

Self Monitoring Goal Driven Autonomy Agents

Dustin Dannenhauer

Department of Computer Science and Engineering
Lehigh University, Bethlehem, PA 18015
dtd212@lehigh.edu

1 Research Problem

As intelligent systems become a regular part of our everyday lives, robust and safe operation is ever more important. My research focus is to endow agents with the ability to monitor themselves in order to detect when their behavior has exceeded their boundaries. Previously, we have explored different forms of expectations for anomaly detection in agents operating in Real-Time Strategy (RTS) games, as well as dynamic domains involving planning and execution. My current work aims to achieve agents that can reason about and use expectations in both dynamic and partially observable domains, as well as investigating meta-cognitive expectations for detecting anomalies in the agent’s own cognitive processes (reasoning, planning, etc) instead of anomalies in the world.

2 Goal Driven Autonomy

Goal-driven autonomy (GDA) is an agency model where an agent revises its goals by reasoning over discrepancies. Discrepancies arise when the agent’s own expectations do not match the agent’s observations. Such discrepancies arise when acting in dynamic environments (i.e., changes occur independently from the agent’s actions). When discrepancies occur, the GDA agent will suggest alternative goals. An example, adapted from [Molineaux *et al.*, 2010], involves an agent performing navy operations. A naval convoy is in route to deliver some equipment and along the way an escort vessel identifies an unknown contact. At this point the agent could pursue one of multiple alternative goals including (1) abort the mission and route back the vessels to the departing port or (2) hold the convoy and send escort vessels to identify the contact.

Figure 1 shows a GDA situated agent. The core GDA process is shown within the controller. After discrepancy detection identifies an anomaly, the discrepancy d is sent to the explanation generator which hypothesizes one or more explanations e . These explanations are then used by the goal formulator which may decide to formulate new goals g . Finally the goal manager selects which goals for the agent to pursue. An underlying motivation for GDA is that in the face of an anomaly, it may be better for an agent to change its goal(s) instead of replanning.

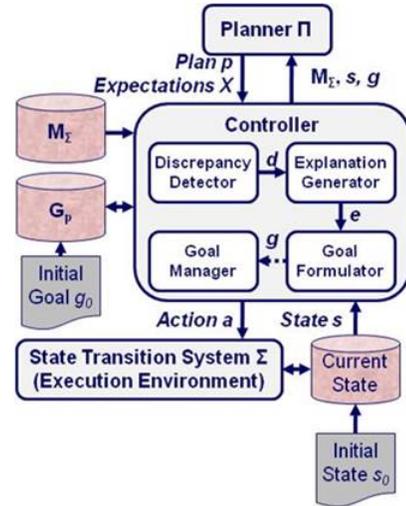


Figure 1: The GDA Cycle

3 Expectations in Real Time Strategy Games

Expectations are central to discrepancy detection, as they are the knowledge the agent uses to identify when an anomaly is present, or when the agent has acted outside its boundaries. The source of expectation knowledge and method of detection differs depending on the agent and the domain. In the RTS game Starcraft, we showed that inferred expectations enable high level planning to allow an agent to use more complex plans (e.g. coordinating different types of attacks using multiple groups to attack the enemy) [Dannenhauer and Muñoz-Avila, 2013a; 2015b].

In many of the top Starcraft playing agents, the focus is on winning battles as a single army and producing new troops as efficiently as possible. However, humans are often able to beat these bots by using more tactical strategies such as harassing the enemy workers at the base with one group of units and then attacking the other side of the base with another group. In order to enable these kinds of explicit higher level goals, such as “surround the enemy base”, more expressive expectations are needed. Previous expectations used in RTS games were based on individual units [Weber *et al.*, 2010; Jaidee *et al.*, 2012]. In our work the agent can produce an expectation such as “player0 controls regions 5 and 6” (where

player0 is the agent). This kind of expectation is possible with the use of rules, such as a description logic rule: “A player controls a region if there is at least 1 unit in the region and that player owns every unit in the region.” (Partitioning a map into polygon shaped regions is possible with the API Brood War Terrain Analyzer [Perkins, 2010]). The discrepancy detector can identify if this expectation was met by adding the expectation statement (player X controls region Y) to an ontology of facts of the current game state, running it through a reasoner, and checking consistency. Upon an inconsistency the reasoner provides an explanation trace of what facts are conflicting. In this trace would be the DL rule which would have as its consequent the fact that player X controls region Y. The trace also shows which antecedents of the rule are inconsistent, which enables useful explanation.

