

Final Report: Adapting Memory for Long-Duration Autonomy

MARK ROBERTS

Email: mark.roberts@nrl.navy.mil

Navy Center for Applied Research in AI (NCARAI)
Information Technology Division

Keywords: Artificial Intelligence, Autonomous Systems, Cognitive Priming, Perpetual Learning

10 September, 2021

This **NCARAI Technical Note** is the final project report for the NRL Base Program project “Adapting Memory for Long-Duration Autonomy”, funded FY19-FY21 in the IT Focus Area.

Technical Objective

Our objective is to demonstrate that goal priming and perpetual learning are essential as a system's lifetime and capability increase. Our planned contributions include: (1) a theory of goal priming that combines previous work in goal reasoning and cognitive priming; (2) extending previous work on perpetual learning to long-duration systems; (3) formal and empirical analyses showing how goal priming and perpetual learning combine to support increasingly more capable, long-term autonomy; and (4) demonstrating a system running for 18 continuous months that consolidates experience and retrieves relevant goals based on context in scenarios from two simulated environments: logistics delivery for Foreign Disaster Relief, a Navy-relevant domain, and Minecraft, a challenging domain from Microsoft Research.

Technical Approach

The key innovation of this proposal, the pending memory, extends work developed at NRL on goal reasoning, cognitive priming, and perpetual learning to help a system remain reactive while learning. The pending memory consists of a priming queue and a learning agenda, which will (1) for retrieval, speed up subsequent retrieval by storing recently primed strategies in the priming queue; (2) for learning, prioritize potential learning opportunities in a learning agenda, and, (3) for both, facilitate associations between the environment and strategies in long-term memory.

The priming queue serves as a layered storage memory for retrieving context-relevant goal strategies from the long-term memory. Goal strategies are built from goal reasoning. In goal reasoning, as formalized by PI Roberts (and others), a system's goals are tracked in a goal memory and modified by a set of goal strategies, which control how a system creates, selects, plans for, drops, or amends its own goals. Ideally, a system would leverage its experience to retrieve the best goal strategies based on context while directing its own learning agenda to improve future performance.

Strategies will be retrieved into the priming queue via goal priming, a mechanism that blends goal reasoning with cognitive priming. Priming learns associations between related items in long-term memory, allowing the current context in the working memory to prime, and thus retrieve, related items from longterm memory. Co-PI Hiatt (with others) showed cognitive priming to be a fundamental component in higher-level cognitive processes. The PI Roberts and Co-PI Hiatt recently published results showing that an offline version of goal priming performs similarly to decision tree induction when learning to select subgoals in a maze and that priming can improve online learning for determining the best tool to perform a job when trained with textual information. Together, these results provide evidence that goal priming will be an effective retrieval mechanism for our system.

The learning agenda allows a system to selectively focus its learning effort and will extend our ongoing work in perpetual learning. Our proposed system will use the learning agenda to prioritize the possible learning tasks of the system as a set of learning goals. Formulating these learning choices as goals means that we can leverage and extend existing work in goal reasoning, which already provides mechanisms for deciding which goals are most critical for the system. With respect to learning goals, if the system already has effective strategies for a task, it need not

learn (i.e., it does not post a learning goal). If the system only knows a little, it can post a learning goal to invest more learning effort the next time it encounters this task.

If the system is continually failing to meet its expected or desired performance on a task, it can post a learning goal to explain why and set up a better opportunity for learning, perhaps through practice. If the system is oversubscribed, it can create a goal to revisit its history later when more resources are available. Learning goals focus learning on improving performance, reducing the overall memory footprint and maintaining reactivity.

The combination of the priming queue and the learning agenda in the pending memory will further contribute to improved performance. Recently primed strategies in the priming queue enable instantaneous future retrieval. As these primed strategies in the queue relate to recent experience, such as during early learning of a new task, the pending memory catalyzes learning since the very strategies that may need improvement are already in the queue. As a system focuses on learning goals, goal priming can further improve learning by speeding up retrieval from long-term memory. Over time, we expect a system using the pending queue to shift from learning to fast retrieval. As a system becomes acquainted with its current environment, learning will decrease and retrieval will be more effective, resulting in decreased computational effort and a lower memory footprint.

