Soar-inspired Agentic LLM Capabilities: Hierarchical Decomposition & Knowledge Compilation

Center for Integrated Cognition

Robert Wray, James Kirk, John Laird

45th Soar Workshop

5 May 2025

Long-term Goal: Rapid, Robust, General Task Learning

- General agents need to perform a lot of tasks well (efficiently, robustly, etc.)
- Training/teaching a general agent to perform these tasks is iterative and time-intensive
- LLM agents offer potential to both **increase generality** and **decrease cost** of training/instruction
  - Exploit existing NL artifacts (written documentation) to bootstrap learning
  - Speed/lessen friction in interpreting instruction (less brittle human/agent interactions)
  - Leverage huge reservoir of “priors” for many tasks?
- Two primary limitations:
  - Hallucination, tendency to prefer priors
  - Limited ability to retain instruction (persistent change)

Long-term Goal: Rapid, Robust, General Task Learning

- General agents need to perform a lot of tasks well (efficiently, robustly, etc.)
- Training/teaching a general agent to perform these tasks is iterative and time-intensive
- LLM agents offer potential to both **increase generality** and **decrease cost** of training/instruction
  - Exploit existing NL artifacts (written documentation) to bootstrap learning
  - Speed/lessen friction in interpreting instruction (less brittle human/agent interactions)
  - Leverage huge reservoir of “priors” for many tasks?
- Two primary limitations:
  - Hallucination, tendency to prefer priors → Human supervisory role
  - Limited ability to retain instruction (persistent change) → Limited scaling (always supervised)

Strategies for Realizing General Agents

Alternatives for Pursue General Agent Capabilities (like Instructability)

