Self-adaptive Mission Planning using High-Fidelity Open World Simulation

Wiktor Piotrowski[1], **James Chao**[2], **Sachin Grover**[1], **Shiwali Mohan**[1], **Douglas S. Lange** [2]

[1] PARC, a part of SRI International, CA, USA
[2] NIWC, San Diego, CA, USA
{wiktor.piotrowski,sachin.grover,shiwali.mohan}@sri.com, {james.chao.civ douglas.s.lange2.civ}@us.navy.mil

Abstract

AI and ML agents are developed with closed world assumptions, that can change during execution. This demo paper presents HYDRA, a framework for developing self-adaptive autonomous agents capable of handling unexpected domain shifts (also called *novelty*) during execution, applied to a high fidelity simulator for military mission planning. The framework is divided into a base agent, responsible for basic predict-decide-act cycle, and novelty monitoring to detect, characterize and adapt to the novelty. AFSIM is a high-fidelity mission simulator that incorporates many real-world military models; and has been used for mission planning in several scenarios. This paper shows successful integration of HYDRA with AFSIM, and demonstrates HYDRA agents efficiently adapting to novelty in realistic simulated military scenarios. Demonstration of our system is available at https://tinyurl.com/wb3z2edv.

HYDRA

Artificial Intelligence (AI) and Machine Learning (ML) research on sequential decision-making usually relies on the assumption of a static world. That is, all relevant characteristics of the environment are known ahead of deployment, during agent design time and remain unchanged during runtime. However, the real world is open, evolving, and can change without any prior indication. Online adaptation and learning is a desirable property of any real-world AI system. Prominent model-free learning techniques such as reinforcement learning acquire knowledge in non-interpretable representations (as network weights, biases etc.), posing significant challenges in assessment, validation, and regulation of what has been learned. Our research explores open-world adaptation in model-based reasoning systems which encode knowledge explicitly. Explicit representation of acquired knowledge enables inspection of what has been learned, supporting assessment, validation, and regulation of the system's evolving behavior in an open world.

Novelty can be a new object, a new skill available to the agent, or significant shift in underlying distribution. For agent to function with such changes it should detect, accommodate and adapt it's internal models based on the changes in the environment. To meet this challenge we present HYDRA, a domain-independent architecture for implementing novelty-aware agents in complex, mixed discrete and continuous domains (Mohan et al. 2023; Piotrowski et al. 2023a,b; Stern et al. 2022; Piotrowski et al. 2021). The HYDRA architecture includes a *base agent* and *novelty meta-reasoning* components designed to detect novelties and adapt the base agent's behavior to them. A notional architecture is shown in Figure 1.

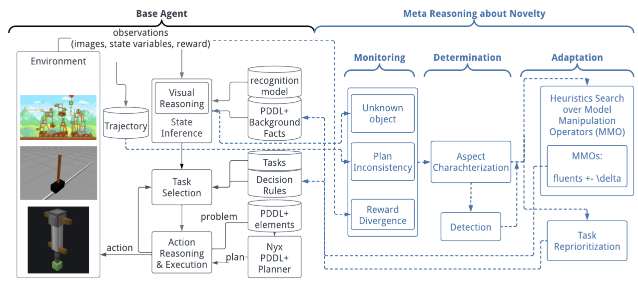

Figure 1: HYDRA: a self-adaptive architecture

The base agent (Figure 1 - left) implements a *perceive-decide-act* cycle in the environment. Explicit knowledge for the base agent is provided using PDDL+ (Fox and Long 2006), and we use our mixed-discrete PDDL+ planner called Nyx (Piotrowski and Perez 2024) to generate plans to be executed in the simulator. The novelty meta-reasoning components in HYDRA (Figure 1 - right) detect the presence of novelty, characterize it, and accommodate the changes by updating the agent's knowledge base. By default, novelty detection is done by measuring model inconsistency, that is the Euclidean distance between the planned state trajectory and the observed state evolution upon executing the plan in the simulator. More specifically, the distance is measured over a subset of planning state variables that are directly observable in the simulator (e.g., aircraft positions and velocities). Upon discovering novelty (when the measured inconsistency exceeds some predefined threshold), HYDRA engages model repair to update the agent's internal PDDL+ model such that it accurately reflects the post-novelty world. HYDRA's model repair method utilizes heuristic search, guided by inconsistency minimization, to compose candidate repairs which explain the discrepancy between the expected and observed outcomes. Candidate repairs are sequences of explicit atomic modifications to the PDDL+ model (also called model manipulation operators or MMOs). The repair process yields a modified PDDL+ model that the AI agent then uses to reason about its surroundings and plan its actions in

