Levels of Learning in Natural and Artificial Agents

(FA9550-19-0180)

PI: John Laird (University of Michigan)
Co-PI: Shiwali Mohan (PARC)
Graduate Student: Bryan Stearns (UM)

AFOSR Program Review:
Computational Cognition and Machine Intelligence Program
(October 8, 2020; Cyberspace)

How is Learning Integrated with Performance for Autonomous Agents in Dynamic Environments?

- Key Challenge: Performance & learning must be real-time
  - Must keep up with environmental dynamics with bounded computation.
  - Even as long-term knowledge grows.
  - Across the breadth learning we find in humans.

- Surprisingly little analysis on complexity of online learning algorithms in AI.

Possible Types of Learning

Self-Explanation

Recognition

Discovery

Episodic Learning

Learning by Analogy

Category and Concept Learning

Learning by Instruction

Sequence Learning

Learning by Demonstration

Rehearsal

Procedure Learning

Meta-Learning

Temporal-Difference Learning

Experimentation

Imitation Learning

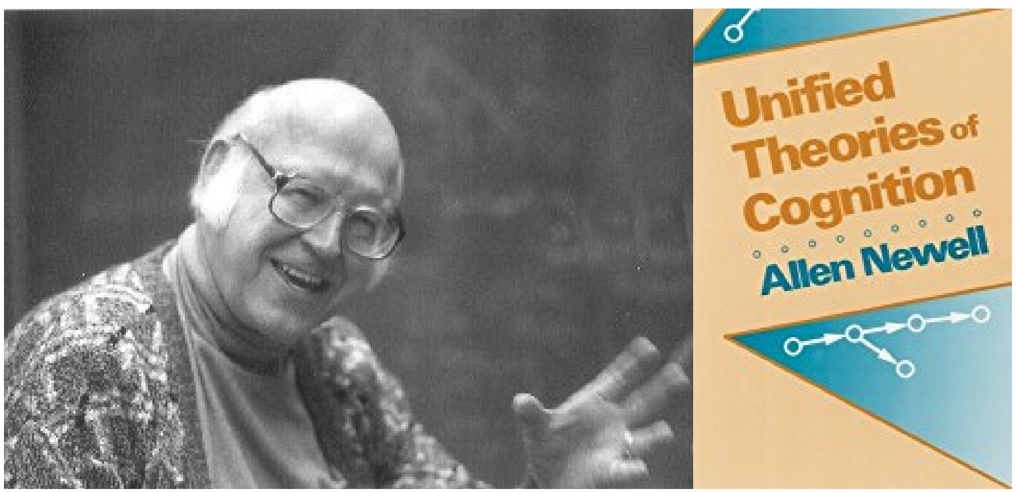
Perceptual Learning

Practice & Rehearsal

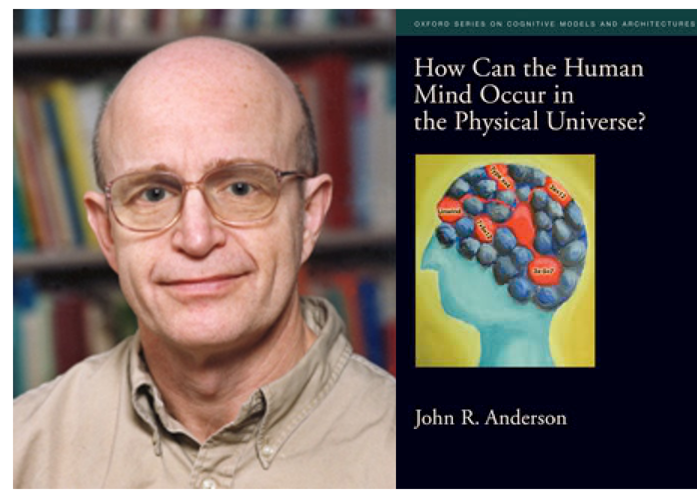
How are these types of learning integrated in agent architecture?

Cognitive Architecture Hypothesis

- Complex cognition arises from a combination of:
  - a fixed set of computational building blocks (memories, processes, representations, learning mechanisms); and
  - knowledge (innate and learned through experience).



Allen Newell



John Anderson

Our Hypothesis: Two Levels of Learning

1. Architecture mechanisms (L1) for *basic* learning
    - Innate, automatic, continuous, online, not under agent control.
    - Parasitic process on top of task performance.
    - Small fixed number of mechanisms.
    - Examples: production composition, chunking, episodic memory storage.
2. Knowledge-based strategy (L2) for *complex* learning
    - Hijacks task processing via metacognition and deliberation.
    - Create experiences for L1s: no additional learning mechanisms.
    - Can be learned – no fixed number.
    - Examples: learn from instruction; retrospective analysis; deliberate training; experimentation or exploration; ...