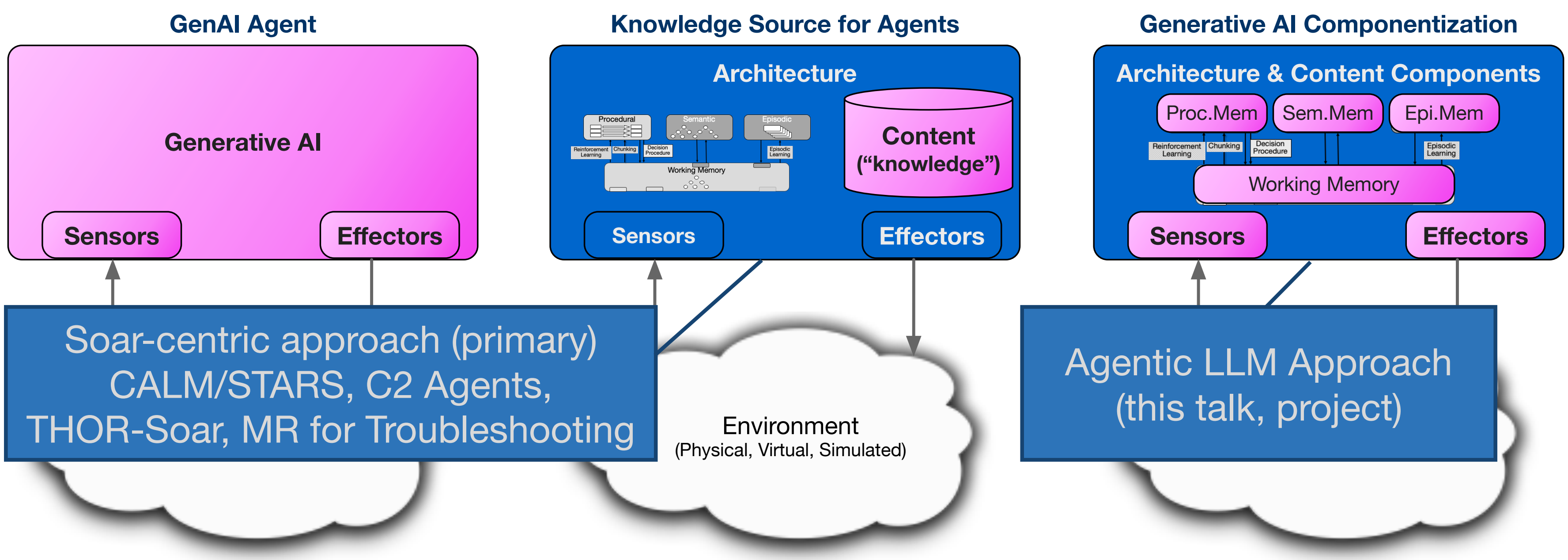
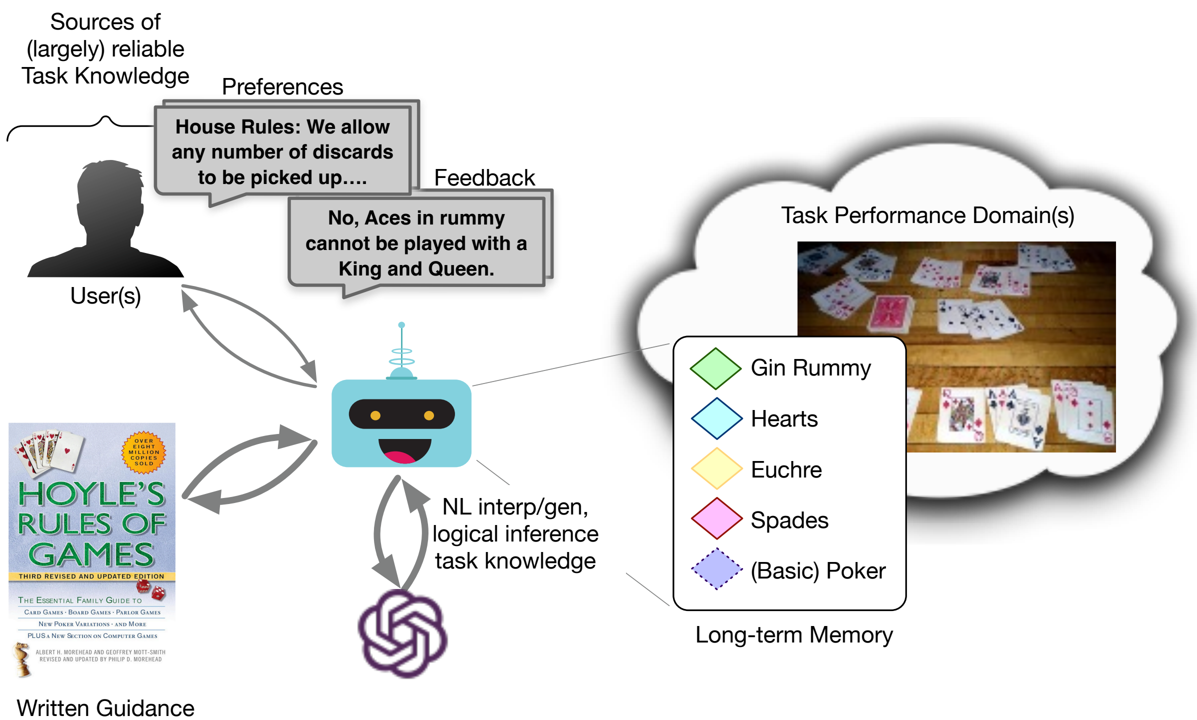


Illustration: Learning to Play Cards



Research Context

Cognitive design pattern: Abstract specification of a functional role within an integrated cognitive system (see pvs talk for example and details…)

