The Thirty-Third International
FLAIRS Conference (FLAIRS-33)

Case-Based Explanations and Goal Specific Resource Estimations

Venkatsampath Raja Gogineni,[1] Sravya Kondrakunta,[2] Matthew Molineaux,[3] Michael T. Cox[4]
Wright State University,[1,2] Wright State Research Institute [3,4]
gogineni.14@wright.edu[1]

Abstract

Autonomous agents often have sufficient resources to achieve the goals that are provided to them. However, in dynamic worlds where unexpected problems are bound to occur, an agent may formulate new goals with further resource requirements. Thus, agents should be smart enough to manage their goals and the limited resources they possess in an effective and flexible manner. We present an approach to the selection and monitoring of goals using resource estimation and goal priorities. To evaluate our approach, we designed an experiment on top of our previous work in a complex mine-clearance domain. The agent in this domain formulates its own goals by retrieving a case to explain uncovered discrepancies and generating goals from the explanation. Finally, we compare the performance of our approach to two alternatives.

Introduction

Autonomous agents that operate in a dynamic world often encounter unexpected events or discrepancies that signal the existence of novel problems. These agents will perform better when they can reason about and change their existing goals or formulate new ones in response to such discrepancies. The *goal reasoning* (Aha 2018; Cox 2007; Munoz-Avila et al. 2010; Molineaux, Klenk and Aha 2010; Dannenhauer and Munoz-Avila 2015) approach we use for goal formulation is to explain the cause behind a discrepancy and to specify a goal to eliminate the cause. In doing so, the agent aims to prevent a reoccurrence. However, goal formulation often creates multiple goals, and each goal requires an independent amount of resources to achieve it. Since agents often have limited resources, they must be careful in their goal selection and management. Our approach to these problems is to provide each goal with corresponding resource and priority estimates to help goal management. The contribution of this paper is the strategy that uses these estimates to manage the formulation, selection and pursuit of goals in an effective manner given the potential for future changes.

In our system, case-based explanations are retrieved by the event understanding system Meta-AQUA (Cox and Ram 1999) and applied to discrepancies in an event sequence. We have also integrated this sub-system within the MIDCA cognitive architecture (Cox et. al. 2016) that perceives and acts directly on the world to examine the interaction between explanation generation (by Meta-AQUA), goal formulation and management (provided by MIDCA). We refer to the combined system as the *Goal-driven Autonomy for Trusted Autonomous Reasoning (GATAR)* agent.

The paper continues as follows. Firstly, we introduce the technical approach which include a casebase of explanations, the agent's algorithm to manage goals, case retrieval, and goal formulation. Later we introduce a mine clearance domain, examples that might happen in the domain along with explanations and formulated goals. The section following it discusses the experimental setup for evaluating the performance of the GATAR agent, a random goal selection agent and a standard agent which does not formulate new goals followed by the results. Related research is discussed in the next section. The conclusion and future scope complete the paper.

Technical Approach

In our work, we use case-based explanations (Ram 1994; Cox and Burstein 2008; Schank et al. 2014). Each explanation in our case-base is an abstract *explanation pattern (XP)* (Schank 1986) as shown in Figure 1. An XP is a data structure that represents a causal relationship between multiple states and/or actions; variables adapted during or after case retrieval abstractly define each action/state. An action or state is referred to as a node, and different types of nodes are described based on their role in an XP as follows.

- *Explains node*: An unexpected observed action or state (i.e., the target of the XP).
- *Pre-XP node*: An observed action/state along with the Explains node.
- *XP-asserted node*: An action, state, or XP that contributes to the cause of the Explains node.

An explanation pattern represents a causal structure in which XP-asserted nodes form an antecedent and Pre-XP nodes form the consequent. The Pre-XP nodes represent those states that must hold for the XP to be a candidate, including the Explains node itself.

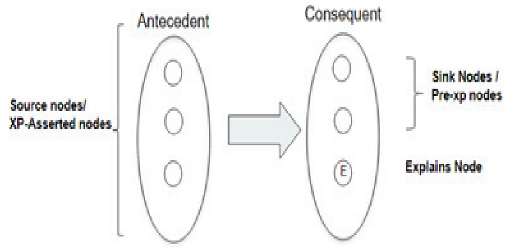

Figure 1. The XP Explanatory Case Structure.

