Toward Problem Recognition, Explanation and Goal Formulation

Sravya Kondrakunta[1], Venkatsampath Raja Gogineni[1], Matt Molineaux[1], Hector Munoz-Avila[2], Martin Oxenham[3] and Michael T. Cox[1]
kondrakunta.2@wright.edu
[1]Wright State University, [2]Lehigh University, [3]Defence Science & Technology Group
http://www.mcox.org/colab2

Abstract

Goal reasoning agents can solve novel problems by detecting a discrepancy between expectations and observations; generating explanations about plausible causes for the anomaly (i.e., the discrepancy); and formulating goals to remove the cause. This paper considers the broader challenge of discerning the difference between benign anomalies and those that represent an actual problem for an agent. Furthermore, we call into question the very concept of a problem itself. This paper formulates a new problem representation tied to the challenge above. While doing so, this paper discusses the role of explanatory hypotheses and goal formulation under these circumstances and illustrates them via an example in the maritime domain. In support of our ideas, we show the empirical difference between a standard planning agent, agents that detect anomalies, and those that recognize problems.

1 Introduction

An intelligent autonomous agent in a partially observable world should formulate its own goals, make plans to achieve those goals and execute those plans to achieve its mission. [Paisner et al. 2014] states that an agent can formulate its own goals from a discrepancy or anomaly by generating a hypothesis that explains that discrepancy, and generating new goals that respond to that hypothesis in the blocks world domain. However, many discrepancies that arise in the real world do not represents a problem to an agent's mission or activity. For example, the playing of loud music may or may not be a problem for a roommate. If the roommate is preparing for an upcoming exam this is a problem; if, on the other hand, she is doing her laundry, this is not a problem. More generally, an agent does not need to respond to every observed anomaly; they should be capable of distinguishing between those that signal a problem and those that do not.

The *Goal Driven Autonomy (GDA)* [Cox, 2007] [Munoz-Avila et al., 2010] [Molineaux et al., 2010] approach to agency constitutes a sufficient approach for an autonomous agent to respond to discrepancies, but much of the existing research does not formally address the issue of which anomalies are worthy of being considered problems. In this paper, we consider the task of recognizing whether an anomaly should constitute a problem for the agent. Perfoming this task efficiently will improve both the efficiency and robustness of the agent. We use the term *problem* in this paper to refer to discrepancies that require a response in order to meet the agent's objectives. [Cox, 2013] states that GDA is required in order for the agent to not just plan, act and perceive, but also to comprehend so that it is capable of recognizing threats to its current goals, plans and intentions. The author also claims that this will result in autonomous agents which are more sensitive and flexible to the environment and he tries to achieve this flexibility through goal insertion, in which the agent generates and inserts a new goal into its planning process with the help of explanations as outlined in the next section.

In the remainder of this paper, we present a fomalism for the problem recognition task and demonstrate it via an example in an uncertain, dynamic environment. We will also discuss a solution to this task that modifies the goal generation step of a typical GDA agent, via the use of the *Metacognitive Integrated Dual-Cycle Architecture (MIDCA)* and the Meta-AQUA explanatory system.

The paper continues as follows. Section 2 defines the problem recognition task and introduces a running example in the form of mine clearance in the maritime Defense domain. Section 3 describes *explanation patterns (XP)* and their role in understanding problems. Section 4 discusses the implementation of a simulation of the mine clearance operation that involves the use of an autonomous underwater vehicle (AUV). It also details the application of the problem formalism and explanations to identify and respond to problems. This is followed by the evaluation of GDA agents and the presentation of the results in Section 5. Related research is discussed in Section 6. Finally, some concluding remarks are made in Section 7, along with discussion abput some ideas for future research.

2 Problem Recognition

An anomaly occurs when the expected state of the agent does not match its current observed state, but a problem only occurs when the above anomaly needs to be addressed. As such, problem recognition refers to reasoning about the anomaly and deciding whether it is something that the agent needs to deal with. There might be different types of

problems and also different ways to recognize and address them, one such problem being the planning problem. A classical planning domain is defined [Ghallab, et al., 2004] as a finite state-transition system in which each state $s \in S$ is represented by a finite set of ground atoms. A planning operator is a triple $o = (head(o), pre(o), eff(o))$, where *pre(o)* and *eff(o)* are preconditions and effects. Each action, $a \in A$ is a ground instance of some operator o. An action is executable in a state s if $s \vDash pre(a)$.

For a classical planning domain, the state-transition system is a tuple $\Sigma = (S, A, \gamma)$, where S is the set of all states, and A is the set of all actions as above. In addition, γ is a state transition function $\gamma : S \times A \rightarrow S$ that returns the resulting state of an executed action given thes current state i.e., $\gamma(s, a) \rightarrow s'$.

A classical planning problem is a triple $P = (\Sigma, s_0, g)$, where Σ is a state transition system, s_0 is the initial state, and the goal state $g \subset G$ is a conjunction of first-order literals. A state s_g satisfies a goal if $s_g \vDash g$; in this situation we refer to s_g as a goal state. A plan $\pi \in \Pi$ represents a solution to P if it consists of a sequence of plan steps $\langle a_1, a_2, \ldots a_n \rangle$ that incrementally changes the world, starting from the initial state s_0 and ending in a goal state. That is, it is a solution if $\gamma(\ldots\gamma(\gamma(s_0, a_1), a_2)\ldots, a_n) \vDash g$. We say that the state transition function takes the initial state and the plan to achieve the current goal and results in the goal state of the agent when the goal is satisfied. We represent the above as $(\gamma(\pi_{gc}, s_0) = s_g) \vDash g$. It is a more generic way of representing the above solution.

The classical planning definition was extended by Cox [2017] to fit in the GDA context. According to this extension, the planning problem is a 6-tuple, $P_{gda} = (\Sigma, s_c, g_c, s_e, \hat{G}, \Delta)$, where Σ is a state-transition function, s_c is the observed state, g_c is the current goal, s_e is the expected state, $\hat{G}$ is the set of goals of the agent, and Δ is the goal transformation function, which helps the agent to change its current goal to some other goal for various reasons, for example, an insufficient amount of resources being available to achieve the current goal. In this paper, we formulate a new definition of the planning problem by modifying the above definition to suit our needs.

