Control and Metacognition: Insights from EPIC, Soar, and ACT-R

Richard L. Lewis

Department of Psychology
University of Michigan

May 25, 2006

Overview

1 Goals

2 Analysis and critique of current architectures

3 A Three Component Framework for Control

4 Specific proposal for architectural changes

5 Various kinds of nuggets

Main Goal

Develop a **framework for control** that will lead to promising new points in the space of architecures, and a better understanding of existing architectures.

And along the way, lead to better understanding of **architectural requirements for metacognition**.

What makes EPIC control distinctive

1. **Unlimited cognitive parallelism** (directly supports multiple parallel threads of control)
   - Performance is limited by P/M subsystems, learning, and similarity-based interference
2. And therefore **maximum strategic flexibility** in control
   - All control functionality (with exception of synchronous clocking) off-loaded to knowledge
   - Via *task-specific* productions
   - Via *general* productions

What makes ACT-R control distinctive

1. **Serial production rule firing**
2. **Conflict resolution (choice mechanism) determined by learned utility estimates**
   - Choice is function of the rule itself, *not* the rule + context
   - See, for example, Marsha Lovett's stick-building task model—she was forced to split up two simple strategies ("under-shoot" and "over-shoot") into multiple productions in order to learn the appropriate contextual utilities

Some key differences

- Differences in architectural **choice mechanism**
  - **EPIC**: none
  - **ACT-R**: context-independent utility
  - **Soar**: decision procedure; integrates over arbitrary context-dependent knowledge (elementary deliberation)
- Differences in **default speed-accuracy tradeoff**
  - Soar *impasses* (takes no action) at first sign of trouble
  - If ACT-R *can* do something, it *will*
- Differences in **automatic detection and representation of meta-information about the choice**
  - Soar can detect and represent (architecturally) lack of knowledge, including *response* and *cognitive conflict*
  - ACT-R and EPIC are blissfully oblivious
  - Functional implication of this is a difference in the support for deliberation in *novel* contexts

A Three Component Framework for Control

Productions

Choice mechanisms

Monitors

(Controlled subsystems)

Productions provide:

- Computational completeness
- Arbitrary, fine-grained contingencies for behavior

Choice mechanisms provide:

- Support for choice, possibly deliberation; locus of learning of control

Monitors provide

- Immediate representation of information about *internal processing state* in a form that control can be made contingent upon
- Integration over different time scales, modules

Controlled subsystems provide

- Internal agent resources (LTM, WM, perception, motor, etc.)
- "Control knobs" (commands, inhibition, etc.)

Examples of existing monitors

ACT-R

- Motor module state flags
- Retrieval module state flags (failure)
- Temporal module (?)

