Semantic Memory (SMem)

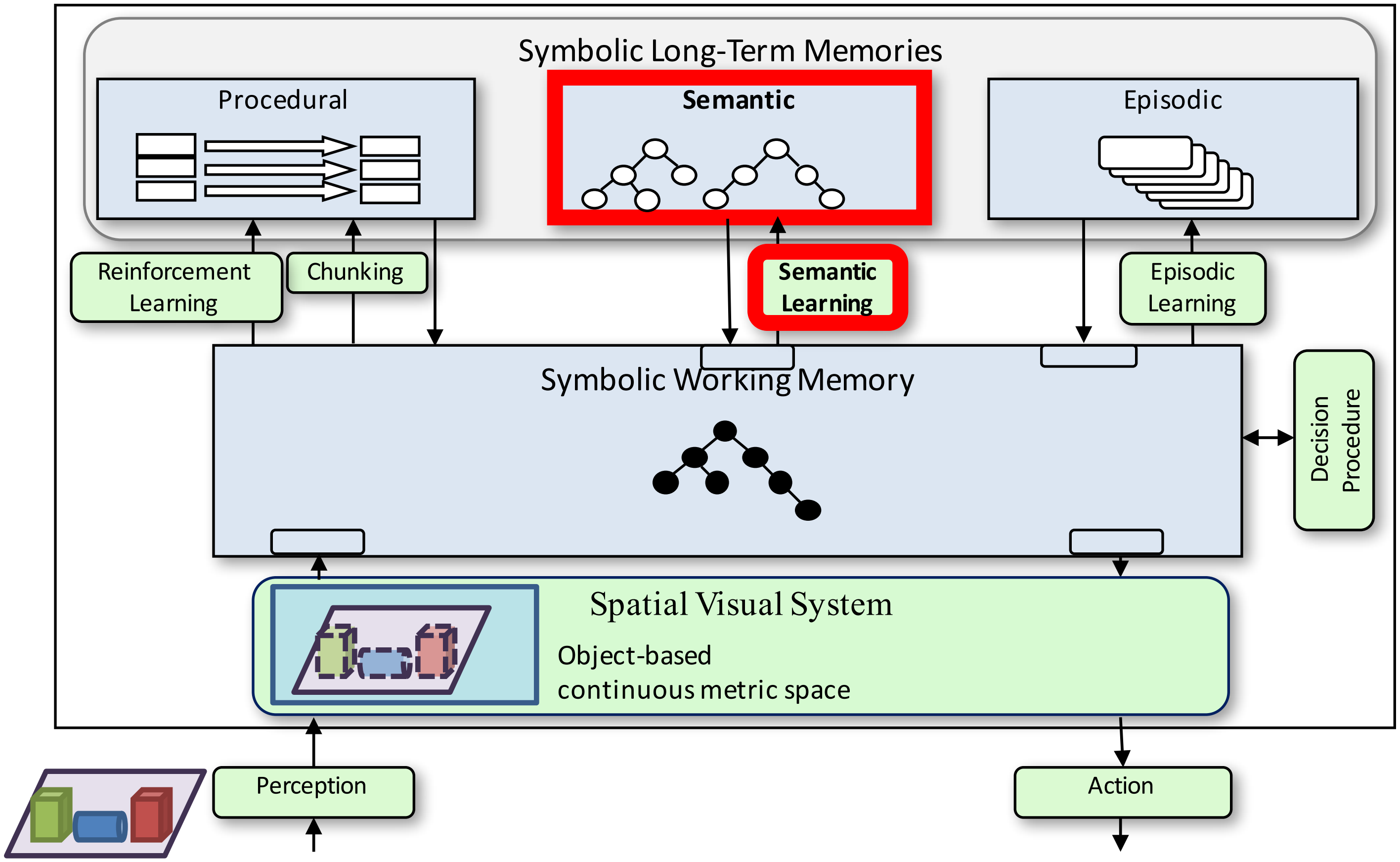
IJCAI 2016

Nate Derbinsky

Agenda

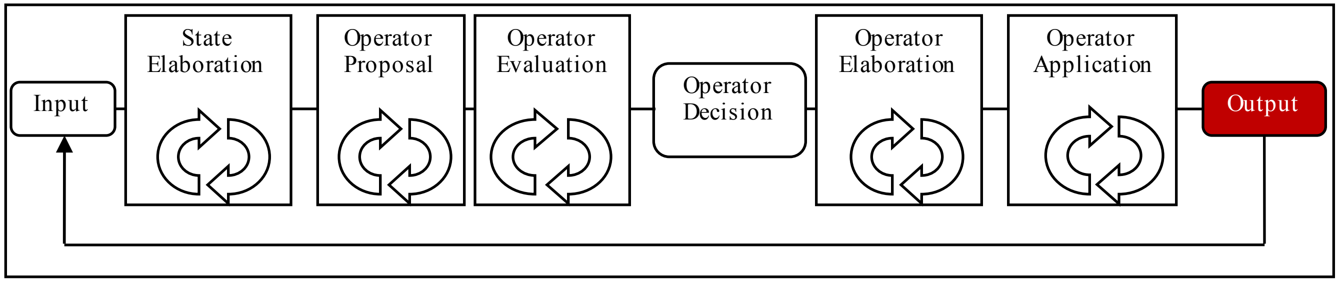
- Big picture
- Basic usage
- Example agents