When an agent is working on its current goal, g_c, and a problem occurs, we refer to the existing plan for achieving that goal as π_{gc}. One subsequence of that plan, π_c, has already been executed, and π_r refers to the rest of the existing plan, which has not been executed. These three plans must always satisfy the relationship $\pi_{gc} = \pi_c \,.\, \pi_r$, where '.' denotes the concatenation of the plans.

The current state s_c of the agent from the above two definitions of the plan and the state transition function can be defined as $s_c = \gamma(\pi_c, s_0)$. The expected state s_e of the agent when the current observed and expected states do not match should also be included within the problem definition in order to determine a problem when any discrepancy occurs in the real world.

The history of what the agent has been doing is also important to determine the reason for the cause of the problem. Hence, the history of the agent is defined to be $H_c = (\pi_c, g_c)$.

We also use *background knowledge* Bk, for the purposes of the paper, to determine if a particular anomaly is a problem or not. Therefore, in order to provide the agent with the necessary background information we provide it with a state-transition system Σ as in classical planning and the goal transformation function Δ. The goal transformation system allows the agent to change its current goal to a different goal for various reasons. Adopting the classical definition of Σ and the definition of Δ given in Cox [2017], we represent the background as $Bk = (\Sigma, \Delta)$.

Once an agent acquires all the above information it now needs to explain whether a particular anomaly is a problem or not. Such an explanation χ should aid in determining the problem, as well as aid in generating a new goal (g_n). Explanations are elaborated on in Section 3.

Putting all of these together, we define the planning problem as $P_{gda} = (s_c, B_k, s_e, H_c, \pi_r, \chi, g_n)$, where s_c is the current state of the agent, B_k is the background knowledge of the agent, s_e is the expected state, H_c is the history of the agent, π_r is the remaining plan, χ is the explanation of why the anomaly encountered is a problem for the agent and g_n is the newly generated goal.

2.1 Example Situation – Naval Mine Clearance

In order to illustrate these and other concepts as they are developed in this paper and to assess the performance of the resulting GDA agents, it will be useful to consider them in the context of the following concrete example from the maritime Defense domain.

To prepare a harbor for use during maritime operations, it is essential to conduct mine clearance activities to ensure that ships can operate safely as they transit between the open sea and the port in the harbor. As searching and clearing mines in the entire harbor is likely to be a time-consuming and expensive undertaking, a network of safe shipping lanes is typically established instead to reduce the size of the area within the harbor which needs to demined. Such a system is known as a *Q-route* [Li, 2009].

In our example, we assume that the Q-route consists of a single shipping lane and that both the mine detection and clearance are performed by an AUV which is controlled by a GDA agent. Furthermore, previous reconnaissance of the proposed Q-route identified two areas – green area 1 (GA1) and green area 2 (GA2) – within the Q-route where it is expected the only mines will be. As such, any mines encountered which do not lie within GA1 or GA2 constitute anomalies, but only the anomolous mines within the Q-route are classified as problems because of the hazard that they pose to shipping. It is the role of the agent to determine how to respond to these anomalous mines in each instance.

3 Explanation and Goal Formulation

Whenever some discrepancy between the expectations and the observed state occurs, explanations help the agent understand the discrepancy and formulate new goals [Paisner et al. 2014]. In this work, we use Meta-AQUA to generate explanations and formulate goals in the cognitive architecture known as MIDCA [Cox et al., 2016].

Meta-AQUA [Ram and Cox, 1992] is a story understanding system that tries to explain "*why the actor in the story behaved (performed a certain action) as he did*" using a case-base of abstract explanation patterns, which are engineered by the experts of a specific domain. However, MIDCA is the cognitive architecture that controls the agent. The observations in the world are fed by MIDCA to Meta-AQUA continously in the form of a story, which is not a natural language text but a report with each action followed by the observed world states. If there exists any discrepancy, Meta-AQUA retrieves an abstract XP, adapts it with the world observations and sends it to MIDCA to formulate goals to avoid such discrepancies in future.

3.1 Explanation Pattern

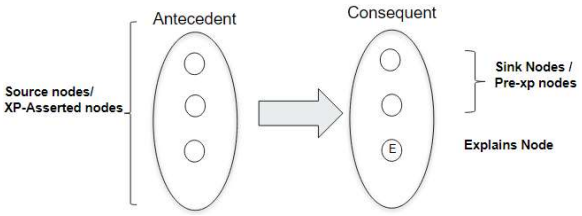

Figure 1: XP Structure

An *explanation pattern (XP)* [Schank, 1986] is a knowledge structure which represents the causal relations between the actions and the states of an agent or an actor in the form of an antecedent causing a consequent as shown in figure 1. A consequent contains the states/actions that are observed in the world while the antecedent may contain the hypothetical states/actions causing the consequent. An action/state is referred to as a node in Meta-AQUA and both are described based on their role in an XP. The actions/states in the consequent comprises an Explains node and the Pre-XP nodes, where an Explains node represents the discrepant action/state, while the Pre-XP nodes represent all other actions/states which are observable and caused by the antecedent. The antecedent contains XP-Asserted nodes/source nodes which are the source actions/states causing the consequent. An XP structure can be as complex as being an XP-Asserted node for another XP structure, in which case the consequent of the XP in the XP-Asserted node is the source cause for the consequent of the other XP. This is further discussed with an example in section 4.1.

3.2 Retrieving an Explanation Pattern

An unexpected observable state/action ($s_c \rightarrow s_c \neq s_e$) when unified with the explains node makes the XP worthy of retrieval and it is only retrieved when Pre-XP nodes are unified with the world observations. However an XP is only applicable if the source nodes are unified and true. If the source nodes are unknown, the agent generates a *knowledge goal* (or *learning goal*) to determine the identity of these sources. However, since an antecedent is responsible for a consequent, a retrievable XP can be reasoned about to formulate new goals g_n for an agent.