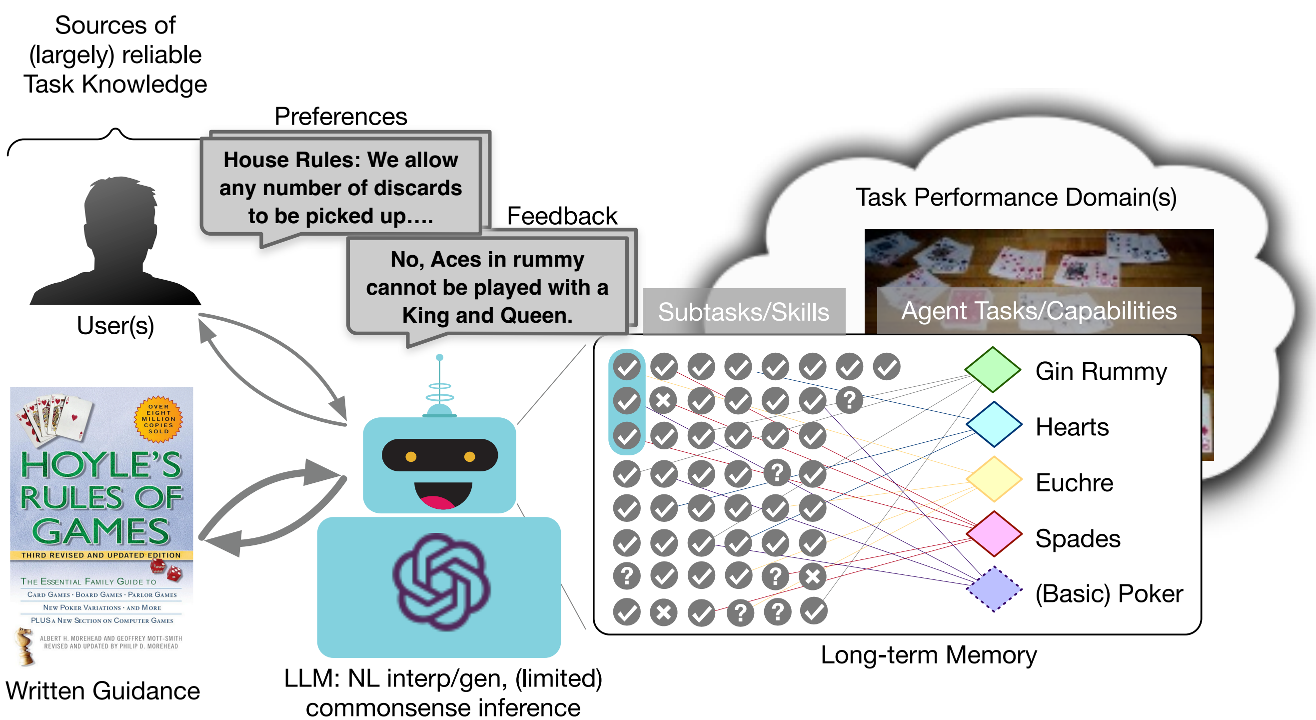
- Many "cognitive design patterns" are being reinvented by LLM Agent researchers
- Oftentimes these reinventions are incomplete or incoherent

CIC "meta" hypothesis: Designing LLM agents with relevant cognitive design patterns will lead to more fine-grained, more controlled "thought generation" in LLMs
→ more reliable LLM reasoning (improved quality and precision)