Implementation. We will implement our system in ActorSim, which is a general platform for conducting studies of autonomy in simulated environments. ActorSim was developed for the study of goal reasoning and includes a goal reasoning module as well as a prototype of perpetual learning. It represents several years of effort by PI Roberts and others in NRL Code 5510 in building a general purpose system for studying autonomy. ActorSim already has baseline implementations for our simulated environments of disaster relief and Minecraft. We will implement an online version of goal priming and test it against baseline systems and existing scenarios to demonstrate our objective. Tasks will vary for each environment. We will evaluate the system on maintaining reactivity and choosing appropriate actions in increasingly challenging environments, which we will measure as fewer errors and improved utility over time.

Evaluation. We will evaluate our system along three axes: (1) Under what conditions does the pending memory improve an agent's adaptation to a new task? (2) Under what conditions does the pending memory help an agent maintain responsiveness in the face of an increasingly complex environment? (3) Under what conditions does the pending memory improve long-term performance over a system that lacks the memory?

For each of these questions, we will generate specific research hypotheses and then test them in controlled scenarios that vary independent variables to observe the effect on dependent variables. Independent variables will include: the environment and its complexity, the learning mechanism, and the model complexity (i.e., provided, learned, or ablated), while dependent variables (i.e., the metrics) will include: cumulative gain (e.g., game score, resource count), length of life, learning rate, and responsiveness to teammates. We will aim to test descriptive, proscriptive, and predictive models.

Technical Progress

Our FY19 effort has primarily focused on extending ActorSim and simulators with the necessary support for long-duration autonomy. This has focused on extending the simulators for the project, designing and implementing a cognitive cycle and pending memory, and extending results for cognitive priming.

The project is using two existing simulation platforms: RoboRescue and Minecraft Malmo. Our initial effort involved extending these platforms to support many tasks being executed for long-running simulations. In the Malmo simulator, we developed an agent that can perform a dozen or so goals for simulations lasting a full day of the agents life (about 10 minutes of wall clock time). The agent is able to explore for and harvest simple resources and craft simple items that it can use. In the RoboRescue domain, we crafted 7 goal types that can be used for specific agents. In both simulators, we created about 10 scenarios that we are using to test the system as we extend it. The bulk of our development effort has been to develop the primary mechanism for supporting a cognitive cycle in ActorSim. The previous implementation supported an event-based response cycle with goal strategies that were tightly coupled to goals. This design was simple for short-lived agents and simple experiments but insufficient for integrating goal priming and long-term memory. Thus, we designed and implemented a cognitive cycle for ActorSim. The cognitive cycle performs 5 steps each time it runs: (1) update beliefs from perception, (2) determine applicable strategies, (3) rank applicable strategies, (4) filter strategies, (5) apply strategies in order of ranking. We redesigned and implemented revised goal strategy interfaces to support storage across multiple memories and this new cognitive cycle. We have also updated the memory mechanism for ActorSim to work with a database backend. ActorSim now connects to a NoSQL database called MongoDB to store its changes in beliefs. We have demonstrated the MongoDB backend for the Malmo simulator.

Finally, we have continued our efforts on cognitive priming. One of the key parts of this project is using cognitive priming to achieve the right balance between learning and retrieving. If not much is learned, retrieval is easy, but adaptability and capability can be decreased; if a lot is learned, adaptability can be higher, but retrieval is much more difficult, again negatively affecting capability. In order to address this, we are examining a phenomenon in people called “Retrieval Induced Forgetting” [C1, A1]. In this paradigm, repeated exposure to some concepts increases forgetting of other, related concepts. We have built a model of retrieval induced forgetting that shows that cognitive priming is a large driver in the forgetting effect. This begins to indicate how we can use priming to manage learning and retrieving. We are currently working to capture this effect with the much larger databases of knowledge that we will on long deployments of autonomous systems.