3.3 Example of an Engineered Explanation Pattern and its Retrieval

With reference to our mine clearance example, engineering an XP structure can be achieved using the causal chain of actions as shown in the Figure 2 which represents an explanation χ of why a mine is located on the Q-route using a causal chain of actions and states. It hypothesizes that an enemy ship existed and was in the state of being near to the Q-route which enabled it to perform the action of travelling into the Q-route, represented in the figure as ptrans (Physical Transfer). This then led the enemy ship to be at the location of the Q-route and further enabled it to lay a mine in the Q-route which is why there is a mine in the Q-route.

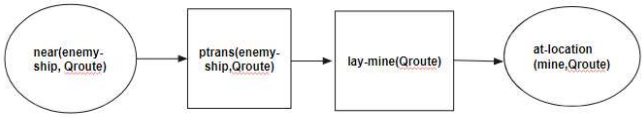

Figure 2: Causal chain of actions and states for explaining the existence of a mine in the Q-route

The observed discrepant state of a mine being at the Q-route constitutes the Explains node (Figure 3, Circle with E) while the observed states of a mine and the actor (AUV) detecting a mine fall into Pre-XP nodes as shown in Figure 3.The XP-Asserted nodes constitute the hypothetical state of the enemy ship being near the Q-route along with the lay-mine action which inherently involves ptrans as the subaction.

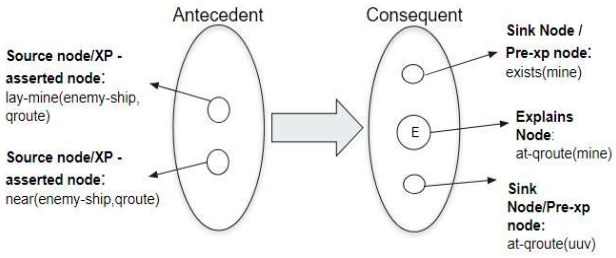

Figure 3: MINE-XP

The MINE-XP is retrieved when an agent detects a mine on the Q-route ($(s_c \rightarrow s_c \neq s_e)$, the discrepant state is unified with the Explains node and it adapts the XP with the specific mine. Later after successful unification, the Pre-XP nodes are unified and adapted to the specific mine and AUV. Finally the MINE-XP is retrieved for use during goal formulation.

3.4 Goal Formulation

An XP structure explicitly says that the XP-Asserted nodes in the antecedent are the root cause for the observed consequent. So eliminating the possibility of actors and objects to perform such actions and/or states lead us to potential goals. These goals avoid such discrepancies from happening in future. MIDCA has a goal removal mapping function that takes in the input as an object or an actor and

outputs a goal that can remove the actor or the object.

For the MINE-XP as shown in the figure 3, the XP-asserted nodes are the lay-mine action and the state of enemy ship near the Q-route. The possible actor for removal is the enemy-ship which is fed to the goal removal mapping function that outputs the goal *apprehend(enemy-ship)*.

4 Naval Mine Clearance Example Revisited

To implement problem recognition and goal formulation in the context of the naval mine clearance example, we developed a simulation using the pMarineViewer simulator in the Mission Oriented Operating Suite (MOOS-IvP[1] [Benjamin et al., 2009]) which is an autonomy middleware solution for controlling autonomous maritime vehicles. Using a method similar to that described in [Wilson et al., 2016] and [Oxenham and Green, 2017], we integrated the cognitive architecture MIDCA with MOOS-IvP, so that control of the AUV was effected by MIDCA rather than by MOOS-IvP itself. This also enabled Meta-AQUA to generate explanations as anomalous mines were encountered.

In the simulation, the default AUV in MOOS (which is representative of a Remus 100 AUV) was assumed to be performing the mine detection and clearance under the control of the MIDCA GDA agent.

Figure 4 illustrates the MOOS-AUV domain. The red cylindrical object in the top left corner represents the Remus 100, the green triangles represent the 50 mines with their respective number, the area between the two parallel lines represents the Q-route, and the octagons on the left and right represent GA1 and GA2 respectively. The mines are randomly scattered throughout the transit and the Q-route.

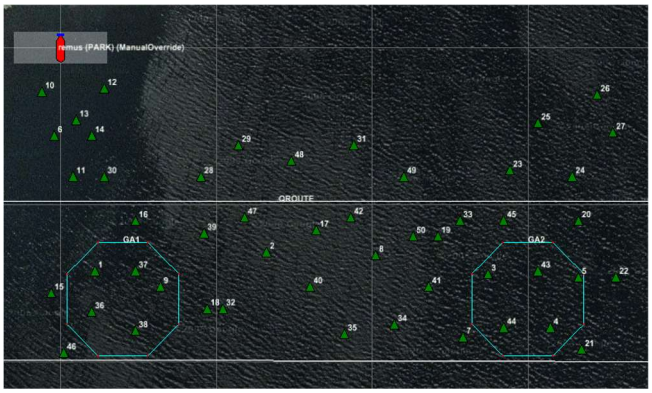

Figure 4: Simulation of the MOOS-AUV domain

In the simulation MIDCA and MOOS-IvP are connected via sockets and they communicate asynchronously. MOOS sends the current location coordinates and speed of the Remus 100 to MIDCA which in turn guides the Remus 100 by sending MOOS the speed of the Remus 100 as well as the coordinates of the location where it is expected to go. MIDCA is connected to Meta-AQUA through sockets synchronously. Meta-AQUA takes the agent's current actions and perceived states from MIDCA and uses them to explain the anomalous state/actions via its case-base of XP's. Each retrieved XP is then passed back to MIDCA, which reasons about the XP and formulates new goals.