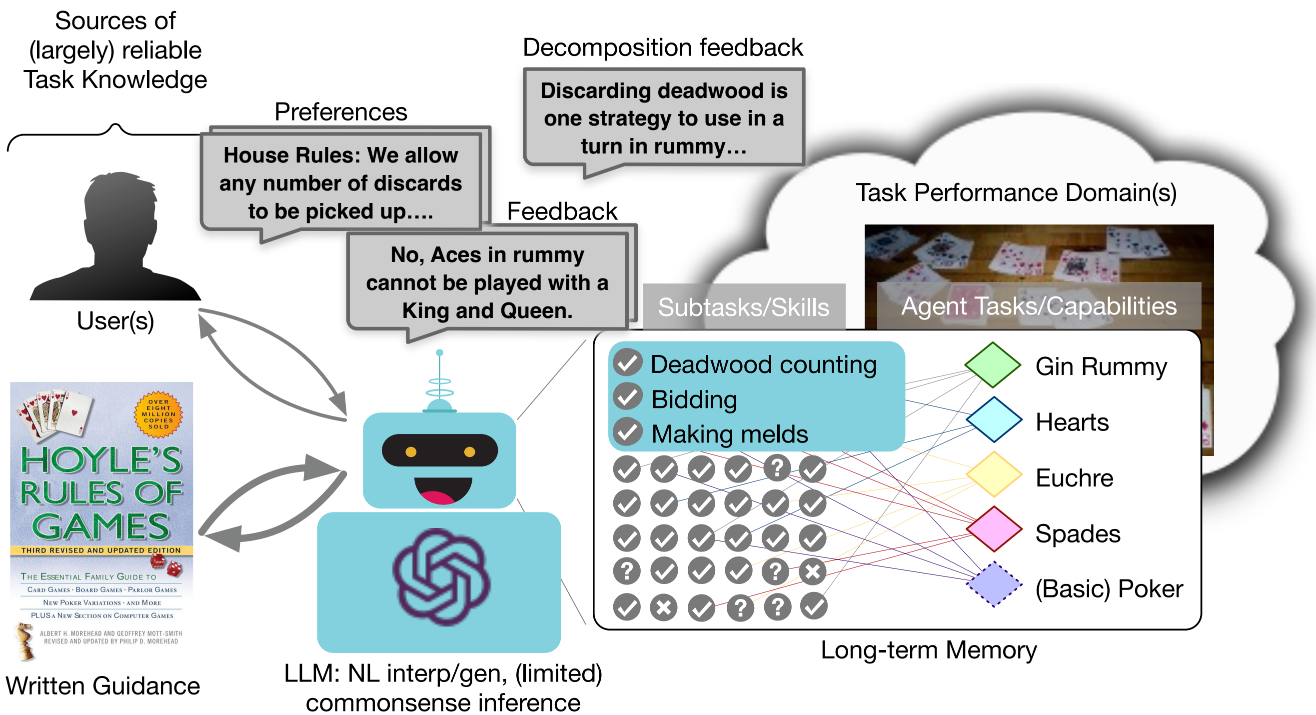
Project Concept

- Choose cognitive design patterns particularly relevant to task learning
  - Hierarchical task (subgoal/subtask) decomposition
  - Compilation of reasoning
  - Synergistic effects: Decomposition → small, bounded segments of reasoning → compilation → reuse of small bits in many different contexts (transfer)
- Goal: Prototype LLM agent system that exhibits subtask decomposition and compilation
- Rough Hypotheses:
  - Significantly enhance task reliability over other approaches (e.g., o1/reasoning models)
    - New instances of a known task draw on "validated" subskills
  - Significantly reduce inference cost over o1 agent (many tasks)
    - New instances of a known task draws on local "knowledge," rather than the LLM
  - Observe non-trivial transfer from one task domain to another
    - New instances of unknown tasks share "subskills" with other tasks, enabling transfer of subskills

Instructable Agents (Vision)

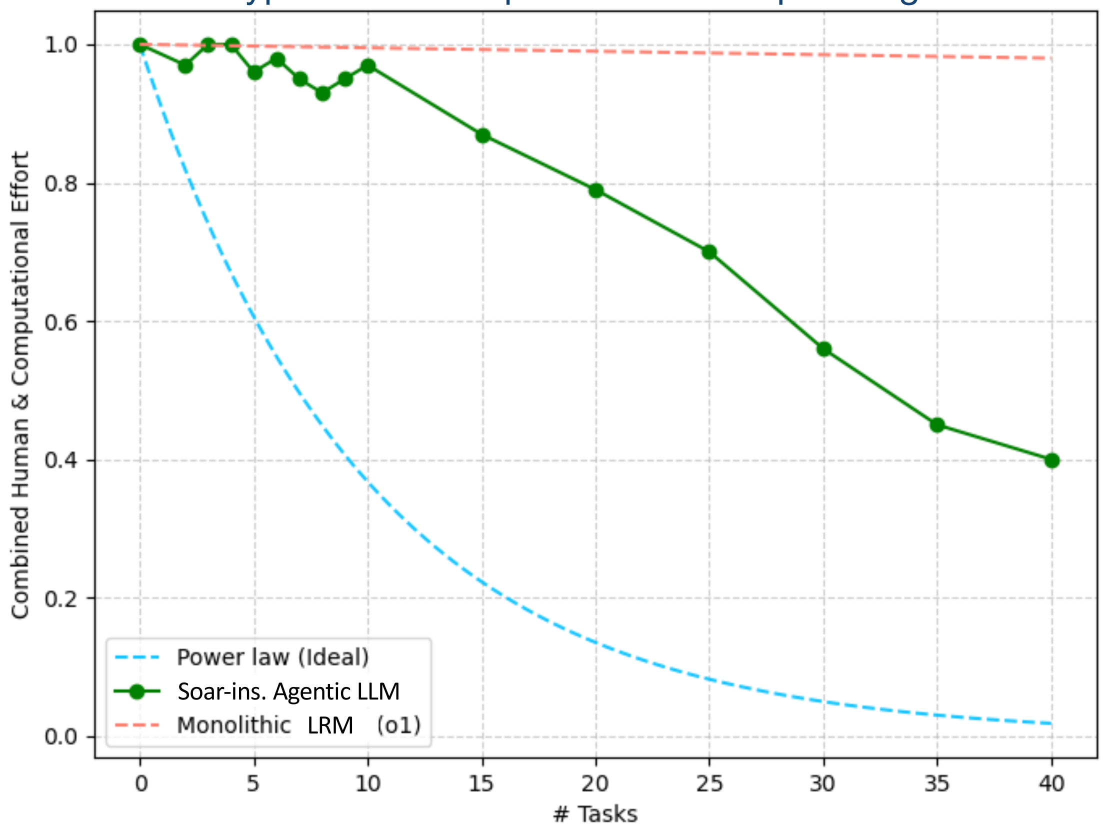


Instructable Agents (Vision)