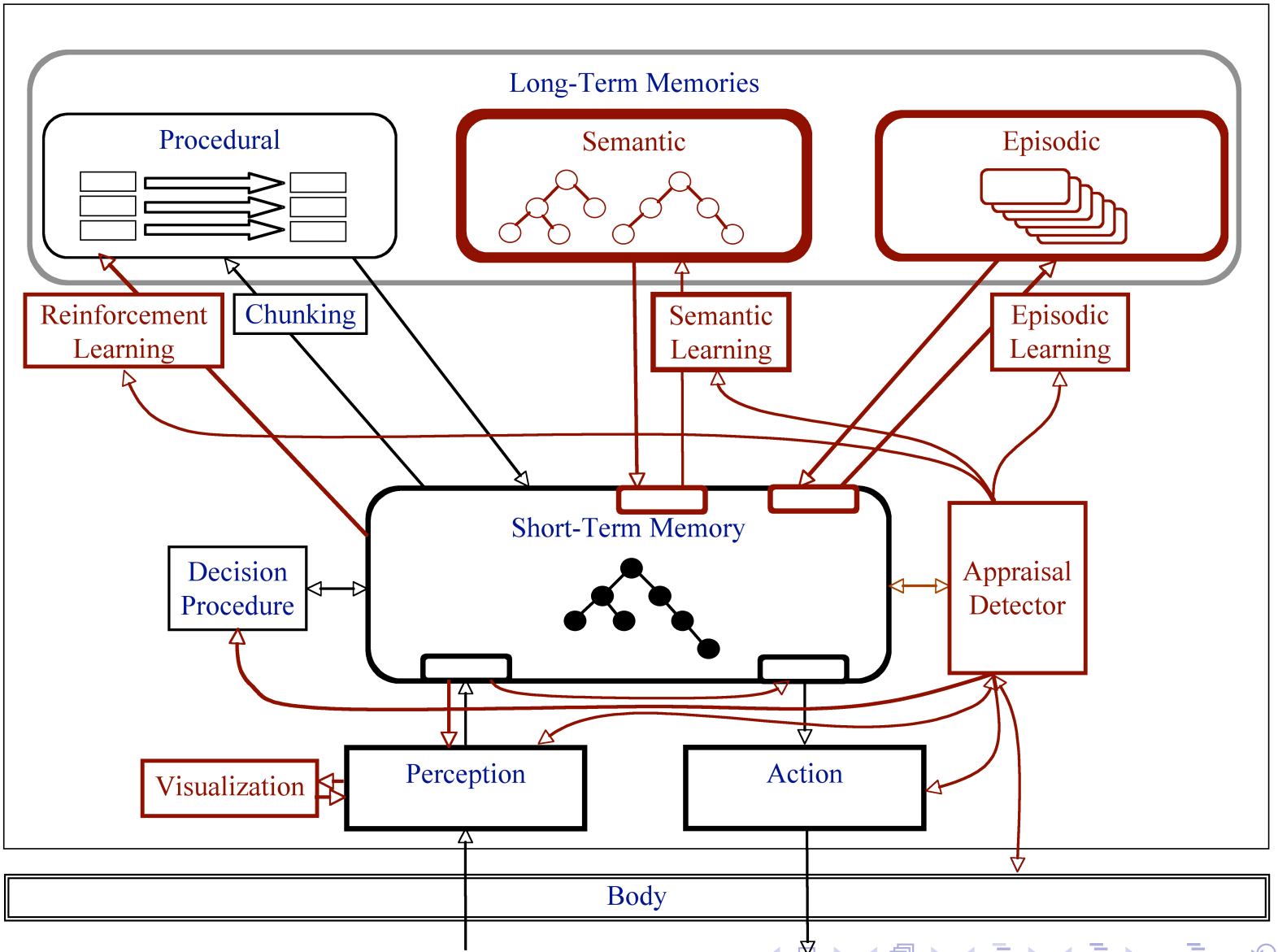
EPIC

- P/M module state flags (e.g., "busy")