Soar 9


Symbolic Long-Term Memories
Procedural
Semantic
Episodic
Reinforcement Learning
Chunking
Semantic Learning
Episodic Learning
Symbolic Working Memory
Decision Procedure
Spatial Visual System
Object-based continuous metric space
Perception
Action

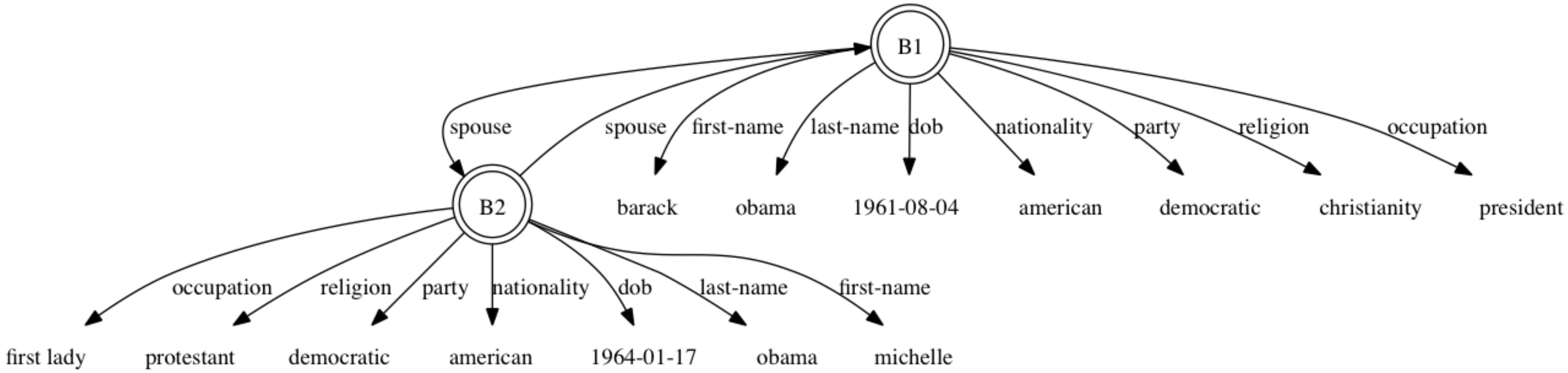
Soar Basic Functions

1. Input from environment
2. Elaborate current situation: *parallel rules*
3. Propose operators via acceptable preferences
4. Evaluate operators via *preferences: Numeric indifferent preference*
5. Select operator
6. Apply operator: Modify internal data structures: *parallel rules*
7. Output to motor system [and access to long-term memories]

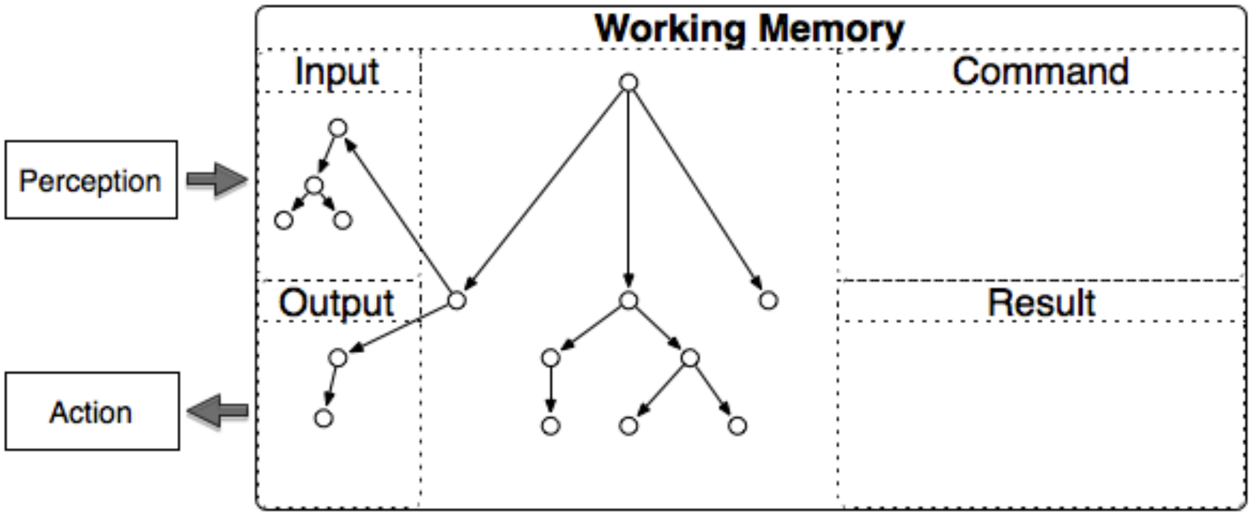


Semantic Memory: Big Picture

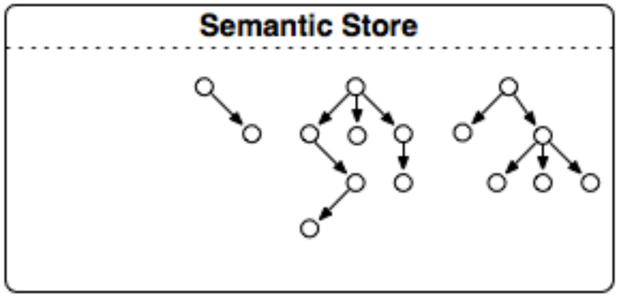
Supports deliberate storage and retrieval of long-term "objects," features, and relations