We also extended our understanding of how to develop hierarchical models for agents that can recognize the goals of another agent [C2, W1, W2]. Goal recognition is an inverse problem to planning and will be essential to understanding when to use a particular solution from experience. Finally, we

Our FY20 efforts included (1) creating long-duration scenarios that support our envisioned experiments, (2) extending the planning in ActorSim to support Temporal Goal Networks (TGNs), a novel form of goal networks that support temporal reasoning over longer periods of time, resulting in two more software releases [S2, S3], (3) spinning off a 6.1 ONR project with a local university that will complement this project with a computational theory and algorithms for TGNs, and (4) identifying two possible transitions paths for the TGN work.

We substantially extended our two simulation platforms, RoboRescue and Malmo. RoboRescue updates and modifications included completing the set of goal strategies for all RR agents. We have implemented 72 goal strategies that allow agents to douse fires, remove road blockages, unbury injured humans, and transport humans to shelters. Updates and extensions to the Malmo Controller, originally from Microsoft Research, will support studies in long-duration autonomy and include generating 40 scenarios (complete worlds instead of synthetic) that extend our scenarios from static-and-closed to dynamic-and-contested environments. We also included problem features that are required to succeed in the environment. We created and prioritized a list of necessary skills for performing increasingly complicated tasks and have extended the character with digging, lighting its area, and interacting with containers. Limitations in the information coming from the simulation platform have delayed our progress. However, we have identified the changes necessary to ameliorate this issue and are currently implementing this change.

Long-running agents must reason about future resources and events. Suppose an agent has a goal to construct a complicated tool several days from now. It must both acquire and reserve those resources for the duration of the goal so the resources are not inadvertently used on a less valuable goal. Few planners support this kind of long-range reasoning, which led to developing a planning technique we call Temporal Goal Networks (TGNs), a key conceptual outcome from this year. We substantially revised the Core library in ActorSim to support TGNs and integrated an existing temporal planner to provide the initial temporal plans we need to generate TGNs. Our draft implementation of TGNs led to identifying several limitations of existing approaches.

Based on our increased understanding of the current limitations, we proposed a 6.1 research project to ONR's Science of AI program, which resulted in a joint \$1.75M, 3 year project with Univ. of Maryland Prof. Dana Nau entitled Learning Online Temporal Goal Networks. In that project, the UMD team will develop a computational theory and algorithms for solving TGNs, formalizing the TGN implementation we developed for this, and other, NRL projects. In future years, that project will shift to creating algorithms for learning TGNs from execution traces and transferring TGNs across increasingly challenging scenarios, which will build on the scenarios created for this project. The two projects complement each other well since UMD can focus on developing the underlying theory and algorithms, while NRL researchers can focus on transitioning these developments into Navy-relevant problems.

COVID impact: Two interns (one HBCU, one SSEP) were unable to join this project in FY20 due to issues over working remotely. Another two interns were supposed to work on extending the skills of the Malmo agent but we could not install our software on their laptops (NRL was

unable to provide machines for their use). So the interns are working on another project. Travel restrictions have limited dissemination of our work to the wider community.

Our FY21 efforts further extended how we represent the short-term memory of an agent to support short-term and long-term reasoning. We further developed Temporal Goal Networks and integrated Reinforcement Learning (RL) controllers with goal networks in service of perpetual learning. We revised several aspects of how ActorSim represents goal networks and introduced the Goal Lifecycle Network [C3], resulting in a final software release for the project [S4] as well as several invited talks to share our recent research [T1, T2, T3]. We extended our knowledge of how to perform plan recognition under various assumptions [W3, A2]. Finally, we overcame a serious flaw in the controller for the Malmo agent, which has improved how ActorSim will integrate with that agent. We are on target for achieving our primary milestone of demonstrating a simulated agent interacting in a world for 18 months of simulated time.