Level 1

Level 2

Self-Explanation

Recognition

Discovery

Episodic Learning

Learning by Analogy

Category and Concept Learning

Learning by Instruction

Sequence Learning

Learning by Demonstration

Rehearsal

Procedure Learning

Meta-Learning

Temporal-Difference Learning

Experimentation

Imitation Learning

Perceptual Learning

Practice & Rehearsal

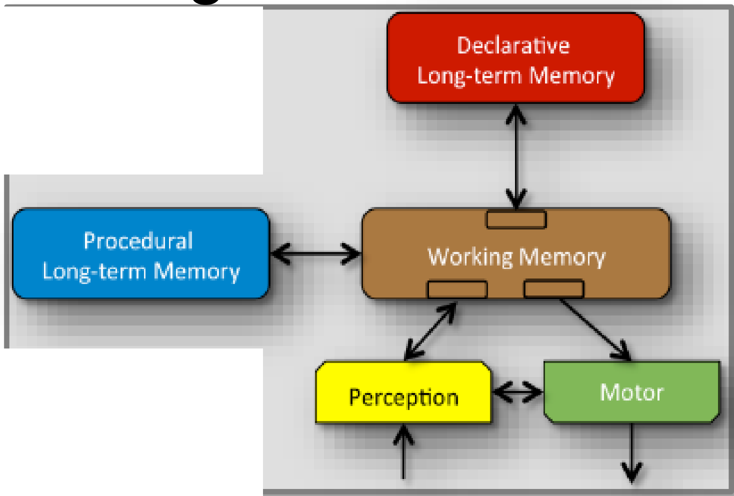
Test Hypothesis: Common Model of Cognition

Dig into how performance and learning are integrated in Common Model for architectural learning (L1).

- What processing is done during task performance?
- How is learning integrated with performance?
  1. What data (and *metadata*) is used for learning?
  2. What processing is performed with that data?
  3. What is the computational complexity of learning?

Common Model of Cognition

- An abstract specification of human-like cognitive architecture
  - Community consensus/agreement
  - Not itself a cognitive architecture
  - Still missing many components
    - Perceptual and motor learning



- A set of communicating processing and memory modules
  - Processing is parallel across modules
  - Processing is parallel within modules
  - Learning mechanisms associated with many modules
- Cognitive cycle: sequential actions
  - Complex behavior arises from a sequence of cognitive cycles:
    - Each cycle is 50msec in humans
    - No additional modules for complex cognition

Laird, J. E., Lebiere, C. & Rosenbloom, P. S. (2017). A Standard Model for the Mind: Toward a Common Computational Framework across Artificial Intelligence, Cognitive Science, Neuroscience, and Robotics , *AI Magazine 38(4).*

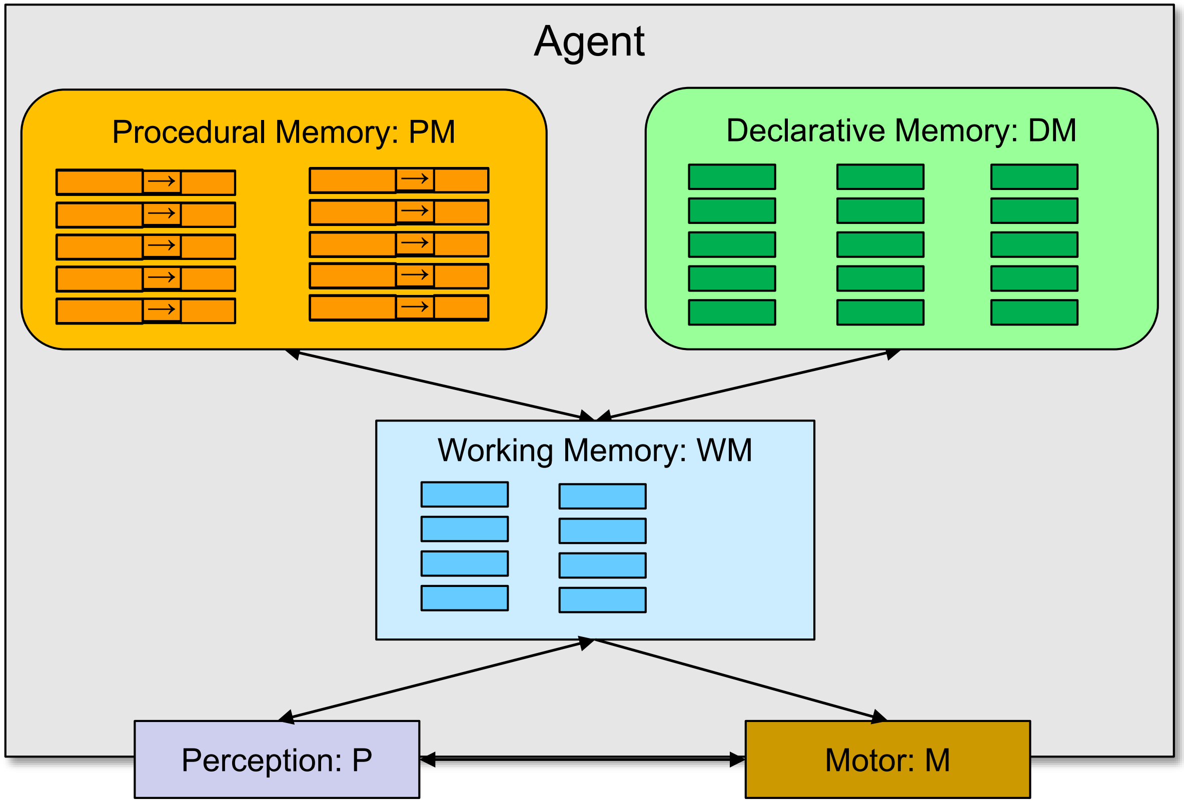
Why Common Model?