Architectural Integration

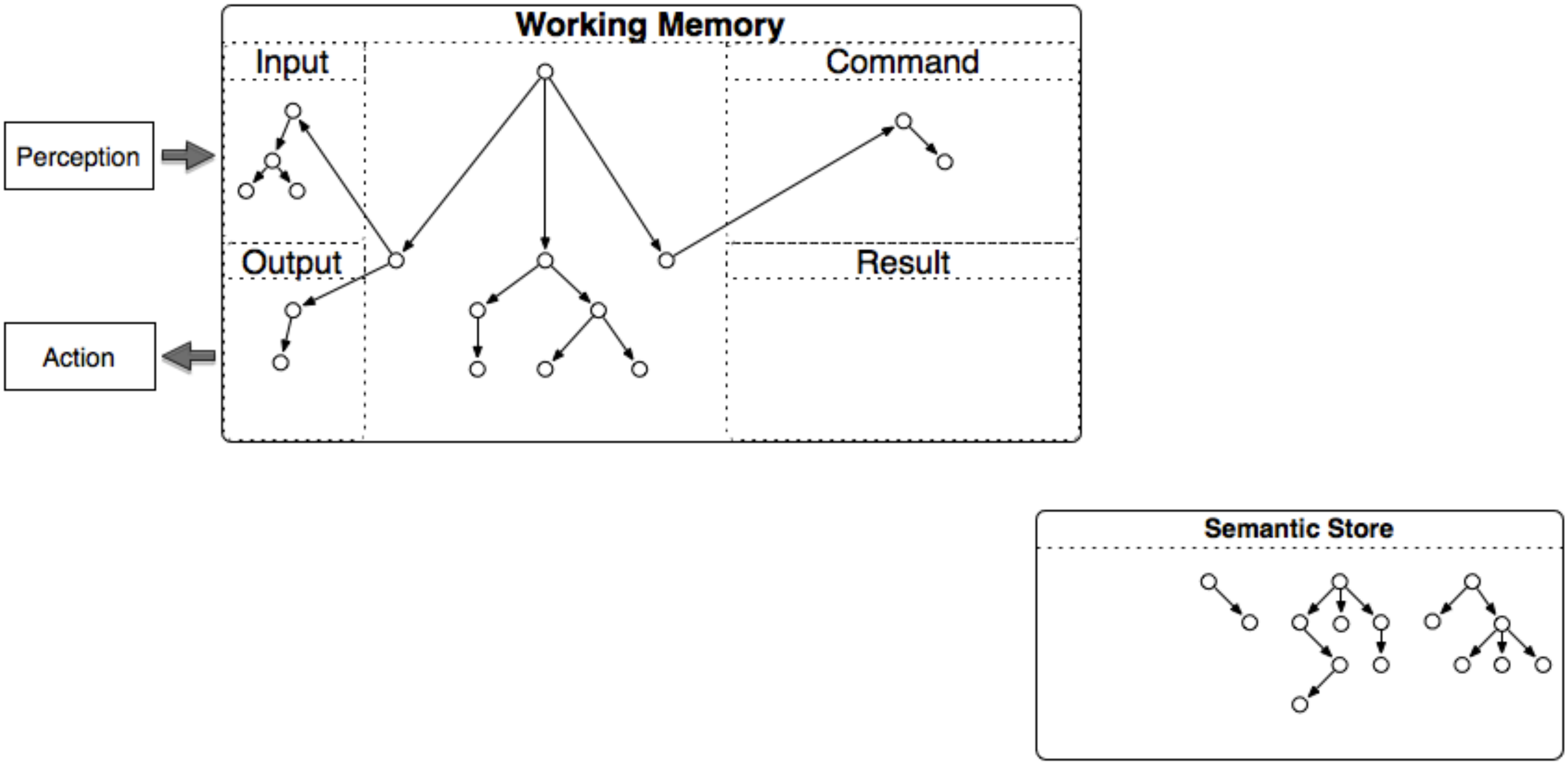


smem "link" available on each state!