Discrepancy Response and Goal Management

Algorithm 1 represents GATAR's approach towards executing in its environment (Gogineni et al. 2019). Following the notation of Ghallab, Nau, and Traverso (2004), this algorithm takes the following inputs: the environment model (Σ), the current observations of the world (s_c), expected observations of the world (s_e), case library (L), current goal (g_c) and a goal agenda ($\hat{G} = \{g_1 \dots g_c \dots g_m\}$). GATAR creates a plan ($\pi = < a_1 \dots a_n >$) of n actions to achieve the initial goal and starts executing the initial plan by modifying its goal agenda when there is a *discrepancy* (i.e., a condition such that the observed state significantly differs from an expected state; see Dannenhauer, Munoz-Avila and Cox 2016). Algorithm 1 extends our previous work in lines 10 to 17 (and line 2). *ManageGoals* monitors the goals and selects the goal that takes minimum resources with maximum priority (line 10). If the current goal (g_c) is switched with a selected goal (g_s) (line 11 and 12), GATAR plans to achieve the selected goal (g_s) (line 13). Similarly, when the plan for the current goal is completed (line 14) then GATAR removes the current goal from the goal agenda ($\hat{G} = \{g_1 \dots g_m\}$) (line 15), selects the goal (line 16) and plans to achieve the selected goal. This continues while GATAR has resources available, as determined by function ($R: S \rightarrow \mathbb{R}^+$) characterizing the resources remaining in a current state using a positive real number.

Algorithm 2 presents GATAR's approach towards goal management. The function *ManageGoals* is called with a current state and existing goal agenda of the most recent discrepancy. It first adds to the goal agenda all goals proposed by the current explanation (Line 1), then calculates the set C of candidate goals whose resource requirements can be satisfied (Line 2). To determine which goals can be satisfied, it uses a hand-coded domain specific resource estimation function ($\hat{R}: G \times S \rightarrow \mathbb{R}^+$) that estimates the amount of resources required to achieve a goal (g) from a state (s). GATAR then filters the set C down to the highest priority goals (line 3) with minimum resource utilization (line 4). The priority function ($\hat{P}: G \times S \rightarrow \mathbb{R}^+$) is a hand-coded domain-specific function that assigns a real-valued priority to a goal (g) in a state (s). Finally, GATAR returns an arbitrary member of candidate goals C (which will usually be unique), along with an updated agenda (line 5).

$\boldsymbol{ExecuteAndMonitor}\ (\boldsymbol{\Sigma}, \boldsymbol{s_c}, \boldsymbol{s_e}, \boldsymbol{L}, \boldsymbol{g_c}, \boldsymbol{\hat{G}})$

1. $\pi \leftarrow Plan(\Sigma, s_c, g_c)$
2. $while\ R(s_c) > 0\ do$
3. $\quad s_c \leftarrow \gamma\ (s_c, \pi[1])$
4. $\quad \pi \leftarrow \pi[2..n]$
5. $\quad s_e \leftarrow s_e \cup pre(\pi[1]) \cup \pi[1]^+ - \pi[1]^-$
6. $\quad if\ \ s_c \not\models s_e$ // discrepancy exists
7. $\quad\quad \chi_c \leftarrow Select(Retrieve\ (L, s_c))$
8. $\quad\quad s_e \leftarrow s_e \cup s_c \cup XP\text{-}asserted\ (\chi_c)$
9. $\quad\quad \hat{G} \leftarrow \hat{G} \cup FormulateGoals(\chi_c)$
10. $\quad g_s, \hat{G} \leftarrow ManageGoals(s_c, \hat{G})$
11. $\quad if\ g_c \neq g_s$
12. $\quad\quad g_c \leftarrow g_s$
13. $\quad\quad \pi \leftarrow Plan(\Sigma, s_c, g_c)$
14. $\quad if\ \ \pi = \emptyset$
15. $\quad\quad \hat{G} \leftarrow \hat{G} - g_c$
16. $\quad\quad g_c, \hat{G} \leftarrow ManageGoals(s_c, \hat{G})$
17. $\quad\quad \pi = Plan(\Sigma, s_c, g_c)$