One of our main accomplishments was collaborating with the University of Maryland during their development of a new planner called Goal-Task Pyhop, or GTPyhop [W5] and its use for replanning [W6]. GTPyhop extends a former Hierarchical Task Network planner released by the University of Maryland, Pyhop, with the ability to plan over goal networks. We also collaborated with researchers from the University of Strathclyde during their development of a new simulation platform for planning and acting [W4].

In terms of extending our simulation efforts, this year we focused our efforts solely on the Malmo platform. Early in the year we identified a critical challenge with the current simulation platform (Malmo) that caused it to perform slowly. We investigated this and overcame it by shifting to a new controller called Baritone. Although this did set us back a few months of development, it should provide a much stronger base for a long-duration simulation going forward.

There were several extensions to the ActorSim framework. First, we developed a unique kind of goal network called a Goal Lifecycle Network [C3] that enables a goal network to track the status of goals as they progress over time. This was a major step in the implementation toward Temporal Goal Networks. Second, we enabled access to the state of the goal memory in ActorSim through a connection to an in-memory document database called Memcached. This allows external clients to direct and respond to ActorSim through simple text-based documents over a network connection. Finally, we extended the work began by the 2020 summer interns on creating a GUI for the goal memory. The GUI shows the current state of the agent's memory as well as the status of which goals it is pursuing. The GUI is written as a responsive Web Application using Spring, JSON, and HTML5.

In terms of goal priming, we began a new collaboration with Assistant Professor Rogelio Cardona-Rivera to explore how Computational Narrative can improve the recall and precision of selecting the best goal decompositions. This work is funded by external ONR funding that was the result of our collaboration with the University of Maryland.

Naval / Marine Corp Needs

Naval R&D Framework Priority: Augmented Warfighter: Focus on human & machine teaming.

Integrated Research Portfolio: Information, Cyber, and Spectrum Superiority: Focus on adapting memory & knowledge.

Enduring Research Responsibility: Machine Learning, Reasoning, and Intelligence: Focus on learning and reasoning in long-duration autonomy.

Key Words

Artificial Intelligence, Autonomous Systems, Cognitive Priming, Perpetual Learning

Transition Plan

In FY20, Jampathom, Roberts, and Shetty proposed a study for UUVs called Aquarius to ONR-32, who is providing seedling funding (\$340,000 / 6 months), for a study of collaborative mission planning for VMOC-UUV-spacecraft. One outcome of this project will be the development of a demo scenario using a simple UxS platform, called CrazySwarm, as a tool to support cross-functional training between VMOC developers and the AI Center. Part of the effort is to work with the sponsor to identify possible transitions to meet ONR ongoing needs. This effort was extended in FY21 to include risk-based communications planning between UUVs and satellites, which has resulted in a university contract with Colorado State University to develop a risk-based planner that we plan to integrate with VMOC in FY22.

Transitions

Our primary transition has been to incorporate Temporal Goal Networks into NRL's Virtual Mission Operations Center (VMOC) system through funding mentioned above.

Publications and Patents (Chronologically ordered)

Summary (brackets before each reference indicates type and number of item):

- 2 Refereed Journal or Magazine Articles (A)
- 3 Refereed Conference Papers (C)
- 4 Software releases (S)
- 3 Invited Talks (T)
- 2 Edited Proceedings (P)
- 6 Refereed Workshop (W)

FY19:

[S1] Mark Roberts. ActorSim Version 3.0. Open source software release. October, 2019.

FY20:

[C1] Laura M. Hiatt, Stephen J. Jones. An associative learning account for retrieval-induced forgetting. In *Conference on Cognitive Science*. July, 2020.

[S2] Mark Roberts. ActorSim Version 2020.05. Open source software release. May, 2020.

[A1] Lawless, W. F., Mittu, R., Sofge, D., and Hiatt, L. Artificial intelligence, autonomy, and human-machine teams — interdependence, context, and explainable AI. In *AI Magazine*. July, 2019.

[C2] I. Rabkina, P. Kathnaraju, M. Roberts, J. Wilson, K. Forbus, and L. M. Hiatt. Recognizing the goals of uninspectable agents, In *Proceedings of Eighth Conf. on Advances in Cognitive Systems*, Held Virtually: Cognitive Systems Press, 2020.

[W1] P. Kantaraju, S. Ontanon, C. Geib, and M. Roberts. Combinatory Categorical Grammar Learning for Plan Recognition in Domains with Type Trees. In *Working notes of the PAIR-2020 Workshop held at AAIL-2020*. February, 2020.

[W2] Rabkina, Irina and Kathnaraju, Pavan and Wilson, Jason and Roberts, Mark and Hiatt, Laura. Recognizing the Goals of Uninspectable Agents. In *Working notes of the PAIR-2020 Workshop held at AAIL-2020*. February, 2020.

[S3] Mark Roberts. ActorSim 2020.08. Open source software release. August, 2020.

FY21:

[P1] Z. Alhavi, M. Roberts, T. Vaquero, eds. Proc. of the Combined Workshop on Integrated Execution and Goal Reasoning. In: Workshop Program at ICAPS 2020. October, 2020.

- [T1] Roberts. Temporal Goal Networks for Collaborative Autonomy. In *Army Research Laboratory Research Roundtable*, Dec. 2020. Invited talk.
- [T2] Roberts. Temporal goal networks: A research agenda. In *Autonomy Community of Interest: Machine, Perception, Reasoning, Intelligence Group*, January. 2021. Invited talk.
- [W3] Rabkina, Irina and Kantharaju, Pavan and Wilson, Jason and Roberts, Mark and Hiatt, Laura M. Comparing Hierarchical Goal Recognition via HTN, CCG, and Analogy. In *Working notes of the PAIR-2021 Workshop held at AAI-2021*. February, 2021.
- [C3] Roberts, L.M Hiatt, V. Shetty, B. Brumback, B. Enochs, P. Jampathom. Goal Lifecycle Networks for Robotics. In *Proc. Int'l Florida AI Research Society (FLAIRS)*, May 2021.
- [T3] Roberts. Goal Provenance for Collaborative Autonomy. In *NATO-SCI 335 RSM Autonomy from a Systems Perspective*, May. 2021. Invited talk.
- [P2] S. Patra, M. Roberts, T. Vaquero, & W. Piotrowski, eds. Proc. of the Workshop on Integrate Execution and Planning. In: *Workshop Program at ICAPS 2021*. August, 2021.
- [W4] Liudvikas Nemiro, Gerard Canal, Oscar Lima, Michael Cashmore, and Mark Roberts. Designing an Adaptable Benchmark and Competition Simulation for Integrated Planning and Execution. In *Working Notes of the ICAPS-2021 Workshop on the International Planning Competition*. August, 2021.
- [W5] Dana Nau, Yash Bansod, Sunandita Patra, Mark Roberts Ruoxi Li. GTPyhop: A Hierarchical Goal+Task Planner Implemented in Python. In *Working Notes of the ICAPS-2021 Workshop on Hierarchical Planning*. August, 2021.
- [W6] Yash Bansod, Dana Nau¹, Sunandita Patra, & Mark Roberts. Integrating Planning and Acting by Using a Re-Entrant HTN Planner. In *Working Notes of the ICAPS-2021 Workshop on Hierarchical Planning*. August, 2021.
- [S4] M. Roberts. ActorSim 2021.09. (Anticipated)
- [A2] Rabkina, Irina and Kantharaju, Pavan and Wilson, Jason and Roberts, Mark and Hiatt, Laura M. Evaluation of Goal Recognition Systems On Unreliable Data and Uninspectable Agents. Under review.