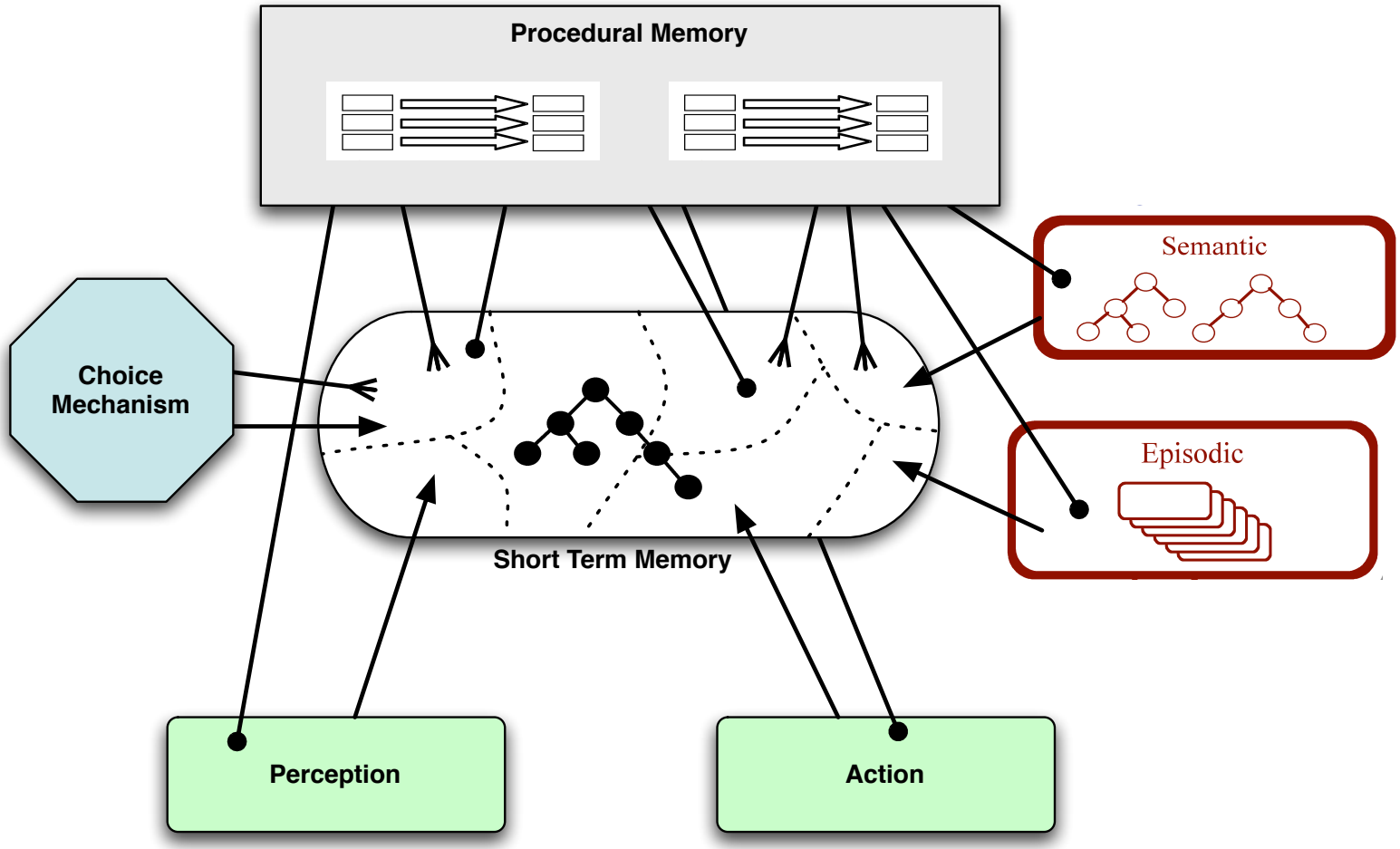
Soar

- Impasse detection (state no-change, tie)
- Parts of the emotion system

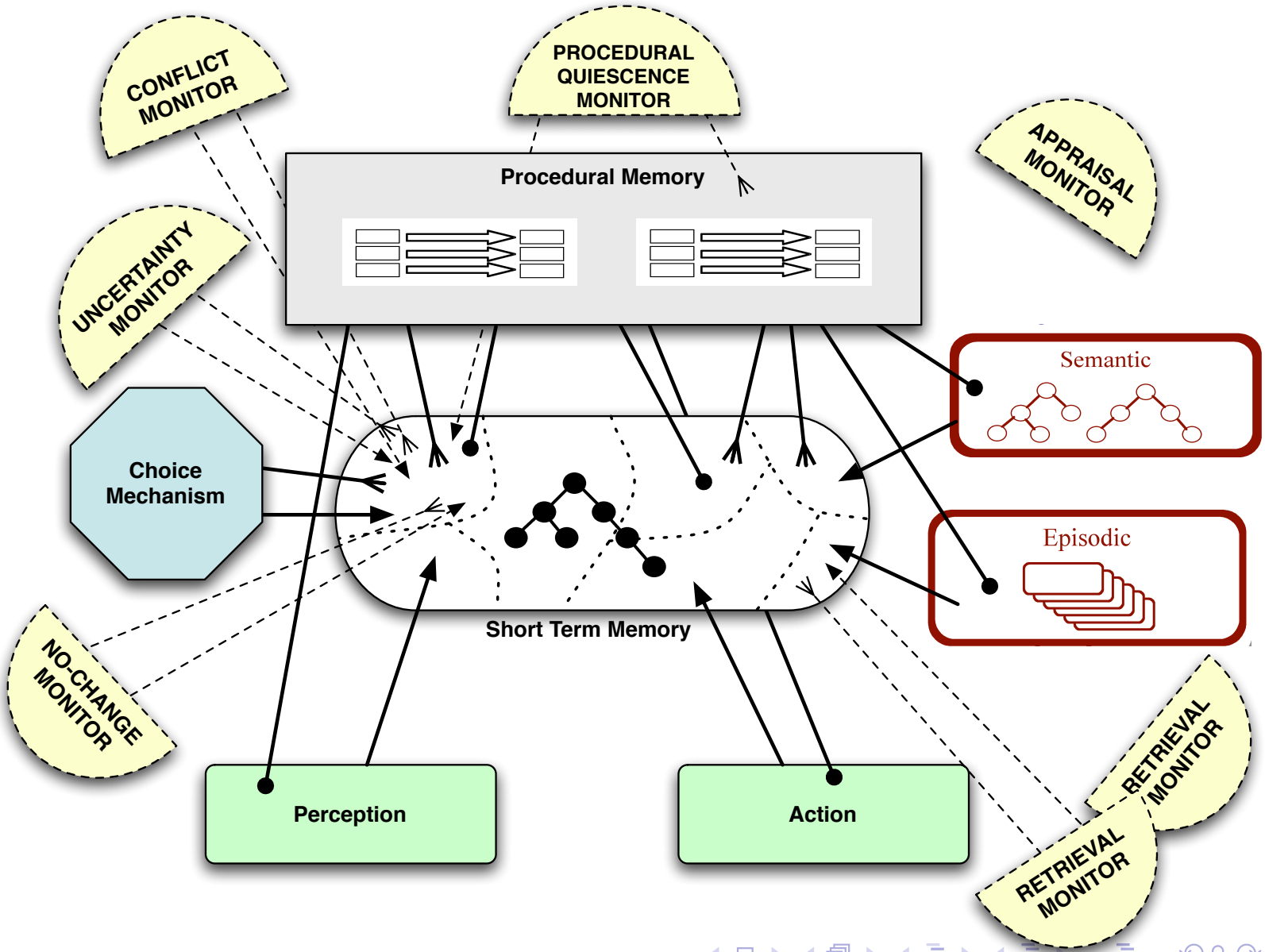
The Old New Soar



The New New Soar



The New New Soar



Specific proposal: Part 1

Monitors Should Be First-class Architectural Objects

- Should be implemented as separate modules, with well-defined interfaces
- Should not be restricted to monitoring a single other module; could in principle monitor multiple modules
- Should not be restricted to a particular time course

Specific proposal: Part 2

Choice Mechanisms Should Be First-class Architectural Objects

- Decision procedure should be independent of quiescence detection and impasse detection
- Should be possible to create multiple (asynchronous) decision/control streams with multiple instantiations of choice mechanisms

Specific proposal: Part 3

Implement and explore the following monitors

1. **Procedural Slack Monitor** (aka "Quiescence Detector")
   - Input from *procedural* and *temporal modules*
   - Output to buffer a representation of time passed since last production fire; analogous to failure in retrieval buffer
2. **Cognitive Operator Slack Monitor** (aka "State No Change Impasse Detector")
   - Input from *WM for operator* and *temporal modules*
   - Output to buffer a representation of time passed since last change in operator representation

Specific proposal: Part 3, continued

3. **Decision Uncertainty Monitor** (aka "Tie Impasse Detector")