4.1 Problem and Explanation in the MOOS-AUV Domain

In this section, our aim is to represent how the problem formalism, explanation and goal formulation defined in earlier sections fit in the MOOS-AUV domain. In the MOOS-AUV domain, the goals of the agent include: (1) clear the mines in GA1, (2) clear the mines in GA2 and (3) head back to its initial position (home). Let us assume that the agent selects the goals in the order they are provided to the agent so the current goal of the agent would be to clear the mines in GA1. Therefore, initially g_c = *cleared mines(remus,ga1).* The expectation of the agent is that mines are present only in GA1. The plan π_{gc} would comprise several steps to achieve its current goal g_c, some of which would be the following:

$\pi_{gc} = \{$
a_1 =*move_to_the_location (remus, home , location-a),*
a_2 =*move_to_the-location (remus, location-a , location-b),*
a_3 =*move_to_the_location (remus, location-b , ga1),*
a_4 =*survey_the_ga1_in (location-c),*
a_5 =*locate_the_mines_in_ga1 (location-c),*
a_6 =*clear_the_mines_in_ga1 (location-c)*$\}$.

Here location-a and location-b are the intermediate locations outside of the Q-route which determine the path that the agent has to follow to reach GA1. Once it reaches GA1, it surveys GA1 and locates/identifies the mines and clears those mines. Location-c is in GA1. The initial state s_0 of the agent is one of being idle/doing nothing. When the current goal of the agent is satisfied then its final state after achieving the goal would be s_g *achieved mine-free ga1*. Let us now assume that the agent has started to work on its current goal and it has completed a_1 and the agent has detected a mine at *location-b*. This would be an anomaly because it has violated the expectation of the agent, in that the expected state s_e here is that there should be no mine on the path. The history $H_c = (\pi_c, g_c)$ leading up to this anomaly is used by Meta-AQUA to determine if it is a problem or not. Since the Pre-XP node of Q-route in the *PROBLEM-XP* pattern (discussed in section 4.2) remains false as the mine is outside of the Q-route, Meta-AQUA does not regard the anomaly as a problem and so does not retrieve any explanation. In summary we would have all of the following in the problem definition, $P_{gda} = (s_c , Bk , s_e , H_c, \pi_r, \chi, g_n)$, where s_c is the state of the agent when it detects the mine at *location-b* and s_e is the expected state that there should not be a mine at *location-b.* H_c consists of the history of the plan that has been executed i.e., $\pi_c = (a_1, a_2)$ and the goal g_c = *cleared mines(remus,ga1)*, in which case $H_c = \{(a_1,$ *cleared mines(remus,ga1),* $(a_2,$ *cleared mines(remus,ga1)*$\}$. The plan π_r contains the remaining steps i.e., $\pi_r = (a_3,...a_6)$. χ is the explanation that the anomaly is not a problem. The goal g_n is the null set and Bk comprises the state transistion system Σ that contains all the state information, all the

[1] https://oceanai.mit.edu/moos-ivp/

actions and their corresponsing state transition function, and Δ consists of the identity transformation since nothing has happened that necessitates changing the goal.

Let us now assume that the goal g_c has been achieved successfully and the agent is traveling towards GA2 to achieve its next goal. Assume that *location-d* and *location-e* are the intermediate locations that the agent passes through to reach GA2 (where both locations lie in the Q-route) and that there is a mine at *location-e*. The history H_c would now also hold all the subsequent changes to the plans and goals which the agent has achieved. The agent would now have a new goal g_c = *cleared_mines(remus,ga2).* The new plan π_{gc} would comprise several steps to achieve this goal g_c similar to the previous goal, some of the steps of which would be:

$\pi_{gc} = \{$
a_1 =*move_to_the_location (remus, ga1 , location-d),*
a_2 =*move_to_the-location (remus, location-d , location-e),*
a_3 =*move_to_the_location (remus, location-e , ga2),*
a_4 =*survey_the_ga2_in (location-f),*
a_5 =*locate_the_mines_in_ga2 (location-f),*
a_6 =*clear_the_mines_in_ga2 (location-f)*}.

Again the initial state s_c of the agent would be *achieved mine-free GA1*, i.e., there are no mines in green area 1. The goal state s_g would be *achieved mine-free GA2* if the current goal is satisfied. The expectations s_e of the agent would be that mines are present only in GA2. The agent encounters a similar anomaly when it reaches *location-e,* but in this case the mine is a problem for the Remus100 because the mine is located in the Q-route and so would endanger any ships traveling along the Q-route. In summary, we would once again have all of the following in the problem definition, $P_{gda} = (s_c , Bk , s_e , H_c, \pi_r, \chi, g_n)$, but in this case s_c is the state of the agent when it detects a mine at *location-e* and s_e is the expected state that there should not be a mine at *location-e*. H_c consists of the history of the plan that has been executed i.e., $\pi_c = (a_1, a_2)$ and also the goal g_c = *cleared mines(remus,ga2),* which can be represented as $H_c = \{(a_1,$ *cleared mines(remus,ga1),...* $(a_1,$ *cleared mines(remus,ga2),* $(a_2,$ *cleared mines(remus,ga2)*$\}$. π_r contains the remaining steps i.e., $\pi_r = (a_3,...a_6)$. χ is the retrieved *PROBLEM-XP* discussed in section 4.2 which explains why the anomaly is a problem for the Remus 100. g_n is the newly formulated goal *cleared mines(remus, qroute)*. Finally, Bk comprises the state transition system Σ that contains all the state information, all the actions and the corresponding state transition function, and the goal transformation function Δ which has transformed the previous goal to the new goal g_n, so that it has become the current goal g_c, while the history preserves the plan at the point where the agent has taken up the new goal so that it can resume the pursuit of the suspended goal at some point in the future.

4.2 Explanation Pattern for the problem

In the second part of the previous example, the information stored in $P_{gda} = (s_c , Bk , s_e , H_c, \pi_r, \chi, g_n)$ infers that the anomaly is a problem because the current state does not match the expected state ($s_c \rightarrow s_c \neq s_e$). The current state s_c of observing a mine on the Q-route by the Remus 100 helps retrieve an *PROBLEM-XP* pattern explaining "why the mine on the Q-route is a problem". However the state relating to the Remus 100 observing a mine as it transits to GA1 does not result in the anomaly being classified a problem because no XP structure is retrieved from the case base.