1. Covers a range of implemented, well-researched cognitive architectures: Soar, ACT-R, Sigma, Spaun, LIDA, …

2. Includes multiple architectural learning mechanisms.

3. Has strong connections to the human mind and brain.
   – *Andrea Stocco presentation @2:15pm.*

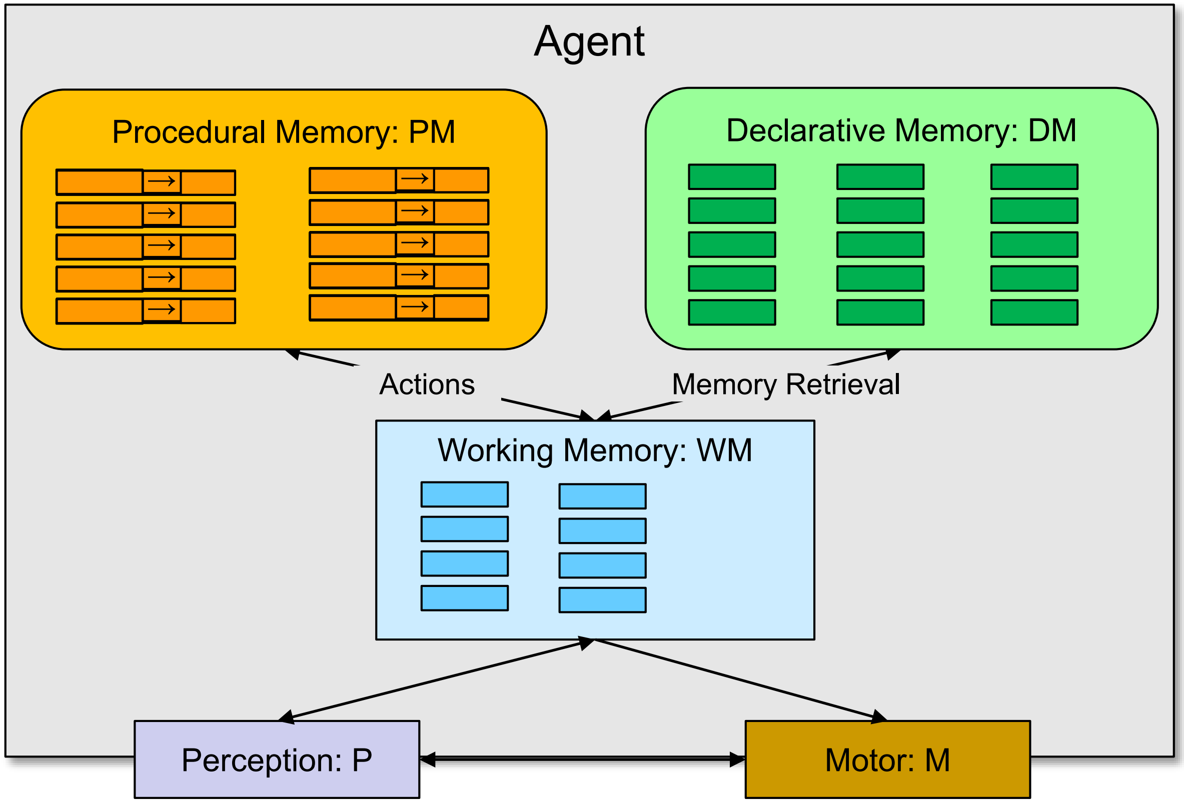
Agent Architecture: Data/Knowledge



- Architecture processing and structure is fixed, not open to learning
- Memories contain independent knowledge elements.

- Working Memory [WM]: Perception, situational awareness, goals, intentions, hypotheticals, …
- Declarative Memory [DM]: Facts, beliefs, experiences, …
- Procedural Memory [PM]: Skills, procedures, goal structures, ...

