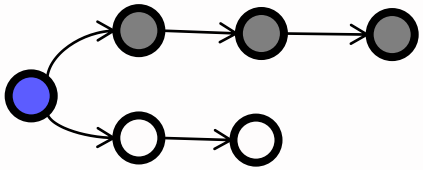
- Good
  - With a single representation, quickly build models of complicated tasks
- Bad
  - Stuck at the level of task
  - Mechanistic: good for 10ms level, bad for the task level
  - Utility cannot handle the dynamics of choice at the task level



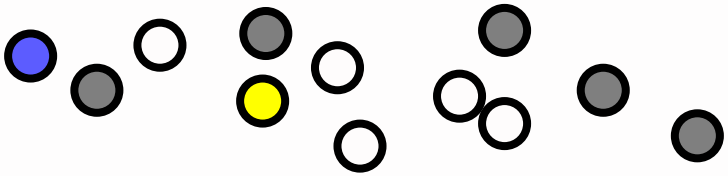
Tasks



Goals



Operators



PRIMs

"Semantic"

"Syntactic"

Example: RITL experiment

Task 1

Instructions

SAME
SWEET
LEFT INDEX

3 s

Trials

Answer: TRUE
(Left index finger)

Grape
Apple

2 s each

Task 1 description:
If the answer to 'is it SWEET?'
is the SAME for both words,
press your LEFT INDEX finger

Task 2

Instructions

SECOND
LOUD
RIGHT MIDDLE

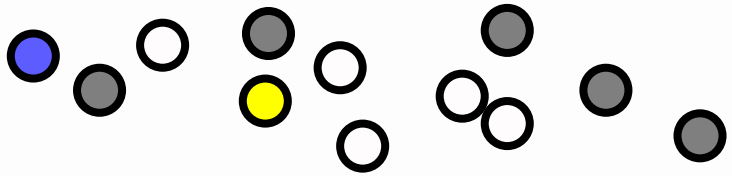
Trials

Answer: TRUE
(Right middle finger)

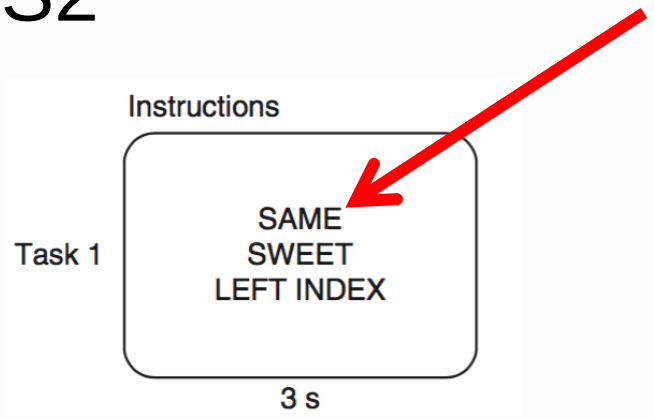
Leaf
Dynamite

Task 2 description:
If the answer to 'is it LOUD?'
is yes for the second word,
press your RIGHT MIDDLE finger

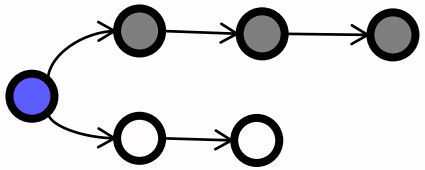
PRIMs



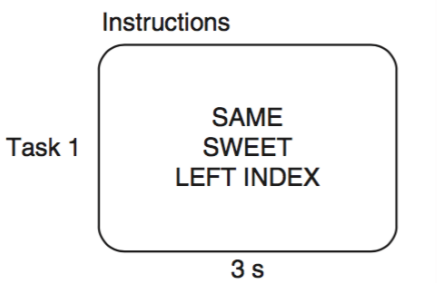
- Sub-procedural
- Do single comparison between two buffer slots, or a single copy operation from one slot to another slot
- Slots in buffers are fixed (slot1, slot2, ...)
- RITL example: PS1=PS2