The *PROBLEM-XP* in Figure 5 comprised of two other XP's *MINE-XP* and *EXPLOSION-XP* explains "Why the Mine on the Q-route is a problem".

MINE-XP provides an explanation of why there is a mine present at a physical location along the lines outlined in section 3.3.

EXPLOSION-XP explains why there is an explosion event; it is because it is expected that the mine and a friendly ship could eventually be at the same location, which would cause the ship to suffer damage. It contains the following nodes.

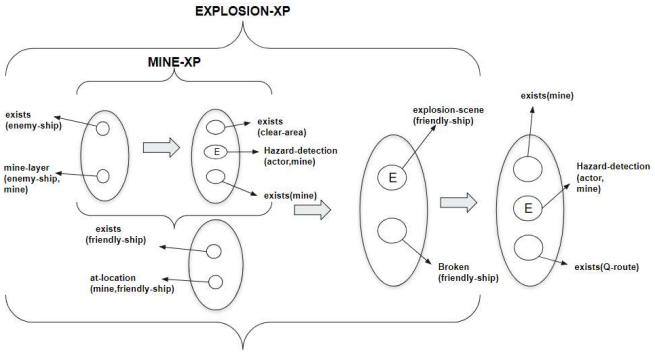

Figure 5: PROBLEM-XP

Explains Node:

- explosion-scene (friendly-ship): an action of a friendly ship being broken into pieces)

Pre-XP Node:

- Broken(friendly-ship): The state of a friendly-ship after being broken into pieces.

XP-Asserted Nodes:

- exists(friendly-ship): A state representing the presence of a Friendly-ship
- at_location(mine,friendly-ship): A state of a mine and a ship being at the same location.
- MINE-XP

PROBLEM-XP explains that the explosion is a problem if there is a mine at Q-route as the friendly ships travel through the Q-route.

Explains Node:

- Hazard-detection(actor,mine): A detection action performed by an actor in recognizing a mine

Pre-XP Node:

- Q-route
- Mine

XP-Asserted Nodes:

- EXPLOSION-XP

The action "Hazard-detection" performed by the MIDCA agent at the Q-route causes the explains node to become true. As the Pre-XP nodes are satisfied, it helps us retrieve the *PROBLEM-XP*, which is reasoned about to obtain the cause of the problem through antecedents.

All the actors and objects from all the antecedents are found by back-chaining through each of the XP-asserted nodes. The actors and objects are fed to the goal removal mapping function which generates the goals to remove the actors and the objects involved. The actors here are the enemy-ship and friendly-ship while the object is the mine. However, removing a friendly-ship is not possible, so the newly generated goals become $g_n = \{$ *cleared_mines(Qroute), apprehend(enemy-ship)* $\}$ and, with the help of $Bk = (\Sigma, \Delta)$, the agent can now transform its current goal to a new goal.

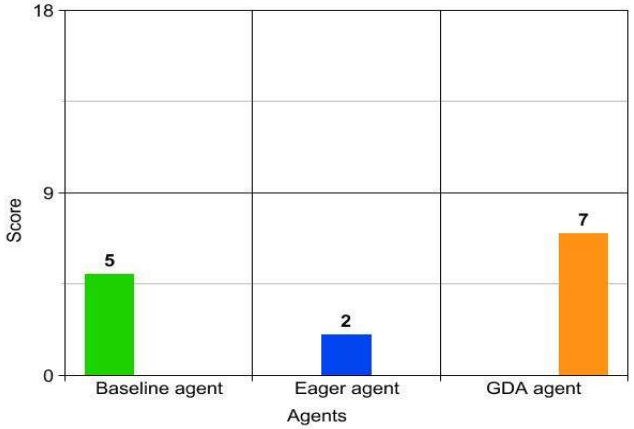

Figure 6: Score obtained by the three agents when the Deadline was 35 Seconds

5 Evaluation of the Implementation in the MOOS-AUV Domain

To perform the evaluation we have introduced two other agents along with our GDA agent, namely an Eager agent and a Baseline agent. Each of the agents is assumed to be set the same task of detecting and clearing all the mines in GA1 and GA2 and that they also all detect the same set of anomalous mines which lie outside of GA1 and GA2. However, they all respond differently to the anomalous mines. The GDA agent detects the anomalous mines but only clears those which are perceived as problems. In contrast, the Eager agent clears all the anomalous mines that it encounters, i.e., it works on each and every anomaly, while the Baseline agent clears only the mines which are inside of the two green areas, i.e., it ignores all the anomalies detected and only performs the tasks assigned to it.

To assess the performance of these three agents we have assigned scores for clearing the mines. For clearing mines outside of the Q-route the agents score 0 per mine. For clearing the mines within GA1 and GA2, the agents score 1 per mine. Finally, for clearing the mines within the Q-route and outside of GA1 and GA2, the agents also score 1 per mine. Finally, all of these scores are added up to see which agent achieves the highest score. The results of these evaluations are presented in the next subsection 5.1.

5.1 Results

In assessing the performance of the three agents, we imposed deadlines ranging from 0 to 80 seconds on the agents in increments of 5 seconds. We begin by discussing the results obtained for three interesting cases corresponding to the deadlines of 35, 70 and 80 seconds, and then summarise the performance across the entire set of deadlines.

Figure 6 presents the scores achieved by the three different agents when the deadline was 35 seconds. The X-axis depicts the agents and the Y-axis indicates the score achieved by each agent. Here the Eager agent achieved a score of 2, the GDA agent achieved a score of 7 and the Baseline agent achieved a score of 5. This is because the Eager agent cleared every mine in its path and wasted a huge amount of time in getting to the Q-route. The GDA agent cleared the mines within the GA1 and some within the Q-route but outside of the green areas, so it achieved the highest score among the three agents. The Baseline agent just performed the actions that it was assigned, so it cleared the mines in GA1 and headed towards GA2. One interesting thing to note while comparing the three agents is that the Eager agent would have performed as well as the GDA agent provided that there had been no anomalous mines in its path. Furthermore, the Eager agent, when provided with a sufficient amount of time, would have reached the score attained by the Baseline agent and with a little more time could have performed better than the Baseline agent provided there were mines outside of the green areas and within the Q-route.

Figure 7 represents the case when the deadline was 70 seconds. Again, the X-axis depicts the agents and the Y-axis indicates the score achieved by each agent. Here the Eager and Baseline agents both scored 10, while the GDA agent scored 15. In this case the Eager agent had enough time to clear the mines within the first green area and also some in the Q-route, whereas the Baseline agent had time to

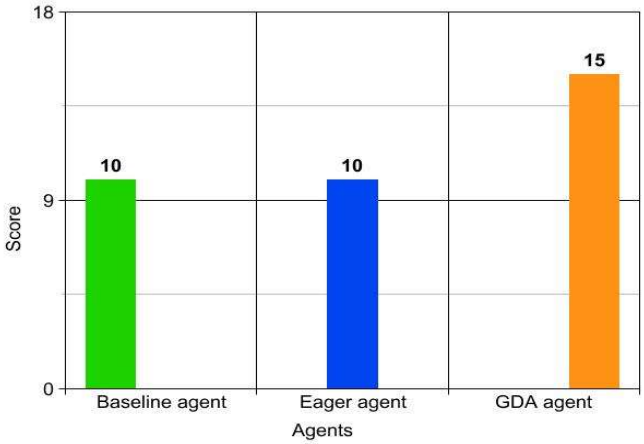

Figure 7: Score obtained by the three agents when the Deadline was 70 Seconds

completely clear the mines in both the green areas and, without clearing any more mines, left for its initial location even when it still had a large amount of time available to clear additional problem mines. Even here the GDA agent outperformed both the other agents because of the advantage that it had by starting to clear problem mines earlier than the Eager agent and because the Baseline agent did not clear any extra mines after achieving its goals.

Figure 8 depicts the score attained by the three agents when the deadline was 80 seconds. As before, the X-axis depicts the agents and the Y-axis indicates the score achieved by each agent. In this case the Eager agent achieved a score of 11, the GDA agent achieved a score of 17 and the Baseline agent achieved a score of 10. In this scenario, the Eager agent performed better than the Baseline agent because the Eager agent cleared every mine which it encountered including the ones within the Q-route and outside of the green areas, whereas the Baseline agent just cleared the mines within the green areas and ignored all the others. Here the GDA agent performed far better than the Eager agent because it started to clear the problem mines way before the Eager agent because the Eager agent was stuck clearing all the anomolous mines, not just the problem ones. Thus, in terms of efficiency, the GDA agent performed better than the Eager agent but they would have performed equally well if it had been important to clear every anomalous mine.

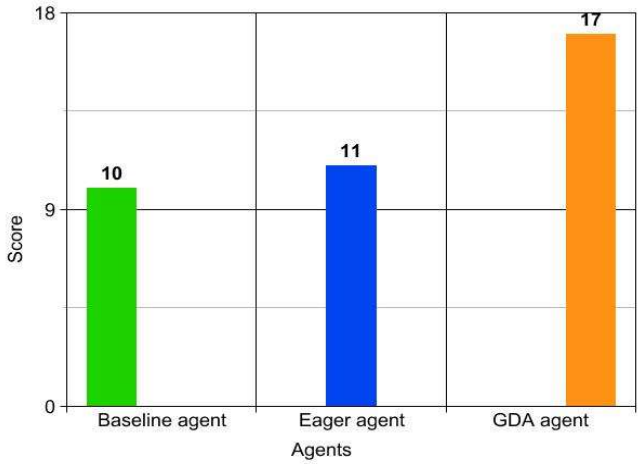

Figure 8: Score obtained by the three agents when the Deadline was 80 Seconds

Figure 9 illustrates the scores achieved by the three agents. Here the X-axis depicts the deadline from 0 seconds to 80 seconds and the Y-axis represents the scores achieved by the agents. The scores are recorded for every 5 seconds. Initially from the deadline of 0 to 25 seconds the GDA agent and the Baseline agent performed similarly because they directly started working on the mines within the green areas and the Q-route, whereas the Eager agent was busily clearing the mines outside the Q-route until that time and thus attained a score of 0. After that from the deadline of 25 seconds to 35 seconds the Baseline agent achieved a constant score as it was travelling from the GA1 to GA2 whereas the GDA agent identified the problems and worked on clearing the mines within the Q-route and outside of the green areas. Thus, the GDA agent increased its score even when travelling from GA1 to GA2, whereas the Eager agent entered the Q-route and started to clear all the mines that it encountered. At the deadline of 60 seconds the Baseline agent cleared all the mines within GA1 and GA2 and then remained idle, so the score achieved by the Baseline agent remained constant, whereas the Eager agent worked on clearing all the mines that it encountered and reached the score of the Baseline agent by the Deadline of 70 seconds. As usual the GDA agent was smart enough to start first on clearing problem mines and so maintained the highest score. Finally at the deadline of 80 the Eager agent performed better than the Baseline agent, whereas the GDA agent maintained its position of achieving the highest score.