Agent Architecture: Data/Knowledge

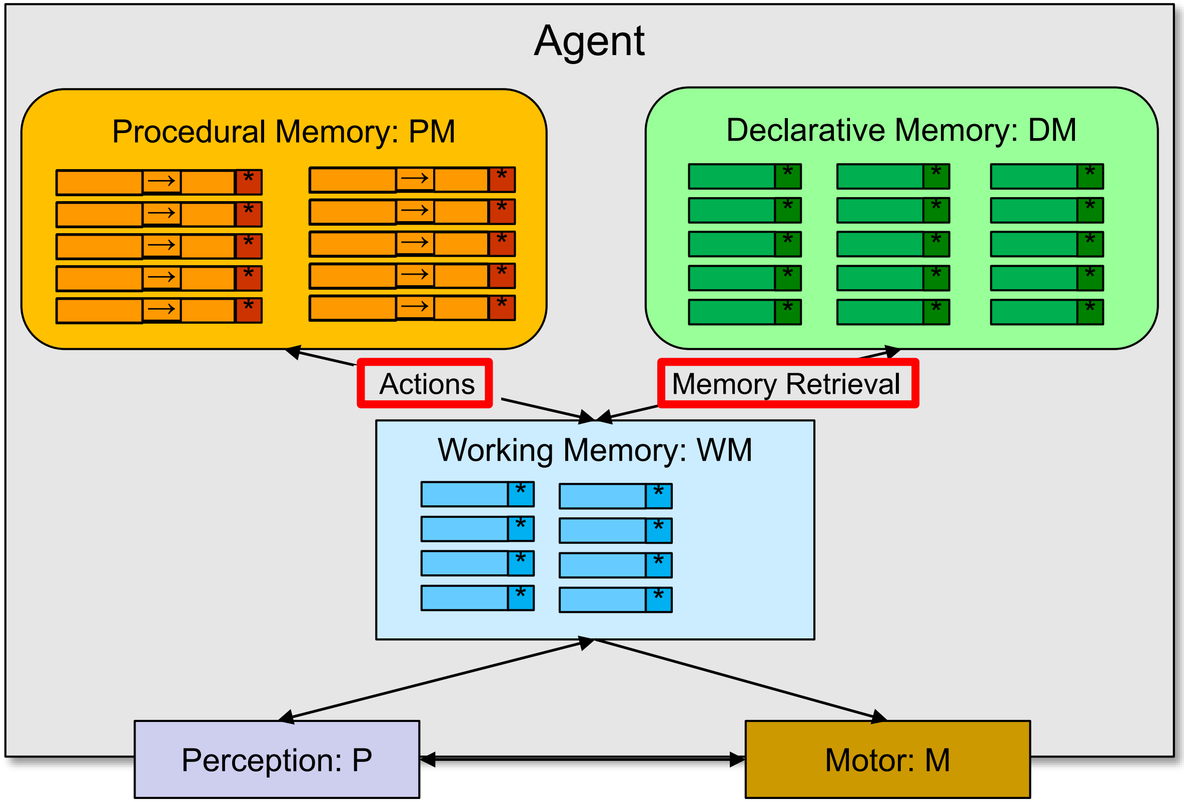


Main computational cost is accessing PM and DM <u>relative to WM</u>.

Everything else is cheap.

- Perception: Change to WM from sensors
  - <u>ΔP</u> → ΔWM
- Internal Action: Changes to WM initiated by procedural memory (PM)
  - <u>ΔWM</u>; <u>WM</u>; PM → ΔWM
- DM Retrieval: Cue in WM leads to retrieval from declarative memory (DM) into WM
  - <u>ΔWM</u>; <u>WM</u>; DM → ΔWM
- Motor Action: Command in WM is sent to motor system
  - <u>ΔWM</u> → ΔM