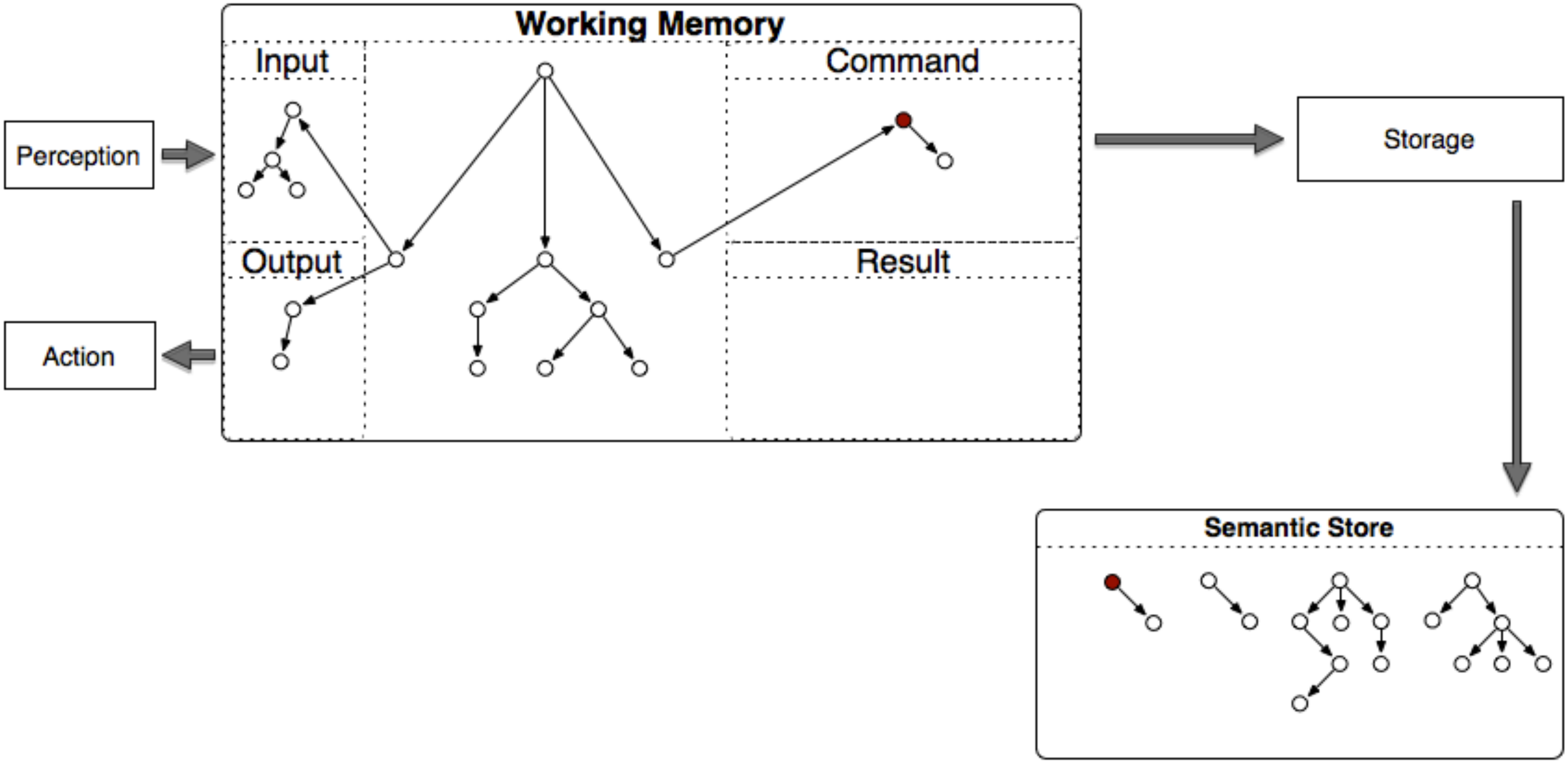
Architectural Integration

Storage



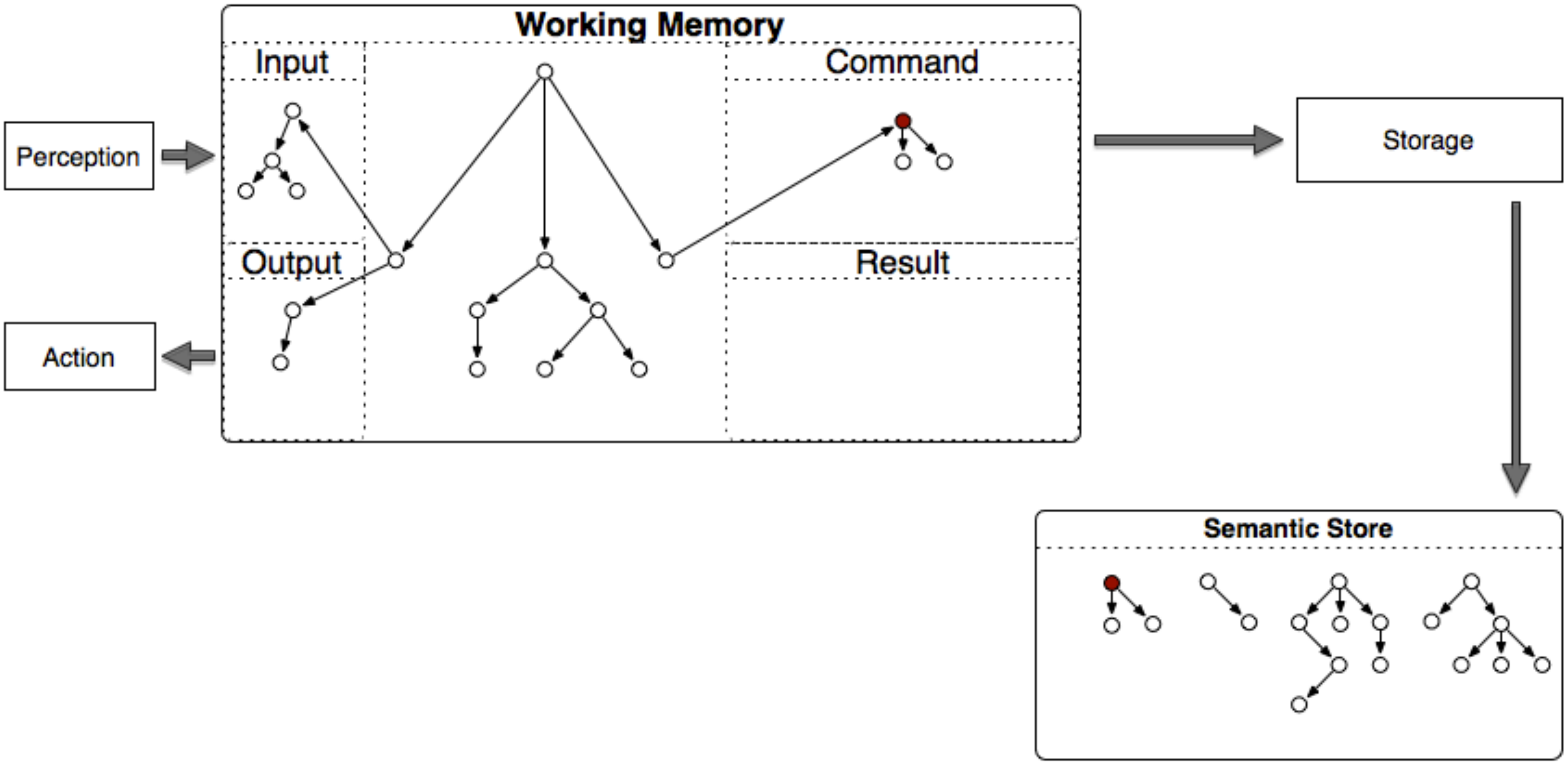
Architectural Integration

Storage



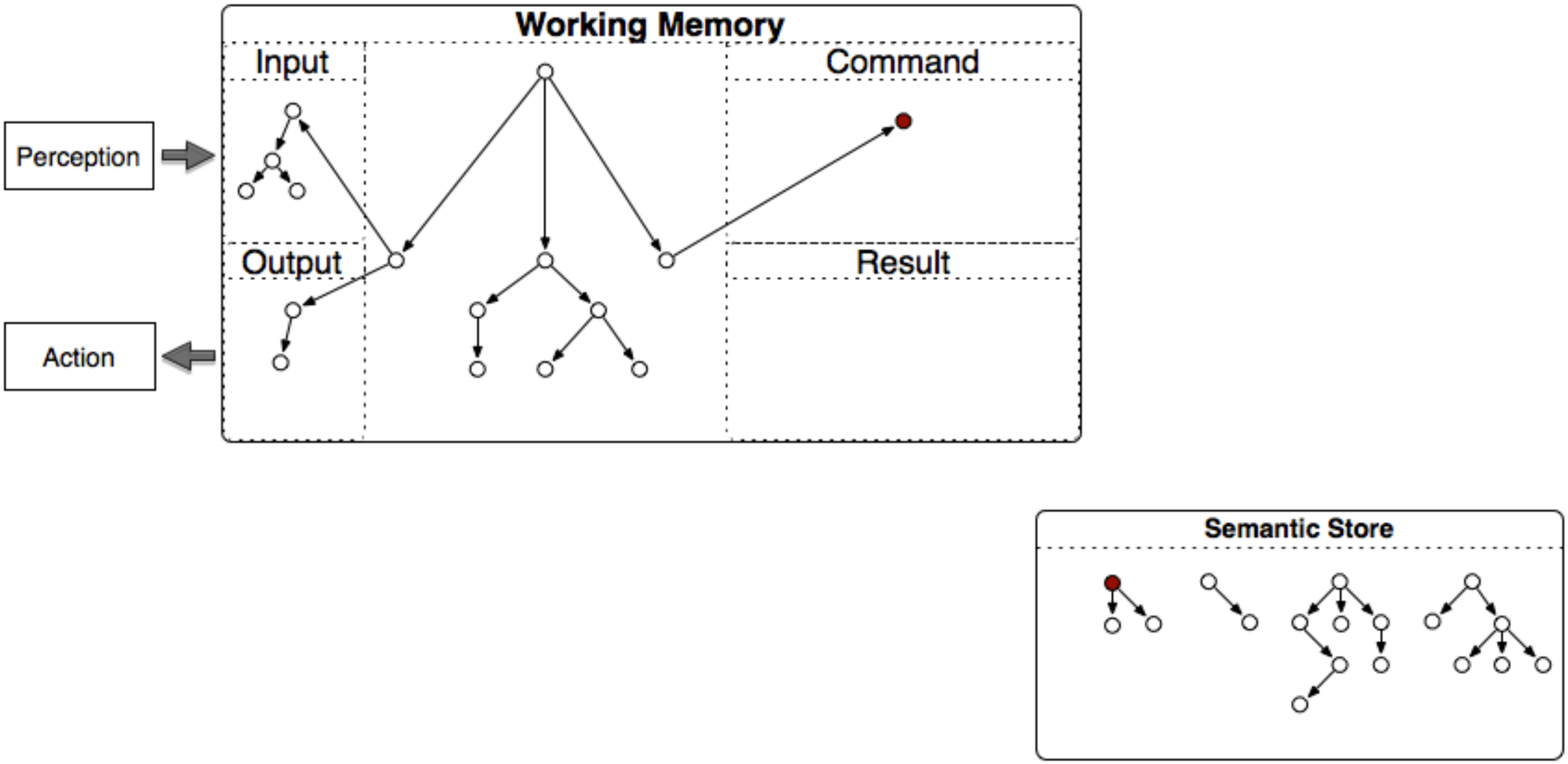
Architectural Integration

Storage



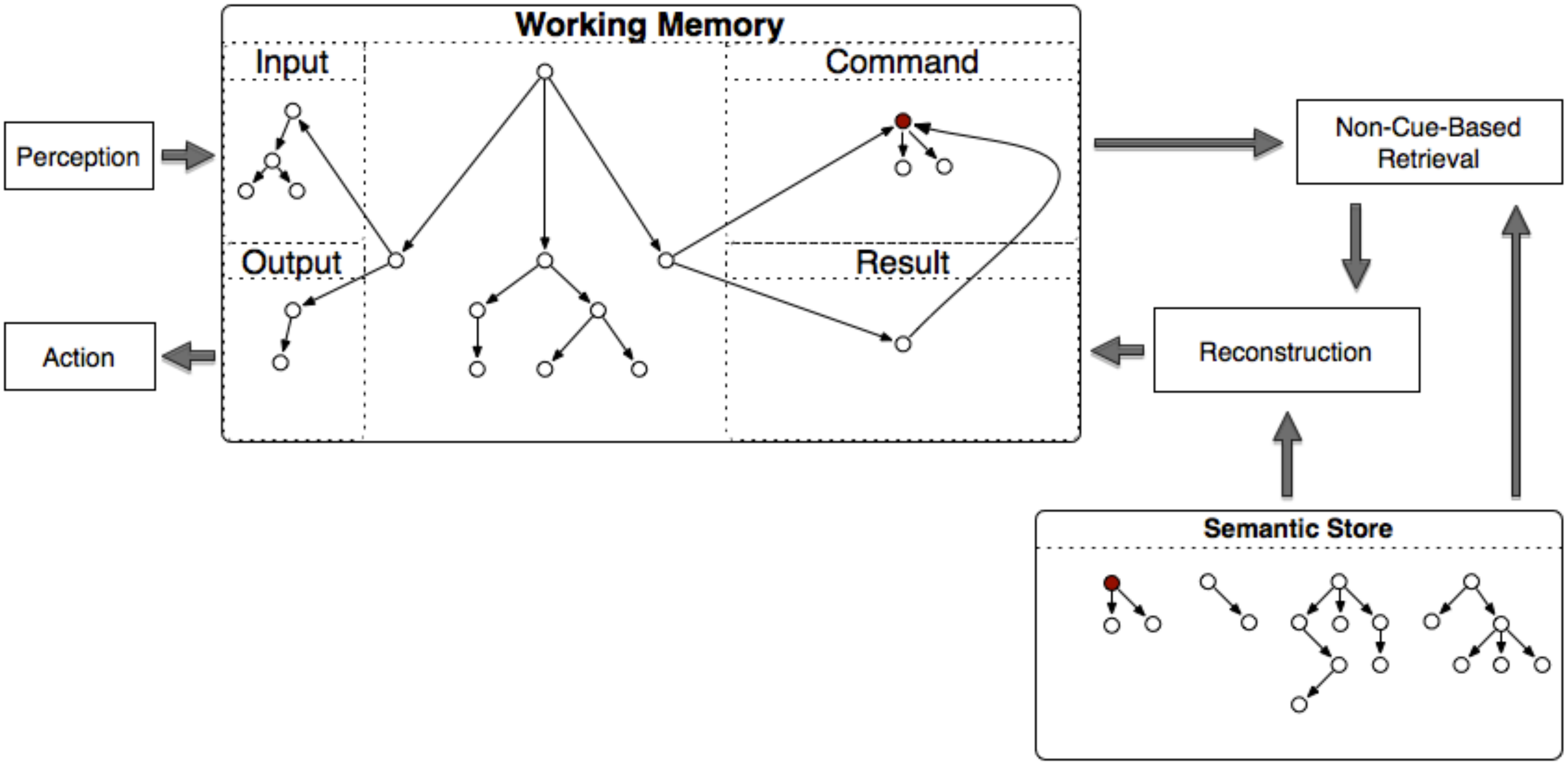
Architectural Integration

Non-Cue-Based Retrieval



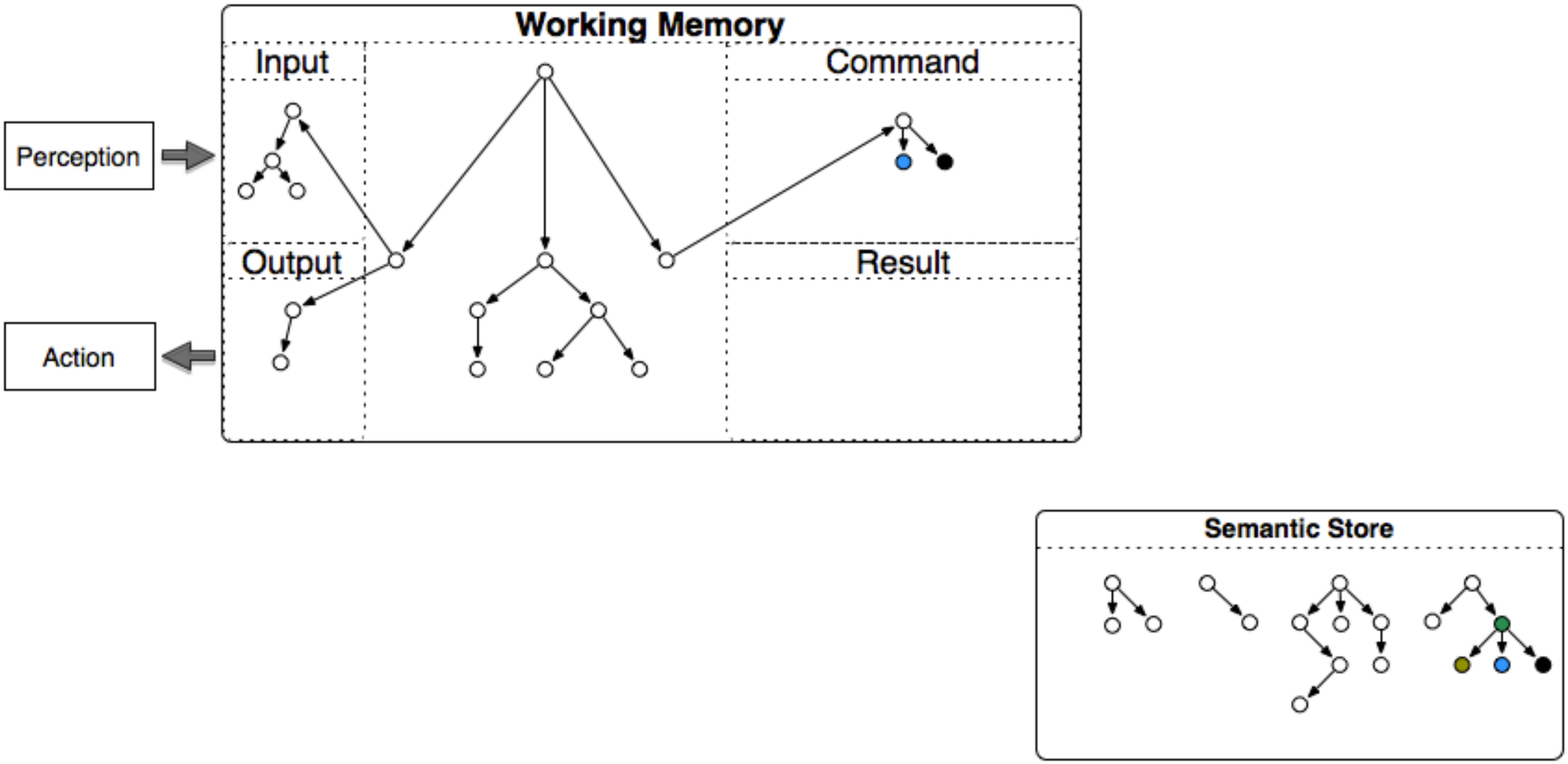
Architectural Integration

Non-Cue-Based Retrieval



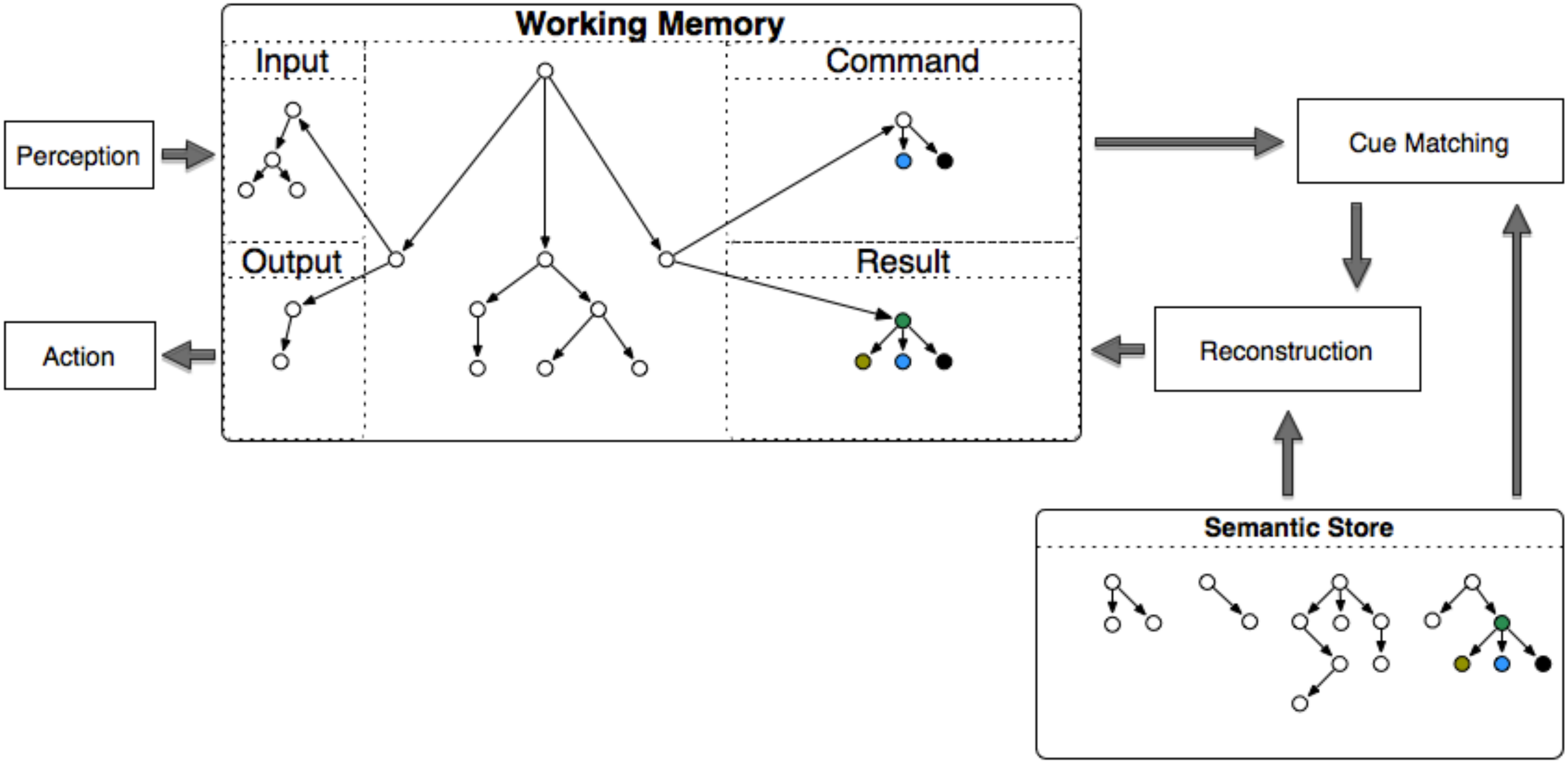
Architectural Integration

Cue-Based Retrieval



Architectural Integration

Cue-Based Retrieval



Storing Knowledge

Manual

Method of appending via command line (especially useful for loading external KBs)

Agent

Deliberate (via rules) addition/modification

Note: both methods can change existing and/or add new knowledge in semantic memory.

Retrieving Knowledge