Operators

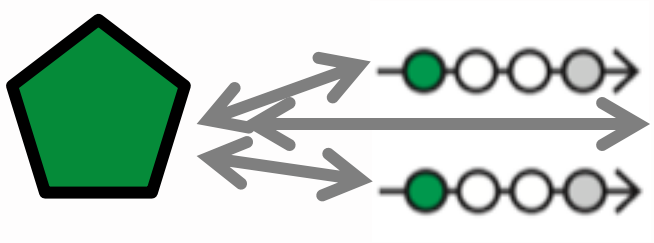


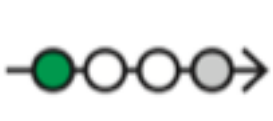
- Combinations of PRIMs (several conditions and actions)
- Operators are declarative structures, but groups of PRIMs are carried out by production rules




```
operator same {
        PS1 = PS2
==>
        affirmative -> PS3
        next-goal -> G2
}
```

Goals

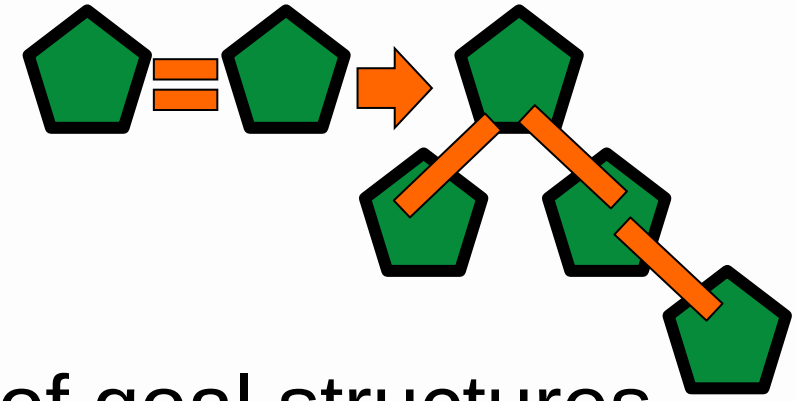




- Goals are associated with operators, so if a goal is currently active, operators associated have a high probability of selection
- Goals have the “traditional” slot-value chunk representation

```
determine-attribute
   isa goaltype
      fact-type sweet
      next-goal same
```

Tasks

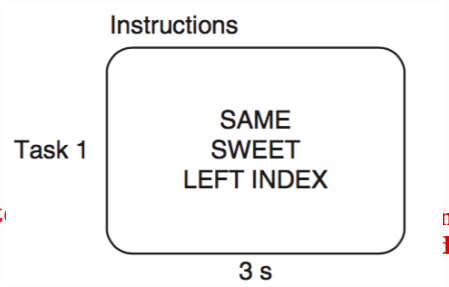


- Tasks consists of goal structures
- Several possible organizations
  - Sequential
  - Hierarchical
  - Parallel
- Link with semantics

```
determine-attribute
   isa goaltype
      fact-type sweet
      next-goal same
```

```
same
   isa goaltype
      next-goal press
```

```
press
   isa goaltype
      finger left-index
```