Agent Data & Meta Data

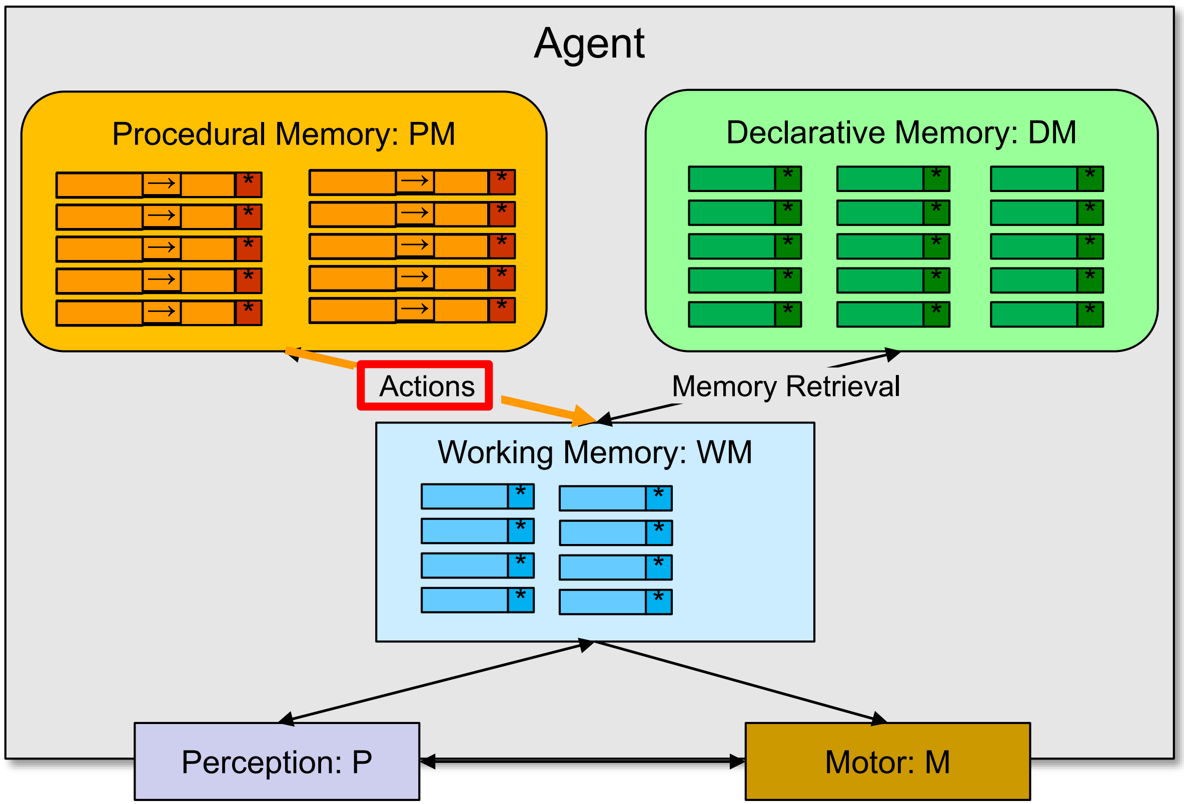


Metadata = numeric/statistical data associated with agent data.

Not directly testable by procedural memory.

- Working Memory Metadata [WM*]: History of creation & access → activation.
- Declarative Memory Metadata [DM*]: History of creation & access → activation.
- Procedural Memory Metadata [PM*]: History of access, expected utility, …

Agent Internal Action: with Metadata



Learning = change to long-term memory structures

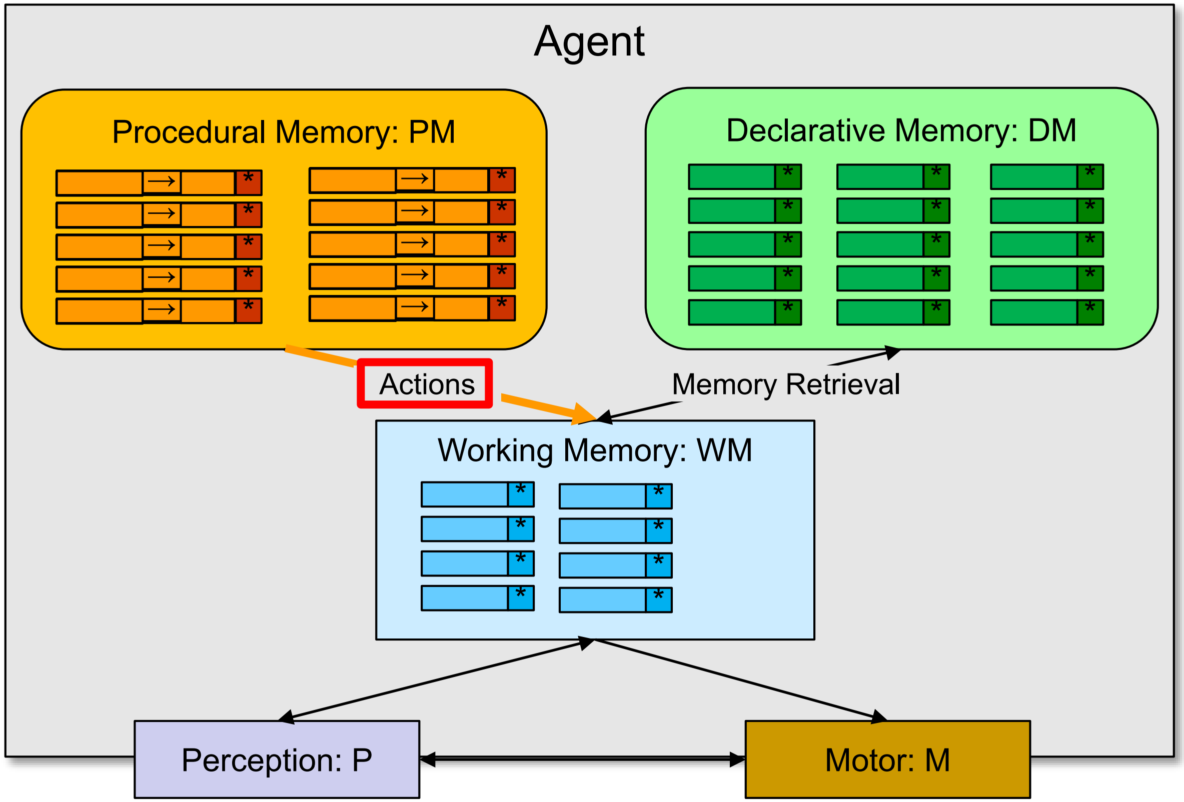
Learning! [RL]

Action: ΔWM; WM; PM → ΔWM

with PM* & WM* ΔWM; WM; WM*; PM; PM* → ΔWM; ΔWM*; ΔPM*

- Procedural metadata [PM*] and working memory metadata [WM*] influence which available actions [PM] modify WM.
- Working memory [WM*] and procedural memory metadata [PM*] are updated.