Non-Cue-Based

Add the features/relations of a known long-term identifier to working memory

Cue-Based

Find a long-term identifier that has a set of features/relations and add it to working memory with its full feature/relation set

Common Constraints:

- Requires that SMem is enabled
- Only one per state per decision
- Processed during *output* phase
- Only re-processed if WM changes to commands
  - Meta-data (status, etc) automatically cleaned by the architecture

Non-Cue-Based Retrieval

Syntax

```
   (<smem> ^command <cmd>)
   (<cmd> ^retrieve <long-term identifier>)
```

Result

```
   (<smem> ^command <cmd> ^result <r>)
   (<cmd> ^retrieve <long-term identifier>)
   (<r> ^<status> <long-term identifier>
        ^retrieved <long-term identifier>)
```

Where `<status>` is…

- `failure`: `<long-term identifier>` is not long-term
- `success`: else (adds all features/relations to WM)

Cue-Based Retrieval: Syntax

```
(<smem> ^command <cmd>)
(<cmd> ^query <q>)
(<q> ^attr1 val1
     ^attr2 <val2>
     ^attr3 @V3 …)
```

The augmentations of the *query* form hard constraint(s), based upon the value type…

- Constant: exact match
- Long-Term ID: exact match
- Short-Term ID: wildcard

Cue-Based Retrieval: Result

```
(<smem> ^command <cmd> ^result <r>)
(<cmd> ^query <q>)
(<r> ^<status> <q>
     ^retrieved <long-term identifier>)
```

Where `<status>` is…

- `failure`: no long-term identifier satisfies the constraints
- `success`: else (adds all features/relations to WM)

Ties are broken by a bias (default: recency)

- See `activation-mode` parameter in Manual
- When you execute `smem -p`, the bias value is indicated

Prohibition

