Design and Evaluation of a Goal-Directed Autonomous Agent

Matthew Molineaux

Knexus Research Corporation; Springfield, VA 22153
matthew.molineaux@knexusresearch.com

Abstract

The ARTUE (Autonomous Response to Unexpected Events) system was built as a prototype to demonstrate the usefulness of Goal-Directed Autonomy. We provide an overview of some of the design decisions made in its construction, as well as a discussion of how we chose to evaluate it. We close with a brief discussion of interesting research questions raised by ARTUE's design.

Introduction

The ARTUE system is an agent architecture designed for three primary criteria: ARTUE should (1) respond competently in a complex environment, (2) handle unexpected anomalies within the environment gracefully, and (3) change its own goals when necessary to achieve a high level of autonomy. ARTUE was designed as a prototype system for a new conceptual agent model, *Goal-Directed Autonomy* (GDA), which can be briefly summarized as the following 6-step reasoning cycle: (1) plan and predict, (2) act, (3) check for discrepancies, (4) explain discrepancies, (5) formulate goals, and (6) manage goals. This model was created to establish a generalized procedure for meeting the second and third criteria; constant monitoring in step 3 leads to detection of anomalies, steps 4 and 5 provide a principled response, and step 6 provides a higher level organization for goals and goal-directed behavior. ARTUE is our design for a GDA agent, and attempts to fulfill all three criteria. In the rest of this paper we describe the design of ARTUE (Section 2), the design of its evaluation, (Section 3), and briefly discuss future research directions (Section 4).

Design of ARTUE

In the following paragraphs, we discuss how ARTUE's design developed, providing components for 5 of the steps of the GDA reasoning cycle (the *act* step being provided by the environment). We describe these components out of order, as the rationale for selecting a later component sometimes affected the choice of an earlier one.

As we see it, an anomaly occurs whenever the agent fails to predict the future correctly. We chose to take the view that the world is deterministic, and therefore, failure to predict arises from hidden state and/or external influences (or lack of knowledge, which is not addressed here). It is also possible to model the world as stochastic, in which case failure to predict might be caused when rare complications occur that an agent ignores for convenience. Our choice to model predictions as deterministic rather than stochastic was practical rather than necessary; we hope others will soon investigate probabilistic models in the framework of goal-directed autonomy.

Our choice of representation was influenced by the requirement of performing in complex environments, as well as the need to represent the causes of failure to predict hidden states and external influences. We chose the planning domain language PDDL+ (Fox and Long, 2006), which models the world using processes and events, which are continuous time-based representations of how the world changes *either* as the result of agent action or external influences. We also extended PDDL+ with a list of hidden predicates, so that we could reason about state elements that cannot be directly observed.

ARTUE's explanation system (which performs the 3rd step of the GDA cycle) was designed to abduce hidden state for reasoning about anomalies. To this end, we chose a freely available explanation system that incorporates a deterministic fact-based representation, the Assumption-Based Truth Maintenance System (ATMS) (de Kleer, 1986). We extended the ATMS with an input translator that constructed ATMS rules based on PDDL+ processes and events. This approach had the benefits of being well-founded and relatively easy to perform, but the disadvantage of relatively poor performance.

For creation of plans, ARTUE uses the SHOP2 Hierarchical Task Network planner (Nau et al. 2003), which is efficient and easily modifiable. We extended

SHOP2 with the capability to incorporate process and event models into its prediction and search mechanisms. See (Molineaux *et al*., 2010b) for a comprehensive discussion and analysis of these extensions. The extended SHOP2 provided the plans and predictions needed in the first step of the GDA cycle.

A component responsible for monitoring and checking the state for discrepancies (the third step of the GDA reasoning cycle) was relatively simple to design given the standard planning state formalism of SHOP2. A bidirectional set difference operation between a prediction and an observation provides a list of atoms that either were predicted and did not occur, or occurred without being predicted.

Goal formulation, step 5, is perhaps the most novel, and least understood step of the GDA cycle. Our second design criterion implies that the designer of a GDA agent may not know in advance what anomalies will occur, so we did not want to provide background knowledge about what goals to pick based on individual anomalies. However, forming a response goal requires some sort of knowledge about the environment and what is important to the agent. In response to this need, we designed a rule-based system that operates based on the agent's *principles*, and creates new goals whenever the conditions of a principle match the properties of the state. Thus ARTUE can formulate a goal for any of the principles defined, which represent high-level values such as "if people are in danger, help them", or "if a threat exists, remove it". The problem of specifying these principles without knowing what anomalies the agent may encounter is still underspecified, and remains an interesting open research problem.

For goal management, the sixth step, we picked a very simple strategy, which may provide a useful baseline for future work: each goal is associated with an ordinal value, called its *intensity*, and the goal with the highest intensity at any given time is selected for planning.