Agent Action with Learning

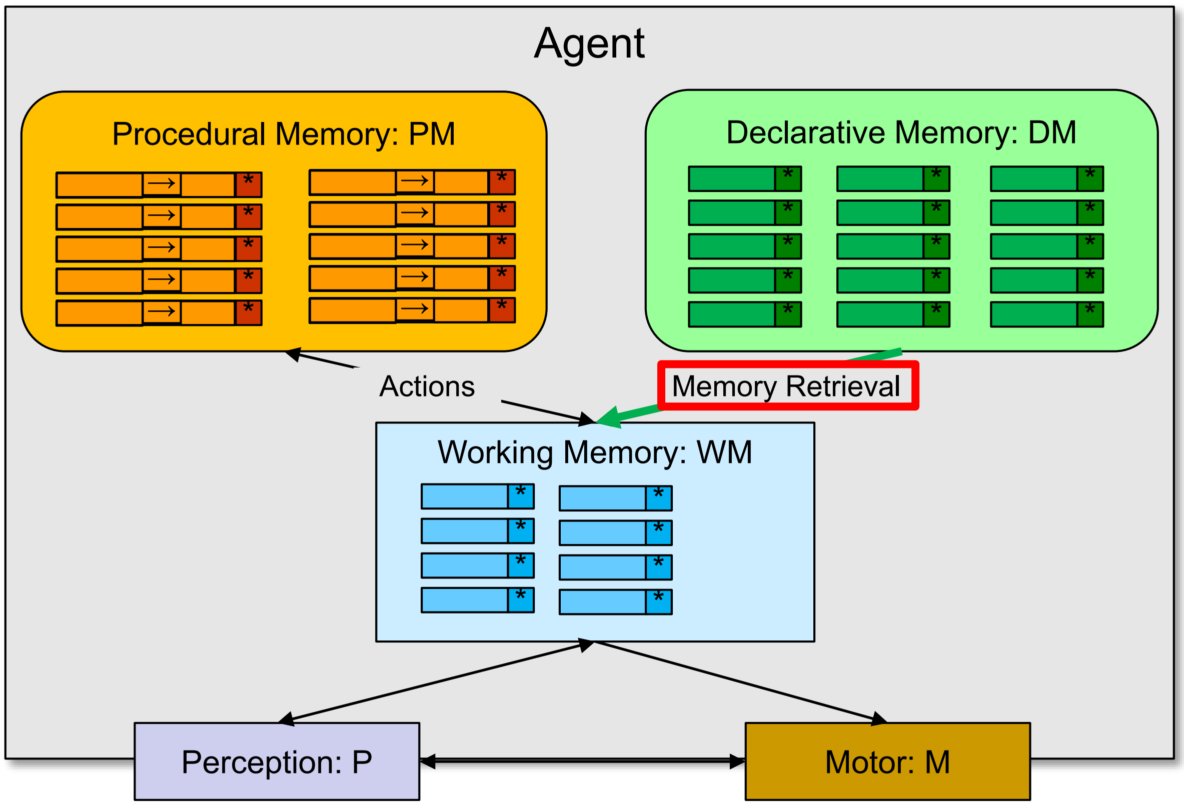


Learning

Action:	ΔWM; WM; PM	→ ΔWM
with PM* & WM*	ΔWM; WM; WM*; PM; PM*	→ ΔWM; ΔWM*; ΔPM*
with learning	ΔWM; WM; WM*; PM; PM*	→ ΔWM; ΔWM*; ΔPM*; ΔPM; ΔDM; ΔDM*

- Learn new procedural knowledge (chunking, procedure composition)
- Learn new declarative knowledge (declarative learning – semantic/episodic)
- Update declarative memory metadata (recency, frequency, …)

Agent DM Retrieval with Learning

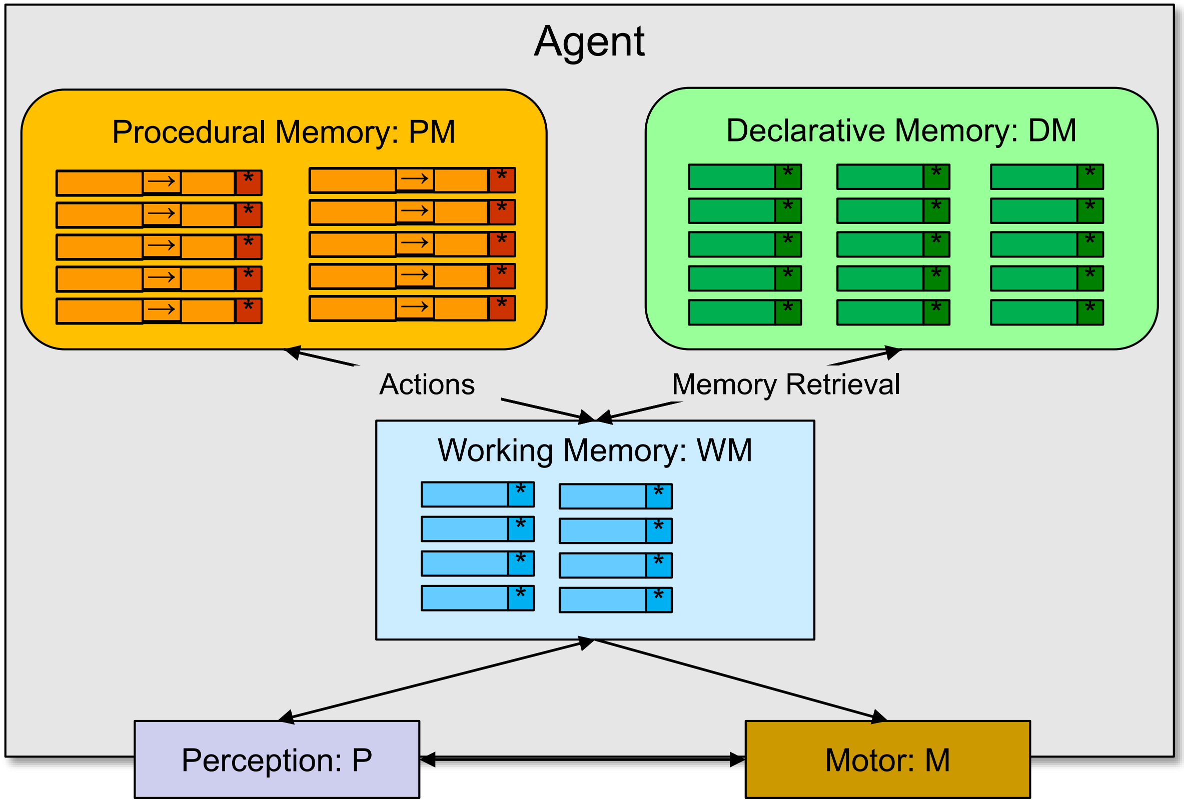


DM Retrieval: $\underline{\Delta WM}$; $\underline{WM}$; $DM \rightarrow \Delta WM$

with DM* & WM*: $\underline{\Delta WM}$; $\underline{WM}$; WM^*; DM; $DM^* \rightarrow \Delta WM$; ΔWM^*; $\boxed{\underline{\Delta DM^*}}$

Declarative memory metadata [DM*] and working memory metadata [WM*] influence which structure is retrieved from declarative memory [DM].

Agent Metadata Maintenance & Forgetting



Local calculations that are easily parallelizable

- WM decay: WM* → ΔWM*
- PM decay: PM* → ΔPM*
- DM decay: DM* → ΔDM*

Update recency/frequency metadata of unchanged data.

- WM Forgetting: ΔWM* → -WM
- PM Forgetting: ΔPM* → -PM
- DM Forgetting: ΔDM* → -DM

Removed data that has decayed beyond threshold.

Summary: Performance and Learning

1. Task Performance: Changes to working memory
   - ΔWM; WM; WM*; PM; PM* → ΔWM
   - ΔWM; WM; WM*; DM; DM* → ΔWM
   - Requires access to PM and DM relative to WM.

Main computational expense is accessing PM/DM

2. Short-term Metadata:
   - ΔWM → ΔWM*
   - Update accessed WM metadata

Avoid additional PM/DM accesses; use accessed

3. Learning: Changes to Long-term Memory
   - ΔWM; WM; WM*; DM; DM* → ΔDM*, ΔDM, ΔPM*, ΔPM
   - Update metadata based on data and metadata accessed by task performance.
   - Create structures based on data and metadata accessed by task performance.

4. Metadata Decay and Forgetting:
   - WM* → ΔWM*; WM* → ΔWM; ΔPM* → -PM; ΔDM* → -DM
   - Local update of WM, PM, DM metadata

Limits of Pure Architectural Learning

1. No “deliberate” learning
   - Can’t decide to learn something.
   - Suggests need for <u>meta-cognitive control.</u>

2. Constrained to incremental, constant-time learning algorithms
   - Suggests need for <u>unconstrained learning analysis.</u>

3. Learns from data recently *accessed* by task processing.
   - Limited temporal horizon.
   - Difficult to quickly learn temporal regularities across longer time scales.
   - Suggests need for <u>access to DM during learning.</u>

   - Difficult to learn *implications* of current knowledge.
   - Suggests need for <u>access to PM during learning for planning/etc.</u>

Hypothesis: Second Level of Learning

Knowledge-based strategies

- Occurs during task slack time (but interruptible)
  - Another task the agent pursues
  - Finesses real time, bounded computation issues.
- Metacognitive reasoning across multiple cycles
  - Accesses DM, PM, M to bring in temporally distant data of L1.
  - Creates “experiences” for L1 mechanisms to learn from (no new mechanisms)
    - Create historical, hypothetical/counterfactual experience from memory
    - Create novel experiences through environmental exploration
    - …
- Allows unconstrained processing of experiences when there was insufficient time during task performance.