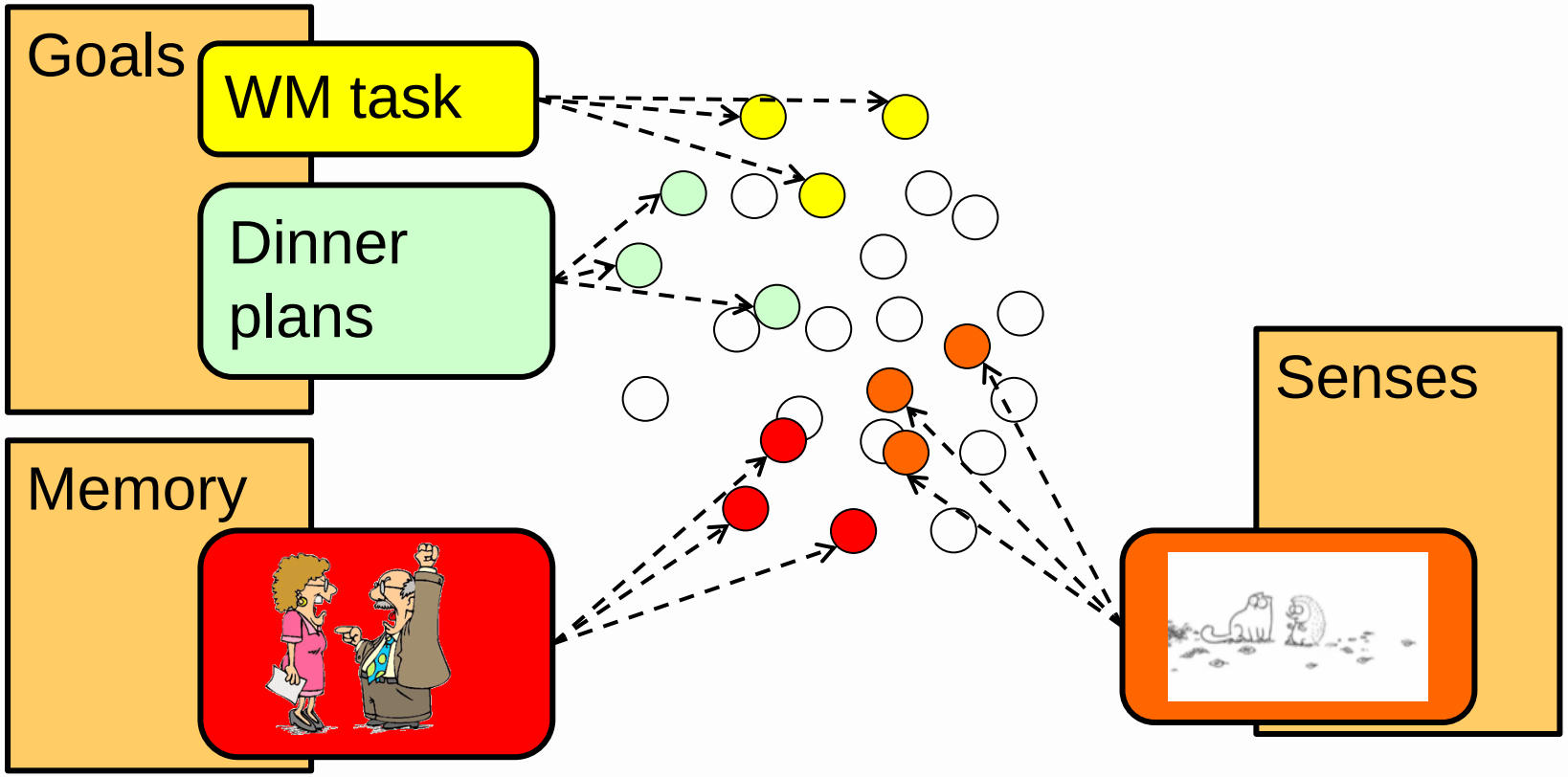
For RITL

- We need one more component: a goal that generates three three goals based on the instruction

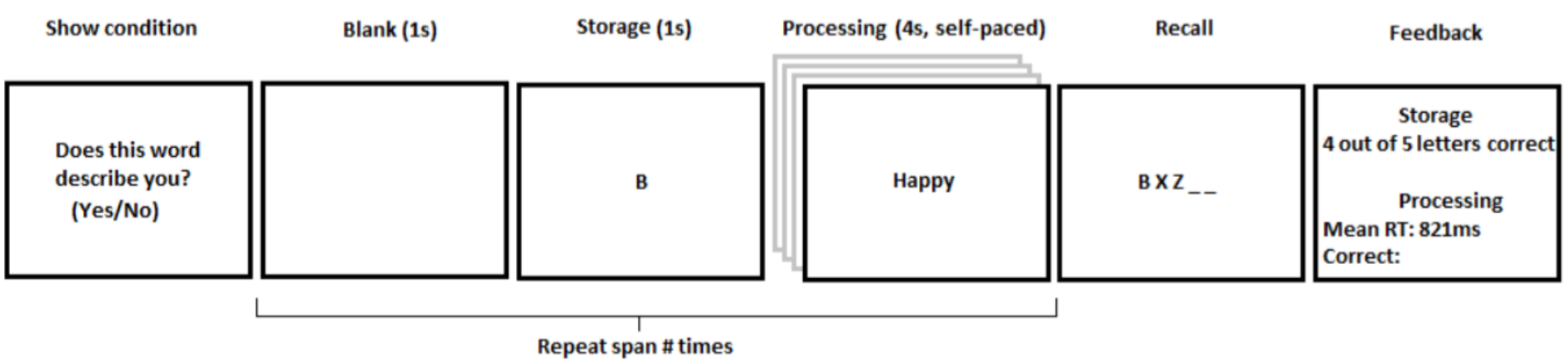
Second example (with data)