Algorithm 1. Execute and Monitor

$\boldsymbol{ManageGoals}(\boldsymbol{s_c}, \boldsymbol{\hat{G}})$

1. $C \leftarrow \{g \in \hat{G} \mid \hat{R}(g, s_c) \leq R(s_c)\}$
2. $C \leftarrow \underset{g \in C}{\operatorname{argmax}}\, \hat{P}(g, s_c)$
3. $C \leftarrow \underset{g \in C}{\operatorname{argmin}}\, \hat{R}(g, s_c)$
4. return $C, \hat{G}$

5. Algorithm 2. Goal Management

Retrieving an Explanation Pattern

To retrieve an XP, GATAR tries to unify a discrepant state or an action with the Explains node of the XP's in the case base. When such a unification is successful, then the Pre-XP nodes of the corresponding case are unified with the observations of corresponding states or actions. Also, if this unification is successful then an XP is retrieved. From the retrieved sets of explanations, an explanation is selected based on a Bayesian inference technique to select and monitor explanatory cases (Gogineni et al. 2019). Finally, Goals are formulated using the selected explanation. The following section describes the goal formulation process given an XP.

Goal Formulation

Goal formulation is essential for an intelligent agent to respond to unpredictable events. In our previous work

(Gogineni et al. 2018), we have formulated goals from the antecedent of the selected explanation pattern. There can be a maximum of three possible types of goals from an explanation pattern: goal to eliminate the actor, goal to eliminate the results of the action by the actor and goal to avoid the discrepancy. Often an actor is responsible for the cause of a discrepancy, so a potential goal to eliminate the actor will hinder the occurrence of discrepancy in future. A potential goal to eliminate the results of the action of the actor is helpful in responding to the current discrepancy that hinders agents' mission. Finally, a potential goal to avoid a discrepancy is to look at the alternate opportunities in which the agent can avoid discrepancies both in present and in future.

For example, from an explanation pattern with a causal relation, an enemy ship laid the mines in a gaussian pattern to hinder the traversal of the freight ships. A potential goal to eliminate the actor is to apprehend the enemy ship. Similarly, a potential goal to eliminate the results of the action is to survey and clear mines in the gaussian pattern. Finally, a potential goal to avoid the discrepancy is to survey and change the route of the freight ships to an alternate route.

Goal Management

Often in real world the agent has very limited resources to achieve its goals. In these situations, the agent must be smart about selecting the goals it can achieve. GATAR uses both priority and resource estimation functions to select and monitor goals. In the underwater mine clearance domain priorities are assigned to goals pertaining to the traversal route of freight ships. For example, if the agent has goal to survey regions where the freight ships traverse and a goal to survey other regions, priority is given to the former.

Each goal type has a resource estimation function to calculate the resources it takes to achieve the goals. To select a goal, GATAR sorts its goals based on the priority. Later, from the sorted list it chooses a goal that takes minimum resources to achieve and monitors the remaining goals. Whenever a monitored goal with the same or higher priority takes less resources to achieve then the selected goal is switched with the monitored goal. Furthermore, if the selected goal cannot be achieved with the current resources the agent has, then it switches the goal with the next possible priority goal that takes minimum resources to achieve and monitors other goals.

Table 1 shows the type of goals and the resource estimation functions used for the Underwater Mine Clearance Domain. Mine Clearance Goals correspond to the agent surveying and clearing mines at specific region. To estimate the time it takes to achieve these types of goals, we have estimated the number of mines the agent can encounter based on Poisson distribution and time taken to clear mines based on an encounter with mines. The Poisson distribution takes the average number of mines encountered in a certain time period as an input and results in the probability of number of mines we might encounter in future time period. We have chosen the probability threshold as 70%.

Apprehension goals correspond to the agent apprehending the enemy. These goals use Distance-based function to estimate the time taken to reach the enemy and return to the current position. A Distance-based function uses speed and distance to calculate the time it takes to apprehend plus the apprehension time which can be learned from the previous encounters.

Equipment goals include rectifying or repairing the agent's failed sensor and use a Distance-based function to estimate the time it takes to reach the repair point plus the repair time, which it learns from previous encounters.

Switching Location goals correspond to the goals of the agent in switching the surveying regions. To estimate the time to reach the new destination, we use the previous Distance-based function estimates and the Poisson distribution estimates to survey and clear mines at the switched location.

Goals	**Resource Estimation**
Mine Clearance Goals	Poisson distribution
Apprehension Goals	Distance-based function
Equipment Goals	Distance-based function
Switching Location	Distance-based function + Poisson distribution

Table 1. Resource Estimation Functions for Goals

The Mine Clearance Domain

Our approach is implemented in a mine clearance domain (Kondrakunta et al. 2018), which is simulated using MOOS-IVP (Benjamin et al. 2010), software that provides complete autonomy for marine vehicles. Figure 2 represents the mine clearance domain with all the agents, regions and mines. The red cylindrical object towards the top left corner is a Remus, which is an unmanned underwater vehicle controlled by the GATAR. Remus has a mine detection sensor represented as a grey region around it. Yellow objects, near the shore, to

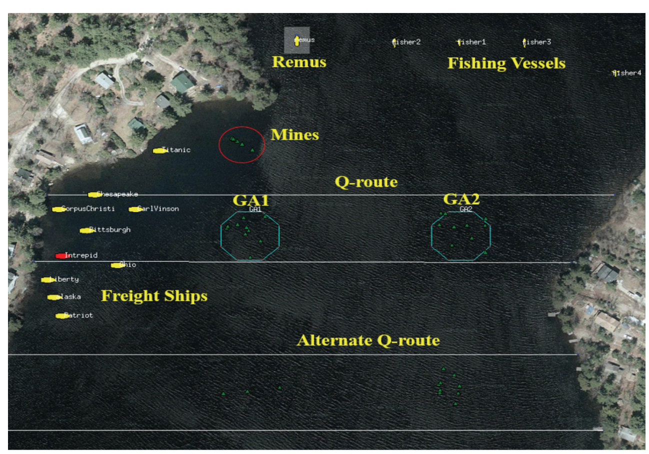

Figure 2. Simulation of Mine Clearance Domain.

the left side of the image are freight ships and the four objects on the top right of the image are fishing vessels. Green triangles are the mines that hinder the ability of freight ships. Area between the two parallel lines is the Q-route. A Q-route is an area where the freight ships use to transport their shipments. There are two Q-routes in this domain Q-route1 and Q-route2. The two octagons are green areas namely GA1 and GA2, where the mines are expected to be present. Clearing mines in these areas are the explicit goals of GATAR.

Discrepancies that occur in the domain

There are several discrepant events that might occur simultaneously in this domain. These events often affect the agent or its mission. GATAR uses its smart goal reasoning strategy to respond to these events. These discrepant events include minelaying, sensor failure, misclassification, reconnaissance failure and environmental effects. Minelaying events occur when an aerial vehicle, enemy ship or a fishing vessel lays a pattern of mines to trap freight ships. Removing such mines within areas GA1 and GA2 is an explicit goal for GATAR. Alternatively, a sensor failure indicates a fault in the agent's sensor to detect and classify mines. Misclassification event indicates that the agent's action of misclassifying a benign to be a mine. A benign is an object that resembles a mine but not a mine. Similarly, environmental effects include tides that displace mines.

Possible Explanations and Goals for the Domain

Whenever GATAR detects a discrepancy, it retrieves and selects an explanation from the case-base to formulate new goals. These goals are down selected based on the availability of resources. Table 2 shows seven different explanations explaining the discrepant events that might happen in the domain. Each explanation has a minimum of one and a maximum of three different goals. In the table, clear-area describes the area where no mines are expected.

Experimental Design

To evaluate the contribution of this paper represented by the *GATAR strategy*, we have introduced two other strategies, namely the *Random Goal Selection* strategy and the *No Goal Formulation* strategy. Each of the agents is assumed to perform the same task of detection and clearing all the mines in GA1 and GA2 and that they will also detect the same set of discrepant mines at different regions in the domain. However, they all respond differently to the discrepant mines. An agent with the Random Goal Selection strategy like the GATAR agent will detect the discrepant mines, retrieves and selects an explanation to explain the discrepancy and formulates goals. However, they differ in their goal selection strategies. The GATAR agent uses a smart selection strategy to select and monitor goals, while the Random Goal Selection agent selects a goal at random and tries to achieve it. Alternatively, No Goal Formulation strategy will only achieve the given set of goals and ignores the discrepancies it comes across.

Explanations	Discrepancies	Goals
Fisher-XP: Fishing-vessel laid a single mine	Single mine detected at clear area.	1.Apprehended (fishing-vessel) 2.ClearedMines (remus, clear-area) 3.ChangedRoute (clear-area, alternate-route)
Sensor-XP: Remus sensor failure	Single mine detected at clear-area or mines not detected at the expected areas.	1.ChangedAgent (remus, alternate-agent), 2.Calibrated (remus, sensor), 3.ChangedSensor (remus, sensor)
Tide-XP: Mines drifted from expected regions GA1 or GA2 with tides	Single or multiple mines detected at clear-area or mines not detected at the expected areas.	1.ClearedMines (remus, clear-area)
Reconnaissance-XP: Reconnaissance failed	Multiple mines detected in the clear-area	1.ReportAgent (remus, previous-agent) 2.ClearedMines (remus, all-clear-areas) 3.ChangedRoute (clear-area, alternate-route)
Enemy-Sub-XP: Enemy ship laid the circular pattern of mines	Multiple mines detected in the clear-area	1.Apprehended (enemy-ship) 2.ClearedMines (remus, clear-area) 3.ChangedRoute (clear-area, alternate-route)
Enemy-Aerial-XP: Aerial vehicle laid the line pattern of mines	Multiple mines detected in the clear-area	1.Reported (aerial-vehicle), 2.ClearedMines (remus, clear-area) 3.ChangedRoute (clear-area, alternate-route)
Benign-XP: Remus misclassified object as mine	Single or Multiple mines detected	1.ChangedAgent (remus, alternate-agent) 2.Calibrated (remus, sensor), 3.ChangedSensor (remus, sensor)

Table 2. XPs for Discrepancies with Resulting Goals.

Each of these agents with the above-mentioned strategies are run across ten different scenarios. Each scenario contains: ten freight ships that transport their shipments using the Q-route, fishing vessels, possible chance of enemy disguised as a ship or a fishing vessel. Furthermore, there are different patterns of mines that can occur simultaneously across different regions. Possible mine patterns are: Gaussian distribution of mines, linear distribution of mines, and a single mine. Table 3 shows different distributions of mine patterns at different areas along with their corresponding probabilities to be chosen randomly across ten scenarios.

Performance of the three agents are measured by percentage of freight ships that safely traverse from one end of the Q-route to the other.

Regions	Mine Pattern	Prob.
GA1, GA2	Gaussian mine pattern	0.5
	Line pattern of mines	0.3
	Single mine	0.1
	No mines	0.1
Transit area between GATAR's starting point and GA1, Transit area between GA1 and GA2	Many Gaussian mine patterns	0.3
	Gaussian mine pattern	0.2
	Multiple line patterns	0.2
	Line pattern of mines	0.2
	Single mine	0.1
Alternate Q-route	No mines	0.2
	Gaussian mine pattern	0.3
	Line mine pattern	0.25
	Many Gaussian mine patterns	0.25

Table 3. Probability Occurrence of Mine Laying Patterns across Different Regions in Various Scenarios.

Empirical Results

Figure 3 depicts the performance of three agents mentioned earlier. The agents follow three different strategies: The GATAR strategy, Random Goal Selection, and No Goal Formulation. On the x-axis is the time delay with which the freight ships start their voyage and the y-axis presents the percentage of ships that successfully completed their journey across the Q-route.

Figure 3. Results Obtained for Different Scenarios.

Time delay of the ships start at 0 and end at 200 seconds, these values are incremented in intervals of 50 sec. Each second in simulation time is approximately equal to 20 seconds in wall-clock time. There are ten ships in each experiment, so the maximum percentage, 100, refers to all ten ships reaching the destination. Each data point in the graph is an average of ten different scenarios.

Time delay 0: Ships start traversing when Remus starts its mission. Remus does not have time to clear any mines. Hence all three agents perform similarly. In scenarios where mines are sparsely placed, very few ships pass through the Q-route. Hence, there is an initial percentage of 17.