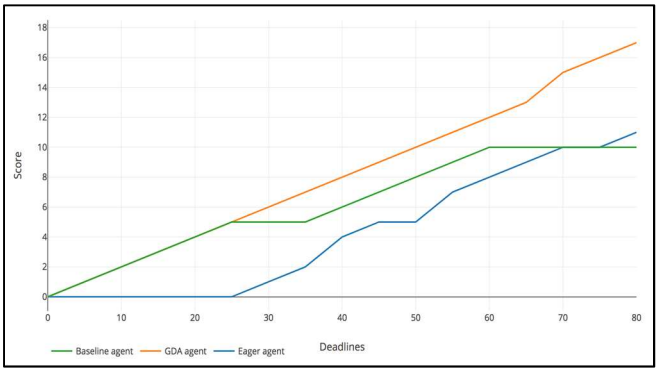

Figure 9: Score obtained by the three agents for the Deadlines varying from 0 Seconds to 80 seconds

6 Related Research

Statistical anomaly detection has been the subject of extensive research because of its applications to a variety of detection tasks such as network intrusion [Kumar, 2005], credit card fraud [Aleskerov et al., 1997], and malignant tumors from MRIs [Spence et al., 2001] among many others [Chandola et al., 2009]. Those works rely on large volumes of data to build statistical models of expected patterns; in that context, anomalies correspond to outlier patterns deviating from expected patterns. In our work, our models are planning models and anomalies correspond to deviations of those models. One of the most challenging problems of statistical anomaly detection is the potentially large number of false positives, which trigger unnecessary alarms. In contrast to those works, in our work, explanations for the discrepancy are generated to ascertain the nature of the anomaly and determine if the agent must deal with it.

The concept of discrepancy detection has played a central role in GDA research. Cox [1997] presents a taxonomy of potential failures an agent might encounter; the taxonomy identifies four categories of failures: domain knowledge, goal, processes and environments. In this work, we are focusing on environmental failures as the discrepancies are the result of the partial observability in the environment. In

Munoz-Avila et al. [2010], it is observed that not all discrepancies require triggering a new goal; in that work, the GDA agent is operating in an adversarial environment with a reward function (i.e., the score of the game). A reward function is also used in Jaidee et al [2011] to use reinforcement learning techniques to learn GDA knowledge. In those works, when the current plan is resulting in a positive reward rate, the agent will ignore discrepancies. In contrast, in our work, we don't assume a reward function; instead, an explanation is generated to determine if the discrepancy is to be ignored.

ARTUE [Molineaux et al 2010] is a GDA system that has been used to provide control in a Naval strategic simulation that is adversarial and partially observable environment. In this work, explanations were generated using a truth maintenance system that identifies plausible worlds that are consistent with the observations made by the agent and trigger a new goal as a result. ARTUE explains all discrepancies, whether problematic or not; goal formulation is responsible for determination of whether the agent should respond. The initial version of ARTUE used rule-based knowledge; extended versions incorporated learning of goal selection knowledge [Powell et al, 2012] and domain-independent motivations [Wilson et al, 2013] responsible for identifying situations that require response. However, these techniques modified the goal formulation process, rather than incorporating a separate problem recognition step prior to explanation generation.

More recently, the notion of GDA agent's expectations has been extended to consider only the necessary effects of the plan executed so far as opposed to considering the whole state [Dannenhauer and Munoz-Avila 2015]. This form of expectations can be used in our work.

Our work is motivated by work on introspective reasoning, where the agent reasons about the decisions that lead to actions taken and how these actions affect the environment. Meta-AQUA [Ram and Cox, 1992] reasons about the processes that lead to a decision which resulted in an discrepancy and considers three types of discrepancies: novel situations, incorrect background knowledge and mix-indexed knowledge structure; the difference between the last two is that in the latter the agent has the knowledge but it is not retrieved in the appropriate circumstances. Fox and Leake [2015] present a mechanism to fix these retrieval mechanisms using introspective reasoning techniques. In our work, we are focusing on novel situations when there is an expectation failure.

7 Conclusion and Future Research

We have described a formalism for agents which enables them to distinguish between those anomalies that they must deal with from those that they do not. The crucial factor in this is the use of *explanation patterns* so that an agent can formulate its own goals to adapt to unexpected events/situations that require the agent's attention.

Real world scenarios often deal with deadlines and it is practically infeasible for an agent to worry about all the anomalies it comes across, reason, react and achieve its prime goals within the given deadline. Although our experiment setting is simulated, adding a deadline to our experiment clearly shows the performance of the GDA agent to be better than the Eager and the Baseline agents.

For future work, we would like to work on several different enhancements that can improve the performance and reasoning capabilities of the GDA agent in our future research. First, adding an importance factor to the problem formalism would help the agent to prioritize anomalies that are classified as a problem with the goals it possesses. Moreover, given that an agent only has finite resources, prioritizing the anomalies could also assist an agent to delegate goals to other agents. Second, adding Goal Monitors [Dannenhauer and Cox, 2018] after formulating goals could help an agent to adapt as the world changes. For example, during mine clearance, if the establishment of a Q-route were to change from one location to another then it is highly likely that it would not be necessary to continue to clear mines along the originally proposed Q-route. Finally, if the number of anomalies flagged were excessive given what might be anticipated in a particular context, then this could serve as a cue for the agent to generate a goal which has broader scope than the current goal. For example, our experimental setting has around ten mines within the proposed Q-route. Instead of clearing just the mines on the agent's path from GA1 to GA2, if the number of mines encountered were too great, then the agent could generate, or delegate to another agent, a goal to survey the entire region between GA1 and GA2.

Acknowledgments

This research was supported by AFOSR grant FA2386-17-1-4063 and by ONR grant N00014-18-1-2009. It is also partially based on research sponsored by AFRL under agreement number FA8650-16-C-6763. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government.

References

[Aleskerov et al., 1997] Emin Aleskerov, Bernd Freisleben, Bharat Rao. Cardwatch: A neural network based database mining system for credit card fraud detection. In Proceedings of the IEEE/IAFE 1997 on Computational Intelligence for Financial Engineering, pages 220-226, New York, USA, March 1997.

[Benjamin et al., 2009] Benjamin, Michael R., John J. Leonard, Henrik Schmidt, and Paul M. Newman. An overview of moos-ivp and a brief users guide to the ivp helm autonomy software. MIT Press.

[Chandola et al., 2009] Varun Chandola, Arindam Banerjee, Vipin Kumar. Anomaly Detection : A Survey. ACM computing surveys (CSUR), 41(3): 15, New York, USA, July 2009.