   - Input from decision mechanism (e.g. confict resolution in ACT-R)
   - Output representation of how "close" the decision was
   - Perhaps the decision mechanism could be modulated in continuous way to achieve various SAT's (purely under knowledge-driven control)
4. **Retrieval Uncertainty Monitor** (aka "Andrew's Match Meta-data")
   - For both Episodic and Semantic memories
5. **Perception-of-Feeling Monitor**
   - Transforms information about emotional state into form accessible to cognition

A decomposition of Soar decisions

DECISION CYCLE =

Cognitive-Operator-Slack-Monitor +
Decision-Uncertainty-Monitor +
Procedural-Slack-Monitor + Decision-Mechanism

Possible Golden Nuggets

1. Provide new class of key functional modules to **guide mapping from functional architecture to brain**
2. New Soar or ACT-R could **detect (the degree to which) its current knowledge is insufficient**, even more flexibly than Soar can now
   - But could choose to use (or ignore) that information
3. The resulting architecture should be **more flexible and "temporally situated" than either current Soar or ACT-R**, and **better able to deal with novel contexts than EPIC**
   - Good evaluation domains: Tasks involving *speed-accurcay tradeoffs* at multiple time scales

Possible Coal

- Soar and ACT-R lose their distinctiveness and relative competitive theoretical advantages
- ???
- (It's easy to have a tiny haul of coal when no systems have been built. . .)