Time delay 50: When following The GATAR strategy, Remus enters the Q-route and clears GA1 and a few mines in transit between GA1 and GA2 before the ships starts. Therefore, there is an increase in the performance. No Goal Formulation agent clears mines in GA1 and will be traversing towards GA2 ignoring all the mines it encounters while travelling. Random Goal Selection agent in some scenarios acts on the goals to apprehend the enemy even before it clears GA1, therefore there is a slight decrease in performance of the Random Goal Selection agent.

Time delay 100: The GATAR agent clears all the mines in GA1, Transit and might apprehend the enemy if it is near Remus. Remus also clears some mines in GA2 if mine density is estimated to be sparse. This agent might also generate a goal to change the Q-route if the estimated density of mines in Q-route is dense, hence the sharp increase in the performance of the agent. Random Goal Selection agent generates goals at random and pursues them without reasoning about the outcome, even then those goals are formulated to eliminate the discrepancy. So, there is an increase in performance of the agent, but the increase is not as significant as the smart agent. No Goal Formulation agent just clears all mines in GA1 and GA2 and returns home.

Time delay 150: The GATAR agent now has more time than earlier to either clear all mines in Q-route including all in GA2, if the density is sparse to medium. If the density is too high, then Remus might change the Q-route and clear the mines in the new Q-route. Therefore, there is an increase in the performance when compared to the previous value. Random agent might choose better goals in some scenarios, but it also has more time to clear additional mines. So, there is an increase in the performance for this agent as well. No Goal Formulation agent remains constant as it did not clear any extra mines when compared to the previous scenario.

Time delay 200: Performance of Smart agent remains constant, as it did the best it can with the previous deadline. The performance of the Random agent dropped because it pursued the goals that are of lesser value. For example, it spent resources to apprehend the enemy rather than clearing mines. Once again there is no change in the No Goal Formulation agent.

Performance of the smart agent clearly depicted significant improvement when compared to other traditional and a slightly smarter agent. This clearly shows the use case of having a resource and priority estimation functions to select and manage goals.

Related Research

Goal management has been a key focus of the goal reasoning research. Goal selection and monitoring plays a vital role in managing goals. (Kondrakunta and Cox 2017) presents a goal selection strategy to look at the ratio of cost to benefit in selecting goals. This work closely aligns with our proposed selection strategy, it uses domain specific estimations to determine the cost and the benefit for a set of goals. However, this paper did not pursue the interactions with selection and monitoring goals which is one of the key focus to better performance of GATAR.

Similarly (Dannenhauer and Cox 2018) introduced the idea of goal monitors. An agent creates rules as preconditions to monitor goals. If the preconditions are satisfied then the agent switches its goal, else drops the goal. This paper did not look at the problem of selecting goals when there are multiple of them to achieve. Moreover, the preconditions are mostly rule based rather than any kind of estimation function, which is often a problem when the agent has very limited resources at hand.

Conclusion and Future Scope

In this paper, we discussed an approach to select and monitor goals when the agent has very limited resources to achieve its goals. Our approach uses goal specific resource and priority estimation function to select and monitor goals. Finally, the results show that the performance of GATAR is better than the two other agents presented.

We acknowledge that our current experimentation process uses estimations manually designed to fit the domain, however in future, we would like to use an automated learning process to relax this assumption. In addition to the above-mentioned experimentation, we would like to explore more with resource and priorities as vectors instead of a positive real number. Moreover, we also want to implement this across different domains and test the generalizability of this approach.

Finally, we also want GATAR to learn these estimation functions from the knowledge obtained from the retrieved explanations.

Acknowledgments

This research is funded by AFOSR under grant FA2386-17-1-4063, by ONR under grant N00014-18-1-2009 and by NSF under grant 1849131. We thank Anthony Mitchell, Danielle Brown and the anonymous reviewers for their comments and suggestions.

References

Aha, D. W. 2018. Goal reasoning: Foundations, Emerging Applications, and Prospects. AI Magazine 39(2): 3-24. doi.org/10.1609/aimag.v39i2.2800.

Benjamin, M. R.; Schmidt, H.; Newman P. M.; and Leonard, J. J. 2010. Nested autonomy for unmanned marine vehicles with MOOS-IvP. Journal of Field Robotics 27(6): 834–875. doi.org/10.1002/rob.20370.