[Cox, 2017] Michael T. Cox. A model of planning, action, and interpretation with goal reasoning. In Proceedings of the 5th Annual Conference on Advances in Cognitive Systems, pages 57-76, New York, USA, May 2017.

[Cox et al., 2016] Michael T. Cox, Zohreh Alavi, Dustin Dannenhauer, Vahid Eyorokon, Hector Munoz-Avila, Don Perlis. MIDCA: A metacognitive, integrated dual-cycle architecture for self-regulated autonomy. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence,pages 3712-3718, Arizona, USA, February 2016. AAAI Press.

[Cox and Burstein, 2008] Michael T. Cox, Mark H. Burstein. Case-based Explanations and the Integrated Learning of Demonstrations. Künstliche Intelligenz journal, 22(2):35-37, 2008.

[Cox, 2007] Michael T. Cox. Goal-Driven Autonomy and Question-Based Problem Recognition. In Proceedings of the 2nd Annual Conference on Advances in Cognitive Systems, pages 29-45, Maryland, USA, December 2013.

[Cox, 2007] Michael T. Cox. Perpetual Self-Aware Cognitive Agents. AI magazine, 28: 32-35.

[Cox and Ram, 1999] Michael T. Cox, Ashwin Ram. Introspective multistrategy learning: On the construction of learning strategies. Artificial Intelligence Journal, 112(1-2):1-55, August 1999.

[Cox, 1996] Michael T. Cox. Introspective multistrategy learning: Constructing a learning strategy under reasoning failure. PhD thesis, Georgia, Atlanta, 1996.

[Dannenhauer and Cox, 2018] Zohreh A. Dannenhauer, Michael T Cox. Rationale-based perceptual monitors. AI Communications, 31(2):197-212, 2018.

[Dannenhauer and Munoz-Avila, 2015] Dustin Dannenhauer and Hector Munoz-Avila. Raising Expectations in GDA Agents Acting in Dynamic Environments. In Proceedings of the Twenty-fourth international joint conference on Artificial Intelligence, pages 2241-2247, Buenos Aires, Argentina, July 2015.

[Fox and Leake, 1995] Susan Fox and David B. Leake. In Proceedings of the Fourteenth international joint conference on Artificial Intelligence – Volume 1, pages 391-397, Quebec, Canada, August 1995.

[Kumar, 2005] Vipin Kumar. Parallel and distributed computing for cybersecurity. IEEE Distributed Systems Online, 6(10):1-10, November 2005.

[Li, 2009] Pei-Chieh Li. Planning the Optimal Transit for a Ship through a Mapped Minefield. Thesis for a Master of Science in Operations Research, Naval Postgraduate School, Monterey, California, September, 2009.

[Molineaux et al., 2010] Matthew Molineaux, Matthew Klenk, David W. Aha. Goal-Driven Autonomy in a Navy Strategy Simulation. In Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, pages 1548-1554, Atlanta, USA, July 2010.

[Munoz-Avila et al., 2010] Hector Munoz-Avila, David W. Aha, Ulit Jaidee, Matthew Klenk, Matthew Molineaux. Applying goal directed autonomy to a team shooter game. In Proceedings of the Twenty-Third Florida Artificial Intelligence Research Society Conference, pages 465-470, Florida, USA, May 2010. AAAI Press.

[Oxenham and Green, 2017] Martin Oxenham, Ryan Green. From Direct Tasking to Goal-Driven Autonomy for Autonomous Underwater Vehicles. In Proceedings of the Goal Reasoning Workshop at the 2017 International Joint Conference on Artificial Intelligence (IJCAI), Melbourne, Australia, 19-25 August, 2017.

[Paisner et al., 2014] Matthew Paisner, Michael T. Cox, Michael Maynord, Don Perlis. Goal-Driven Autonomy for Cognitive Systems. In Proceedings of the Annual Meeting of the Cognitive Science Society, pages 2085-2090, Quebec City, Canada, July 2014. Cognitive Science Society.

[Powell et al., 2011] Jay Powell, Matthew Molineaux, David W. Aha. Active and Interactive Discovery of Goal Selection Knowledge. In Proceedings of the Twenty-Fourth International Florida Artificial Intelligence Research Society Conference, Florida, USA, May 2011. AAAI Press.

[Ram, 1990] Ashwin Ram. Incremental learning of explanation patterns and their indices. In Proceedings of the seventh international conference on Machine learning, pages 313-320, Texas, USA, 1990. Morgan Kaufmann.

[Ram and Cox, 1992] Ashwin Ram and Michael T. Cox. Introspective reasoning using meta-explanations for multistrategy learning. 1992.

[Roberts et al., 2018] Mark Roberts, Daniel Borrajo, Michael T. Cox, Neil Yorke-Smith. Special issue on goal reasoning. AI Communications, 31(2):115-116, February 2018.

[Schank, 1986] Roger C. Schank. Explanation Patterns: Understanding Mechanically and Creatively. Psychology Press.

[Spence et al., 2001] Charles Spence, Lucas C. Parra, Paul Sajda. Detection, synthesis and compression in mammographic image analysis with a hierarchical image probability model. IEEE Workshop on Mathematical Methods in Biomedical Image Analysis(MMBIA), pages 3-10, Kauai, USA, December 2001. IEEE.

[Wilson et al, 2013] Mark Wilson, Matthew Molineaux, David W. Aha. Domain-Independent Heuristics for Goal Formulation. In Proceedings of the Twenty-Sixth International Florida Artificial Intelligence Research Society Conference, Florida, USA, May 2013. AAAI Press.

[Wilson et al., 2016] Mark A. Wilson, James McMahon, Artur Wolek, David A. Aha, and Brian H. Houston. Toward goal reasoning for autonomous underwater vehicles: Responding to unexpected agents. In Proceedings of the Goal Reasoning Workshop at the 2016 International Joint Conference on Artificial Intelligence (IJCAI), New York, USA, 9-15 July, 2016.