Cue-based retrievals can optionally prohibit the retrieval of one-or-more long-term identifiers

Syntax

```
   (<smem> ^command <cmd>)
   (<cmd> ^prohibit <lti-1> <lti-2> …)
```

WordNet Demo

https://github.com/SoarGroup/Domains-WordNet

- Scripts to convert WN-LEXICAL to SMem
  - Output: `smem --add {` …
    - >821K long-term identifiers, >3.97M edges, ~88MB
  - Source: ~5-10 minutes, ~1GB memory

- SMem uses a SQLite backend
  - Has the ability to save semantic stores to disk and use disk-based databases
    - smem --backup <filename>

WordNet: Make Disk Store

- Soar Java Debugger
  - `source wn.soar`
    - ~2-5 minutes
  - `smem --stats`
  - `smem --backup path/to/filename.db`
    - ~10 seconds

- Soar Java Debugger
  - `smem --set path path/to/filename.db`
  - `smem --set database file`
  - `smem --stats`

WordNet: Representation

“sense” of the “verb” to “soar”

```
smem -q {(<c> ^isa s ^ss-type v ^word soar)}

(@S194181 ^isa s ^sense-number 4
          ^ss-type v ^synset-id 200155406
          ^tag-count 1 ^w-num 1
          ^word soar ^word-lower soar [+194177.000])
```

“gloss” with the “synset-id” 200155406

```
smem -q {(<c> ^isa g ^synset-id 200155406)}

(@G270
    ^gloss |go or move upward; 'The stock market soared after the 
cease-fire was announced'|
    ^isa g ^synset-id 200155406 [+425386.000])
```

WordNet Task

smem-wn-senses.soar

Find all definitions, given lexical word/POS

– High-level algorithm

```
1. query: ^isa s ^word lex ^ss-type pos
2. If successful
   a) query: ^isa g ^synset-id <sense ^synset-id>
   b) If successful
      » write <gloss ^gloss>
   c) prohibit: <sense>
   d) Loop
3. Else
   a) (halt)
```

Eaters!