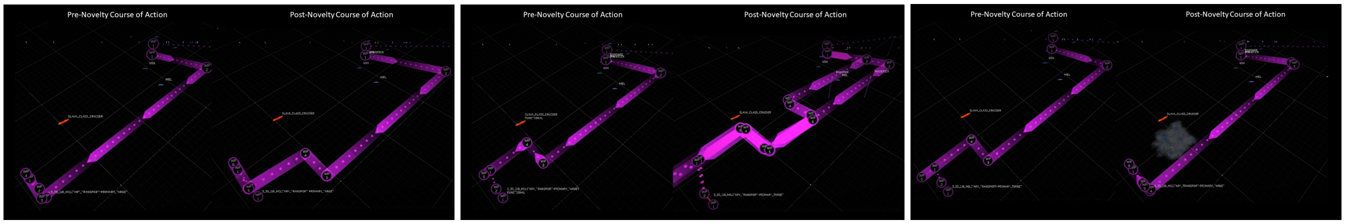

Figure 2: Pre- and post-novelty Courses of Action for each novelty scenario: Advanced SAM (left), Hardened Target (center), Inclement Weather (right).

the post-novelty world. The accepted model modifications are explicit and interpretable (e.g., $\{$`gravity: 10.0`$\}$ indicates that the repair has increased gravity by $10m/s^2$). Thus, HYDRA's repair mechanism simultaneously provides novelty characterization and accommodation.

AFSIM – High-Fidelity Simulator

AFSIM (Clive et al. 2015) provides a realistic simulation of behavior for the entities including F-35 fighter jets and surface-to-air missiles (SAM). The environment simulates real world environmental characteristics, including partial observability, stochasticity, multi-agent, dynamic, sequential, continuous, and asymmetric battles. To allow PDDL+ planning, required by the Hydra architecture, in the AFSIM environment, the environment is modified to resemble an OpenAI Gym interface (Brockman et al. 2016) through a Python framework, where real time is segmented into discrete steps, missions are segmented into episodes (battles in military terms), and multiple episodes are segmented into a tournament (campaign in military terms).

States, time, actions are all continuous variables in AFSIM. The continuous actions controlling air movement with high fidelity physics are taken in a continuous timeline. Time is continuous and there is no concept of a discrete time tick. Space is continuous and there is no concept of a discrete grid cell. However, the PDDL+ planner uses a discretization-based approach to solve planning tasks, discretizing it into a geo-spatial grid, time step, and discrete actions. The execution engine uses a low-level planner to translate the discrete variables used by the PDDL+ model back into continuous variables for the environment.

Experiment & Results

Our experimental setup focuses on U.S. Navy aircraft carrier group missions. In all examples, (blue) F-35 strike fighter aircraft launch from a carrier and attempt to eliminate an enemy (red) vessel using JDAMs. i.e, precision guidance missiles (Bell 2015), without suffering any (blue) casualties. The red target is protected by an anti-aircraft vessel equipped with surface-to-air missiles (SAMs) that can shoot down blue strike fighters. We present three distinct novelties that affect the mission execution such that the default plan/strategy cannot achieve mission objectives in the post-novelty world. Thus, the agent must understand the novelty and generate an alternative strategy to successfully complete the mission. The presented novelties are directly inspired by real military experiences in which unexpected events caused catastrophic mission failure.

Advanced SAM In this scenario, the enemy (red) anti-aircraft vessel is unexpectedly equipped with advanced *long-range* missiles, posing a direct threat to blue strike aircraft at a much greater distance than initially assumed. If taking the pre-novelty route, the blue strike fighters will be shot down.