Cox, M. T.; and Ram, A. 1999. Introspective Multistrategy Learning: On the Construction of Learning Strategies. Artificial Intelligence 112(1–2): 1–55. doi.org/10.1016/S0004-3702(99)00047-8.

Cox, M. T. 2007. Perpetual Self-aware Cognitive Agents. AI magazine 28(1): 32–45. doi.org/10.1609/aimag.v28i1.2027.

Cox, M. T.; Alavi, Z., Dannenhauer, D.; Eyorokon, V.; Munoz-Avila, H.; and Perlis, D. 2016. MIDCA: A Metacognitive, Integrated Dual-Cycle Architecture for Self-Regulated Autonomy. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, 3712-3718. Menlo Park, Calif.: AAAI Press.

Cox, M. T.; and Burstein, M. H. 2008. Case-based Explanations and the Integrated Learning of Demonstrations. Künstliche Intelligenz (Artificial Intelligence) 22(2): 35–38.

Dannenhauer, Z. A.; and Cox, M. T. 2018. Rationale-based Perceptual Monitors. AI Communications 31(2): 197–212. doi.org/10.3233/AIC-180758.

Dannenhauer, D.; Munoz-Avila, H.; and Cox, M. T. 2016. Informed Expectations to Guide GDA Agents in Partially Observable Environments. In Proceedings of the 25th International Joint Conference on Artificial Intelligence, 2493-2499. Palo Alto, CA: AAAI Press.

Dannenhauer, D.; and Munoz-Avila, H. 2015. Raising Expectations in GDA Agents Acting in Dynamic Environments. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, 2241–2247. Palo Alto, CA: AAAI Press.

Ghallab, M.; Nau, D.; and Traverso, P. 2004. Automated Planning: theory and practice. Amsterdam: Elsevier.

Gogineni, V.; Kondrakunta, S.; Molineaux, M.; and Cox, M. T. 2018. Application of Case-based Explanations to Formulate Goals in an Unpredictable Mine Clearance Domain. In Proceedings of the ICCBR-2018 Workshop on Case-Based Reasoning for the Explanation of Intelligent Systems, 42-51. Stockholm, Sweden.: International Conference on Case-Based Reasoning.

Gogineni V.; Kondrakunta S.; Brown D.; Molineaux M.; and Cox M. T. 2019. Probabilistic Selection of Case-Based Explanations in an Underwater Mine Clearance Domain. In Proceedings of the Twenty-Seventh International Conference on Case-Based Reasoning, 110-124. Otzenhausen, Germany.: International Conference on Case-Based Reasoning.

Kondrakunta, S.; and Cox, M.T. 2017. Autonomous Goal Selection Operations for Agent-based Architectures. Paper presented at 2017 IJCAI Goal Reasoning Workshop, Melbourne, Australia, 19-25 August.

Kondrakunta, S.; Gogineni, V.; Molineaux, M.; Munoz-Avila, H.; Oxenham, M.; and Cox, M. T. 2018. Toward Problem Recognition, Explanation and Goal Formulation. Paper presented at 6th Goal Reasoning Workshop at IJCAI/FAIM-2018. Stockholm, Sweden, 13-19 July.

Molineaux, M.; Klenk, M.; and Aha, D. W. 2010. Goal-Driven Autonomy in a Navy Strategy Simulation. In Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, 1548–1554. Menlo Park, Calif.: AAAI Press.

Munoz-Avila, H.; Aha, D. W.; Jaidee, U.; Klenk, M.; and Molineaux, M. 2010. Applying Goal Driven Autonomy to a Team Shooter Game. In Twenty-Third International FLAIRS Conference. Menlo Park, Calif.: AAAI Press.

Ram, A. 1993. Indexing, Elaboration and Refinement: Incremental Learning of Explanatory Cases. Machine Learning 10:201–248.

Schank, R. C. 1986. Explanation Patterns: Understanding Mechanically and Creatively. Hillsdale, NJ: LEA.

Schank, R. C.; Kass, A.; and Riesbeck, C. K. 2014. Inside Case-based Explanation. Hillsdale, NJ: LEA.