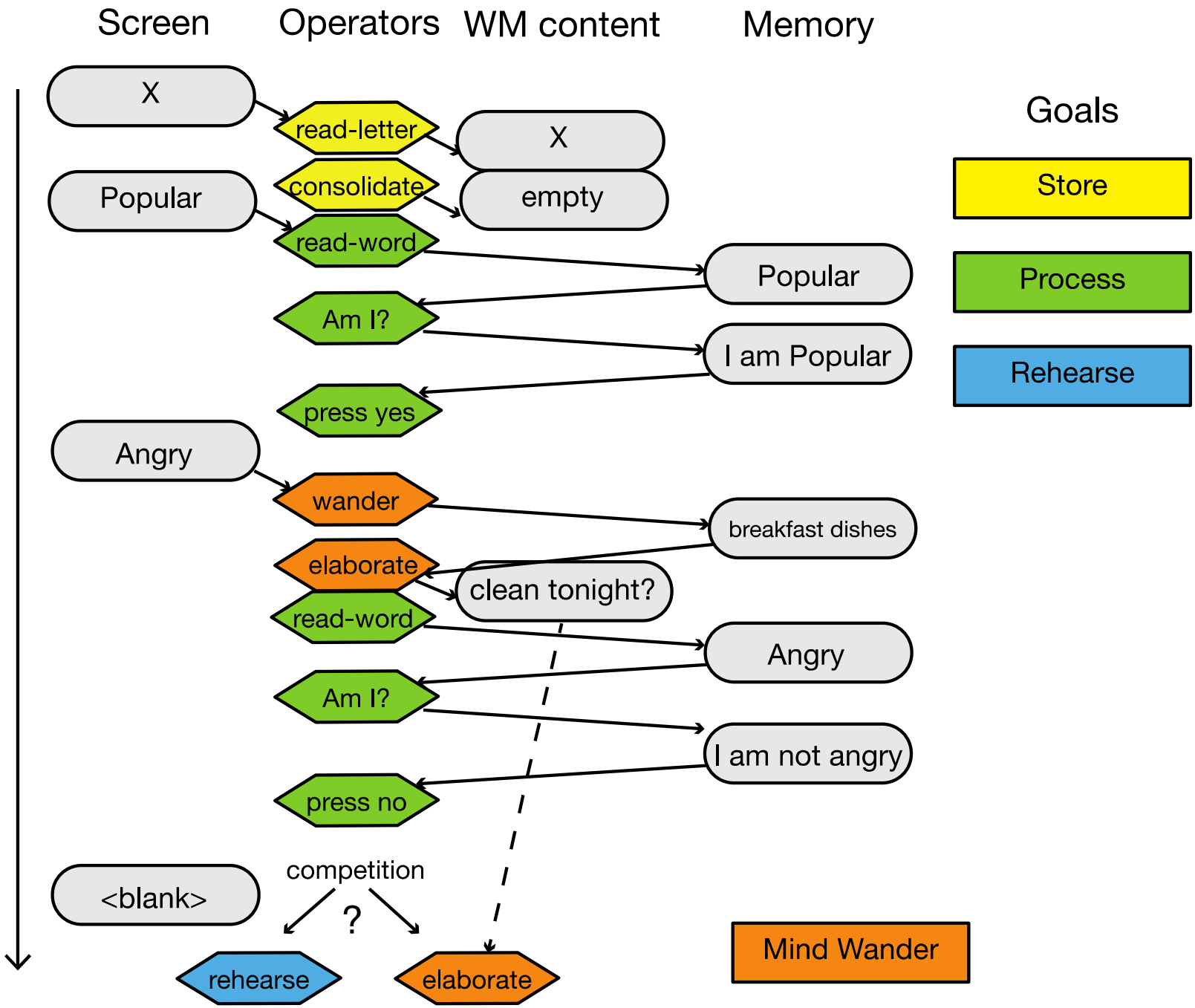
- Distraction/Mind Wandering

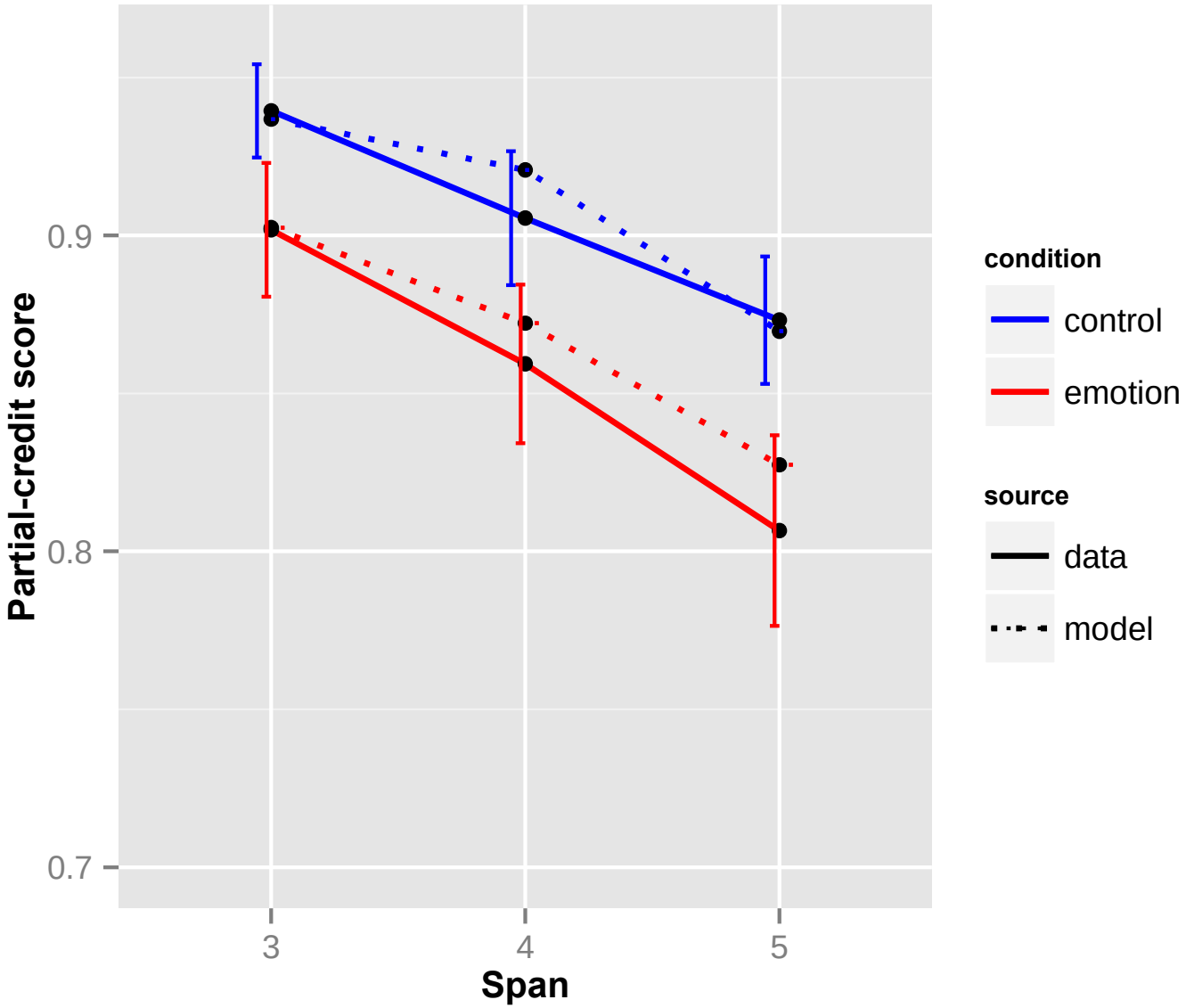
How do we decide what to attend?



Complex WM task




Screen
Operators
WM content
Memory
X
read-letter
X
Popular
consolidate
empty
read-word
Popular
Am I?
I am Popular
press yes
Angry
wander
breakfast dishes
elaborate
clean tonight?
read-word
Angry
Am I?
I am not angry
press no
<blank>
competition
?
rehearse
elaborate
Goals
Store
Process
Rehearse
Mind Wander


Partial-credit score
0.9
0.8
0.7
3
4
5
Span
condition
control
emotion
source
data
model

Discussion points

- From procedural to declarative
  - “Kill you darlings”
- Lower levels are mechanical
- Higher levels deal with meaning and semantics