Potential Impact/Computational Benefit

Illustration of hypothesized impact of Soar-inspired Agentic LLM



- Ideal case: Increasingly minor effort needed to learn new tasks
  - Human effort
  - Computational effort (e.g., run-time token use)
- Expectation for large reasoning models (LRMs): Little/no change as new tasks are introduced
- Hypothesized impact of our approach: Eventually, small marginal effort for new task
  - May see little benefit initially as new tasks are introduced

Hierarchical Task Decomposition

Pattern:

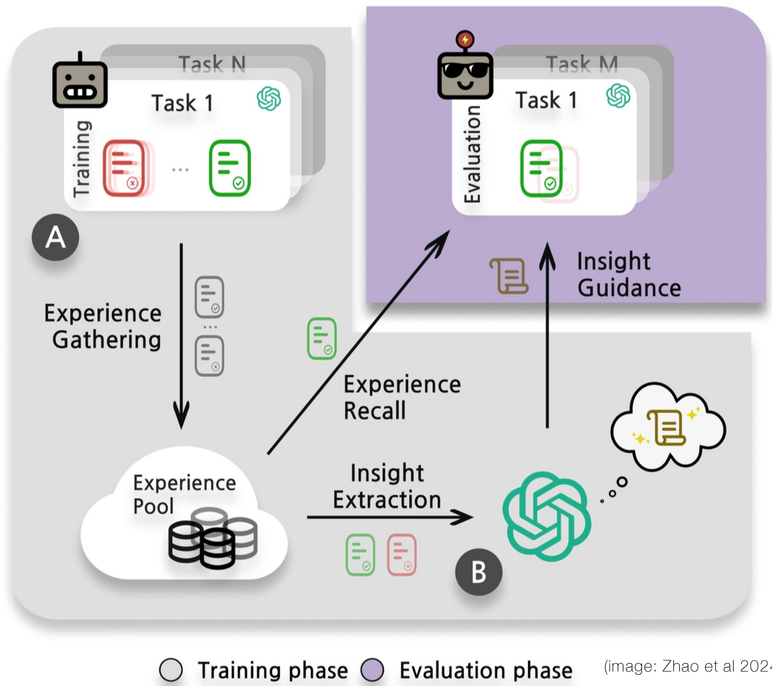
- **Task Decomposition**: Break down a task/goal into a set of (simpler) tasks/goals
- **Hierarchical**: Recursive
- Key feature: Adaptivity
  - Specific run-time situation influences the "paths" chosen to perform the task
- Examples:
  - Soar: Operator subgoaling
  - Plan execution agents: HTNs
- Can LLMs perform hierarchical task decomposition?
  - Many examples (ADAPT, DeAR, TaskLAMA, …)
  - Generally, limited/shallow decomposition
  - Some examples are online/adaptive, most are not (often applied to fixed benchmarks)
  - *Opportunity*: Explore methods inspired by operator subgoaling that can support deeper and more adaptive/dynamic decomposition (e.g., local memory for subgoals)

Knowledge Compilation

- Many tasks and problems require various kinds of deliberation
  - Example: Planning
  - Deliberation is resource intensive; requires time, attention
- Knowledge compilation: Save/cache results of deliberation
  - Generalization enables application in comparable situations
  - Common processing pattern in cognitive architectures and planning (Rosenbloom & Newell, 1986) (Anderson, 1986)
- Can LLMs support/realize knowledge compilation?
  - Very clear examples of compilation to formal representations
    - Voyager, compilation to Soar rules (Zhu & Simmons), many other examples from program synthesis
  - ExpeL offers one example of something akin to compilation of reasoning to a rule-like procedural knowledge store …

Compilation in ExpeL

1. Gather Experiences (attempt tasks)
2. Derive "insights": Induce general rules about domain from problem solving traces
3. Apply insights: Given a new problem, decide what rules are apt and apply them