We have also explored evidence scoring functions to determine strategies in individual battle scenarios in RTS games [Dannenhauer and Muñoz-Avila, 2013b].

4 HTN Based Expectations

In more recent work we have examined new techniques for expectations generated from Hierarchical Task Network (HTN) planning domain knowledge for agents operating in dynamic domains [Dannenhauer and Muñoz-Avila, 2015a]. We introduced Informed expectations: expectations that are the accumulated effects from the planning actions executed thus far. This approach filters out the atoms in the state that are irrelevant to the agents current goal, while still maintaining effects that are needed to reach the goal. In that work, we assumed that enough of the state is known to enable generating concrete grounded plans (i.e. a complete plan is generated for the agent’s goals).

We are interested in exploring domains where perception is limited enough such that it requires the agent to explore its environment to determine what resources it has available before being able to achieve its goals. Generating a grounded plan from the beginning would be impossible, because the agent would not know where it needs to go or what objects it will encounter to use to accomplish its goals. For example, suppose an agent is operating in a domain similar to the Marsworld domain described in [Dannenhauer and Muñoz-Avila, 2015a].

The overall goal of the agent is to create a signal. The agent must either activate a specified number of beacons, drop and light a certain number of flares, or create a specified number of smokefires using piles of wood found nearby. If the agent has limited perception, then only after it explores the state will it know if it is better to activate beacons or go about dropping flares. Thus classical grounded planning is not possible. In these environments agents would likely have models for sensing actions and associated costs. We aim to explore sensing vs. acting as a form of self-monitoring. Additionally, we are interested in dynamic domains that have significant changes over time, requiring the agent to adapt in order to maintain performance.

5 Metacognitive Expectations

Anomalous or failed behavior of an agent may result from a problem in a cognitive process instead of a world state. To detect and correct such a problem relies on metacognitive capabilities. Previously we have explored using a cognitive trace that records data from different cognitive processes (e.g. preception, interpretation, goal selection, planning) and show preliminary results that enable an agent to swap out its planning faculty for an alternative planner [Cox, 2016]. We would like to explore metacognition in other cognitive processes including but not limited to: goal selection, memory, and perception.

Acknowledgements: This work was supported in part by grants NSF 1217888 and ONR N00014-15-1-2080

References

- [Cox, 2016] Alavi Z. Dannenhauer D. Eyorokon V. & Munoz-Avila H. Cox, M. T. Midca: A metacognitive, integrated dual-cycle architecture for self-regulated autonomy. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*. Palo Alto, CA: AAAI Press., 2016.
- [Dannenhauer and Muñoz-Avila, 2013a] D. Dannenhauer and H. Muñoz-Avila. LUGI: A Goal-Driven Autonomy Agent Reasoning with Ontologies. In *Advances in Cognitive Systems (ACS-13)*, 2013.
- [Dannenhauer and Muñoz-Avila, 2013b] Dustin Dannenhauer and Héctor Muñoz-Avila. Case-Based Goal Selection Inspired by IBM’s Watson. In *Case-Based Reasoning Research and Development*, pages 29–43. Springer, 2013.
- [Dannenhauer and Muñoz-Avila, 2015a] D. Dannenhauer and H. Muñoz-Avila. Raising Expectations in GDA Agents Acting in Dynamic Environments. In *International Joint Conference on Artificial Intelligence (IJCAI-15)*, 2015.
- [Dannenhauer and Muñoz-Avila, 2015b] Dustin Dannenhauer and Héctor Muñoz-Avila. Goal-driven autonomy with semantically-annotated hierarchical cases. In *Case-Based Reasoning Research and Development*, pages 88–103. Springer, 2015.
- [Jaidee *et al.*, 2012] Ulit Jaidee, Héctor Muñoz-Avila, and David W Aha. Learning and Reusing Goal-Specific Policies for Goal-Driven Autonomy. In *Case-Based Reasoning Research and Development*, pages 182–195. Springer, 2012.
- [Molineaux *et al.*, 2010] Matthew Molineaux, Matthew Klenk, and David W Aha. Goal-Driven Autonomy in a Navy Strategy Simulation. In *AAAI*, 2010.
- [Perkins, 2010] Luke Perkins. Terrain Analysis in Real-Time Strategy Games: An Integrated Approach to Choke Point Detection and Region Decomposition. In *AIIDE*, pages 168–173, 2010.
- [Weber *et al.*, 2010] Ben George Weber, Michael Mateas, and Arnav Jhala. Applying Goal-Driven Autonomy to StarCraft. In *AIIDE*, 2010.