Accommodation: After a single failed episode, HYDRA increases the assumed enemy missile range forcing the blue strike fighter aircraft to adjust its trajectory to avoid encroaching the increased SAM range.

Hardened Target In this scenario, the red target vessel cannot be destroyed with a single JDAM weapon. To complete the mission, the blue fighters must strike the target with a more powerful weapon or multiple JDAMs.

Accommodation: After an episode in which the red target vessel was damaged but not destroyed, HYDRA increases the target's survivability. The update forces multiple strikes on the red target. In subsequent episodes, two blue strike fighters are simultaneously sent out to engage the target and ensure its elimination.

Inclement Weather In this scenario, the blue team is unaware of the inclement weather en-route to the target and the mission is aborted once the agents reach the bad weather.

Accommodation: After aborting the first post-novelty mission, HYDRA updates the map with inclement weather regions as returned by the observation different and it finds a new route to the target. In the second attempt the blue team agents take the new route and successfully kill the target.

Conclusion

We described HYDRA, a framework for self-adaptive AI agents. HYDRA agents are designed to operate in open worlds whose core characteristics can unexpectedly shift during deployment (novelty). In this system demonstration we integrated HYDRA with AFSIM, a high-fidelity military simulator. We demonstrated HYDRA's ability to autonomously detect, characterize, and adapt to varied unexpected changes in the environment. HYDRA successfully completed realistic U.S. Navy aircraft missions, including overcoming novelties based on actual experiences from past missions, such as improved enemy weapons, adverse weather regions, or underestimated enemy vessel survivability. Future work will extend HYDRA to reason with and adapt to new classes of novelties in real-world scenarios.

References

Bell, W. 2015. Joint Direct Attack Munition (JDAM). *US Air Force Hill AFB United States*.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. OpenAI Gym. *arXiv preprint arXiv:1606.01540*.

Clive, P. D.; Johnson, J. A.; Moss, M. J.; Zeh, J. M.; Birkmire, B. M.; and Hodson, D. D. 2015. Advanced framework for simulation, integration and modeling (AFSIM)(Case Number: 88ABW-2015-2258). In *Proceedings of the international conference on scientific computing (CSC)*, 73. The Steering Committee of The World Congress in Computer Science.

Fox, M.; and Long, D. 2006. Modelling Mixed Discrete-Continuous Domains for Planning. *Journal of Artificial Intelligence Research*, 27: 235–297.

Mohan, S.; Piotrowski, W.; Stern, R.; Grover, S.; Kim, S.; Le, J.; and De Kleer, J. 2023. A Domain-Independent Agent Architecture for Adaptive Operation in Evolving Open Worlds. *arXiv preprint arXiv:2306.06272*.

Piotrowski, W.; and Perez, A. 2024. Real-World Planning with PDDL+ and Beyond. *arXiv preprint arXiv:2402.11901*.

Piotrowski, W.; Sher, Y.; Grover, S.; Stern, R.; and Mohan, S. 2023a. Heuristic Search For Physics-Based Problems: Angry Birds in PDDL+. In *International Conference on Automated Planning and Scheduling*, 518–526.

Piotrowski, W.; Stern, R.; Klenk, M.; Perez, A.; Mohan, S.; de Kleer, J.; and Le, J. 2021. Playing Angry Birds with a Domain-Independent PDDL+ Planner. *International Conference on Automated Planning and Scheduling (Demo Track)*.

Piotrowski, W.; Stern, R.; Sher, Y.; Le, J.; Klenk, M.; deKleer, J.; and Mohan, S. 2023b. Learning to Operate in Open Worlds by Adapting Planning Models. *AAMAS International Conference on Autonomous Agents and Multiagent Systems*.

Stern, R.; Piotrowski, W.; Klenk, M.; de Kleer, J.; Perez, A.; Le, J.; and Mohan, S. 2022. Model-Based Adaptation to Novelty for Open-World AI. In *Proceedings of the ICAPS Workshop on Bridging the Gap Between AI Planning and Learning*.