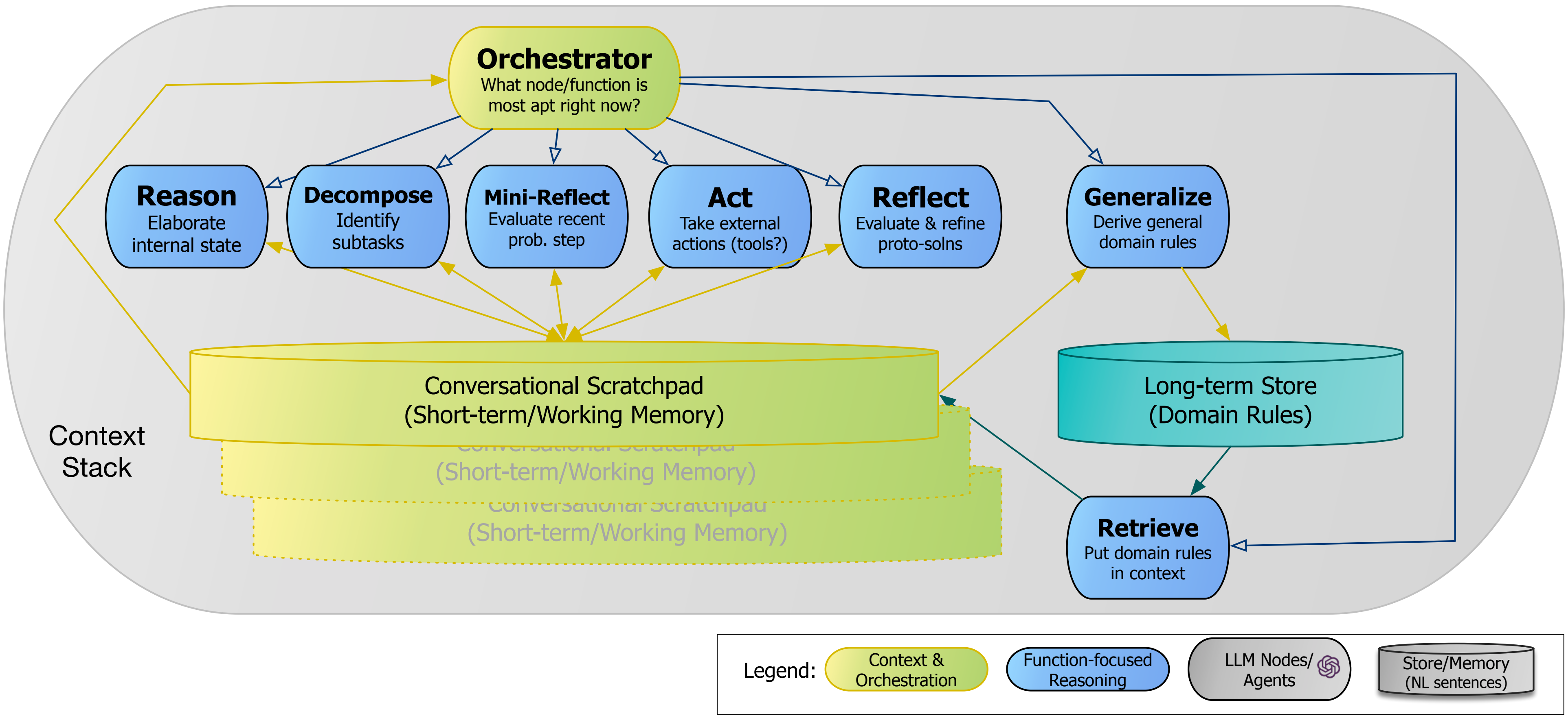
(image: Zhao et al 2024)

Example insights (ALFWorld):

3. If an item is not found in one location, systematically check the next likely location based on the nature of the item and common household organization, without assuming its presence.

4. Prioritize checking locations that are most likely to contain the item you are looking for, based on the nature of the item, common household organization, and the task at hand.

Strawman Target Architecture



Conclusions

Nuggets

- Opportunity to put some the cognitive design patterns concepts into practice.
- Further opportunity to explore instruction/teachability (key missing feature of intelligent systems).
- Going directly after a key limitation of LLMs: lack of persistent adaptation without re-training/fine-tuning: **high payoff**.

Coal

- Early in the effort, so results TBD.
- Outcomes may depend (a lot) on choice of domain(s) but the key properties of domains are difficult to determine in advance.
- Going directly after a key limitation of LLMs: lack of persistent adaptation without re-training/fine-tuning: **high risk**.