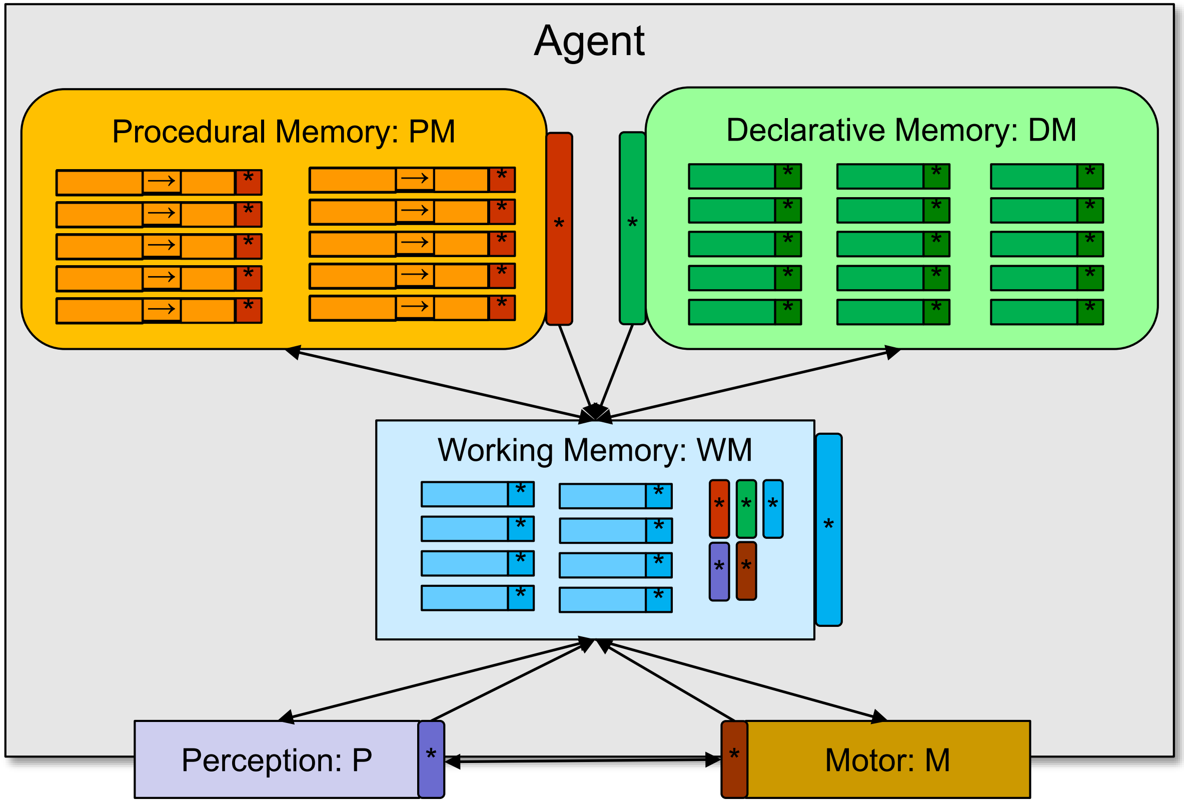
Contrasts between L1 and L2

- L1 continual architectural learning
  - Fast, bounded processing with constrained task knowledge and task metadata all the time.

- L2 intermittent learning strategy
  - Arbitrary processing with arbitrary task knowledge at limited times.

- What about metadata for L2 learning?
  1. Episodic memory
  2. Deliberate maintenance of metadata in semantic memory
  3. Architectural process data that is made explicit (in WM).
     - Failures to retrieve data from long-term memories
     - Failures in perception and motor
     - Innate appraisals: surprise, goal failure, loss of control, …

Architectural Process Metadata



Available in working memory for metacognition, such as reasoning about failure.

Available to architecture for learning: episodic memory and RL.

Procedural Memory Process Metadata [PMP*]:
- Retrieval failure (impasse in Soar)

Declarative Memory Process Metadata [DMP*]:
- Retrieval failure

Working Memory Process Metadata [WMP*]:
- Surprise and other appraisals

Perception Process Metadata [PP*]
- Failure, …

Motor Process Metadata [MP*]
- Completion, fault, …

Summary and Conclusion

- Analysis of how learning can be integrated with performance for autonomous agents under time constraints
- Analysis of L1 learning in Common Model
  - Identify importance of metadata
  - Identify shortcomings that lead to necessity of L2
  - Identify need for metadata access in L2
- Next talk shows how L1 and L2 mechanisms support Interactive Task Learning
- Still long way to go to completely understand this.

List of Publications, Awards, Honors, etc. Attributed to the Grant

- Mohan, S., Klenk, M., Shreve, M., Evans, K., Ang, A., and Maxwell, J. (2020). Characterizing an Analogical Concept Memory for Newellian Cognitive Architectures, *Advances in Cognitive Systems.*
- Laird, J. E. (2020). Intelligence, Knowledge & Human-like Intelligence, *Journal of Artificial General Intelligence 11(2), 41-44. doi:10.2478/jagi-2020-0003.*
- Laird, J. E. (2019). Introduction to *Sciences of the Artificial, 4th Edition, Herbert A. Simon, MIT Press. - Not attributed*
- Laird, J. E., & Mohan, S. (2018). Learning Fast and Slow: Levels of Learning in General Autonomous Intelligent Agents *, National Conference on Artificial Intelligence, AAAI-2018. Senior Track, Winner of Blue Sky Award.*
- John E. Laird was co-winner (with Paul S. Rosenbloom) of the Herbert A. Simon Award for Cognitive Systems.
- John E. Laird was a member of the Games, Exercises, Modeling and Simulation (GEMS) Defense Science Board (2018-2019).
- John E. Laird is co-organizing a DoD (Basic Research Office - OSD), Future Directions Workshop on the Design of General, Integrated Artificial Systems, July 2019.
- John E. Laird organized the first Virtual International Conference on Cognitive Architecture, June 2020.