Design of ARTUE's Evaluation

We chose to demonstrate ARTUE in two environments. First, the Tactical Action Officer (TAO) Sandbox (Auslander *et al*., 2009) is a training simulator designed for use by military officers. Second, Battle of Survival (BoS), is a real-time strategy game based on the Stratagus engine (Ponsen *et al*., 2005), which has received attention from several of AI research groups. These were selected because they both have hidden state, continuous time representations, a variety of goals to achieve, and because both environments include external (to the agent) influences that can change the state. In the TAO Sandbox, outside influences include neutral ships and enemy submarines which move around on their own, fire torpedoes, and lay mines. In BoS, enemy vehicles move and fire upon friendly vehicles.

Recent evaluations of ARTUE have been based on comparing the performance of ARTUE to the performance of a set of ablations, on a set of scenarios which test the three design criteria introduced earlier. Performance has to date been determined using scenario-specific metrics that judge the extent to which the initial goals of the system are completed, as well as the extent to which ARTUE responds in a reasonable way to developing anomalies. Each scenario in our evaluations includes a "scheduled" anomaly that occurs at a specified time without warning, to test criterion 2, "handle unexpected anomalies". To ensure that criterion 3 is also tested, we designed each scenario to incorporate an anomaly-based challenge outside the scope of the initial goal provided by the scenario. ARTUE can \ respond to this challenge only by changing the focus of what it's working on, meaning that ARTUE must form a new goal. To ensure that each scenario has sufficiently high complexity, thus satisfying criterion 1, each anomaly has hidden properties that must be understood in order to respond correctly. For example, in the TAO Sandbox's "Iceberg" scenario, an anomaly occurs when a nearby ship founders on an iceberg. Responding correctly requires ARTUE to save the passengers aboard the vessel, which tests the second criterion. The initial goal in this scenario is to carry cargo to a nearby destination, so ARTUE must change its own goal in order to respond, which tests criterion 3. Finally, ARTUE must assume the presence of an iceberg to understand that the passengers are endangered, which tests criterion 1. The results of evaluations that follow this design are reported in (Molineaux *et al*., 2010a) and (Klenk *et al*., in submission).

Conclusion

ARTUE is successful, in terms of meeting the criteria we initially established. However, its design suggests several issues that have not been considered. One important unaddressed issue is that of stochastic environments. ARTUE was not written with extension to probabilistic tasks in mind, and this choice influenced many later design choices. A new architecture based on the same design criteria and a probabilistic foundation would provide an interesting comparison. Second, ARTUE currently incorporates no learning faculties. In particular, it would be interesting to investigate whether it is possible to learn any background knowledge or heuristics for goal formulation. Third, the space of knowledge for supporting goal formulation is still only shallowly explored. What structures beyond principles are interesting and useful? We hope that other researchers will find these design questions interesting, and that future research in this area will push the frontiers of autonomy.

References

Auslander, B., Molineaux, M., Aha, D.W., Munro, A., & Pizzini, Q. (2009). *Towards research on goal reasoning with the TAO Sandbox* (Technical Report AIC-09-155). Washington, DC: Naval Research Laboratory, Navy Center for Applied Research on AI.

Fox, M. & Long, D. (2006). Modelling mixed discrete-continuous domains for planning. *Journal of Artificial Intelligence Research*, **27**, 235-297.

de Kleer, J. (1986) An assumption-based TMS. *Artificial Intelligence,* **28**(2), 127-162

Klenk, M., Molineaux, M., & Aha, D.W. (in submission). Goal-driven autonomy for responding to unexpected events in complex environments. Manuscript submitted for publication.

Molineaux, M., Klenk, M., & Aha, D.W. (2010a). Goal-driven autonomy in a Navy strategy simulation. To appear in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*. Atlanta, GA: AAAI Press.

Molineaux, M., Klenk, M., & Aha, D.W. (2010b). Planning in dynamic environments: Extending HTNs with nonlinear continuous effects. To appear in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*. Atlanta, GA: AAAI Press.

Nau, D., Au, T.-C., Ilghami, O., Kuter, U., Murdock, J. W., Wu, D., & Yaman, F. (2003). SHOP2: An HTN planning system. *Journal of Artificial Intelligence Research*, **20**, 379–404.

Ponsen, M.J.V., Lee-Urban, S., Muñoz-Avila, H., Aha, D.W., & Molineaux, M. (2005). Stratagus: An open-source game engine for research in real-time strategy games. In D.W. Aha, H. Muñoz-Avila, & M. van Lent (Eds.) *Reasoning Representation, and Learning in Computer Games: Papers from the IJCAI Workshop* (Technical Report AIC-05-127). Washington, DC: Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence.