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]Australian DSTG
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 illustrated by an implementation in a naval 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, agents do 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, 2017] [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 what anomalies are worth to be consideredas 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 response in order to meet the agent's objectives. [Cox, 2013] states that the autonomy is capable of recognizing a threat to a current goals, plans and intentions, the author also claims that this would build an autonomy which would be more sentive and flexible to the environment and he tries to achieve the flexibility through goal insertion, considering which we generate a new goal with the help of explanations in the next section.

In the remainder of this paper, we will present a fomalism for problem recognition task and an example in a complex, dynamic domain. We will also discuss a solution to this task that modifies the goal generation step of a typical goal driven autonomy agent, implemented using 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. Section 3 describes *explanation patterns(XP)* and their role in understanding problems. Section 4 discusses the MOOS unmanned underwater vehicle (MOOS-UUV) domain, experiment setting simulated using MOOS along with the application of problem formalism, explanations followed by evaluation and results in section 5. Related research is discussed in section 6, conclusion and future research in section 7.

2 Problem Recognition

An anomaly occurs when the expected state of the agent does not match its current state, but a problem is occcurs when the above anomaly needs to be addressed. So, a problem recognition is reasoning about the anomaly and deciding whether the anomaly is something that the agent needs to work on. There might be different types of problems and also different ways to recognitze and address them, one such problem is 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 a resulting state of an executed action given a 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 the goals of the agent, and Δ is the goal transformation function. 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 achieveing 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 have the relationship $\pi_{gc} = \pi_c . \pi_r$.

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 and expected states doesnot 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 can be defined as, $H_c = (\pi_c, g_c)$, where the H_c is the history of the current goal.

We also use *background knowledge (Bk)*, for the purposes of the paper, to determine if a particular anomaly is its problem or not; so in order to provide the agent with the necessary background information we are providing the agent with the state-transition system (Σ) and the goal transformation function (Δ), the goal transformation systems allows the agent to change its current goal to a different goal for various reasons, we are using the classical Σ and the Δ is defined in Cox [2017]. So, $Bk = (\Sigma, \Delta)$.

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

Putting all of these together, we define the planning problem definition 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 an expected state, H_c is the history of the current goal, and π_r is the remaining plan, χ, is the explanation of why would the anomaly encounter be an agents problem and g_n is the new goal generated.

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.

Meta-AQUA [Ram and Cox, 1992] is a story understanding system that tries to understand "*why the actor in the story behaved (performed a certain action) as he did*" using a case-base of XP's (Explanation Patterns). An *explanation pattern (XP)* [Schank, 1986] is a knowledge structure which represents causal relations between the actions and the states by an agent or an actor. Certain actions cause certain observable actions to take place, being in a certain state can make the actor perform certain observable actions; similarly certain actions can also cause certain observable states. Whenever a discrepancy is worth being detected using the problem formalism $P_{gda} = (s_c, B_k, s_e, H_c, \pi_r, \chi, g_n)$ then an XP (χ) from the case-base is retrieved if there is a match. These XPs help in understanding the problem and generate goals.

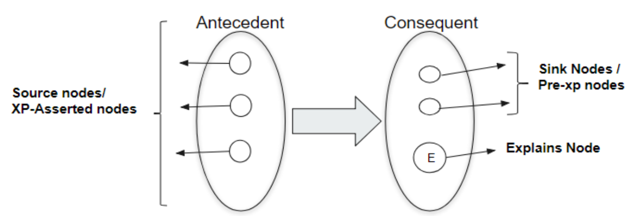

Figure 1: XP Structure

Causal chain of actions and states can be represented as an XP as shown in Figure 1 in the form of an antecedent causing a consequent. A node in Meta-AQUA is a frame definition of a state/action, depending on their placement they are named with different prefixes. The consequent contains Pre-XP nodes and an explains node circle with E that provides frame representations of observable actions/states in the world. Explains node is a frame definition for unexpected observable state/action that needs to be explained, while the Pre-XP nodes are the observable evidences/actions/states caused by the antecedent. The antecedent contains XP-Asserted nodes/source nodes which cause the consequent. An

XP structure can be as complex as being an antecedent to the other XP structure discussed in section 4.1.

An unexpected observable state/action ($s_c \rightarrow s_c \neq s_e$) when matched with the explains node makes the XP worth being retrived and is only retrived when Pre-XP nodes turn true using observed states that are obtained from background knowledge (Bk), However an XP is only applicable if the source nodes are checked and true. If the source nodes are unknown, the agent generates a *knowledge goal* (or *learning goal*) to determine the identity of these sources. Since an antecedent is responsible for a consequent, a retrievable XP can be reasoned to formulate new goals (g_n) for an agent.

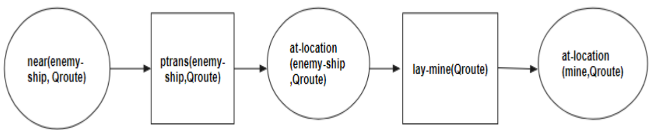

Figure 2: Causal chain of actions and states for explaining the existence of mine in the Q-route

An XP structure can be exemplified using the causal chain of actions as shown in the Figure 2 which represents the explanation (χ) of why a mine is at the Q-route (a transit channel where ships travel) using a causal chain of actions and states. The enemy ship in the state of being near to the Q-route enabled it to perform the action of travelling into the Q-route, represented in the figure as ptrans (Physical Transfer), this lead the enemy ship to be at the location of Q-route and further enabled it to lay a mine at the Q-route which is why there is a mine at the Q-route.

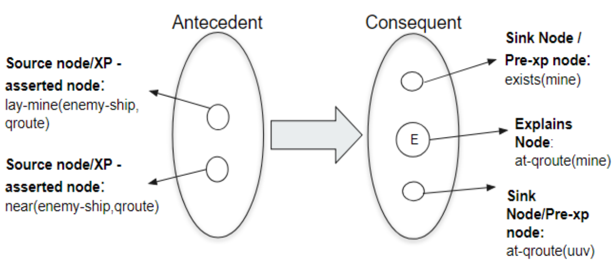

Figure 3: MINE-XP

The discrepancy of Hazard-Detection action by the agent at a certain physical location/Q-route confirms the presence of mine at qroute making the Explains node (Figure 3, Circle with E) to be true and all the Pre-XP nodes i.e *at_qroute(uuv)* and *exists(mine)* are true from the observable world which helps the agent retrieve the MINE-XP pattern, while the whereabouts of the enemy pilot and enemy ship are unknown that can be reasoned from the XP-Asserted nodes from the antecedent helps the GDA agent to formulate the goal to find the whereabouts of the enemy ship and the enemy pilot.

4 Example problem in a complex navel domain

We implemented the problem recognition and goal formulation from the detected problem in the simulated unmanned underwater vehicle (UUV) domain. We have used MOOS-IvP[1] [Newman, 2003] to simulate a Remus100 UUV. The MIDCA [Cox et al., 2016] cognitive architecture controls the Remus100 in the MOOS-IvP simulator. The MIDCA also has an explanation generation system meta-aqua linked to it to generate explanations.

The simulated world in the MOOS contains a Remus100 submarine UUV which would be the agent; a Q-route, which is a safe passage area for the ships to traverse; and 50 mines, whose positions are initially unknown to the agent. There are two areas in the Q-route where mines are expected to be present and those areas are called green area 1 (GA1) and green area 2 (GA2).

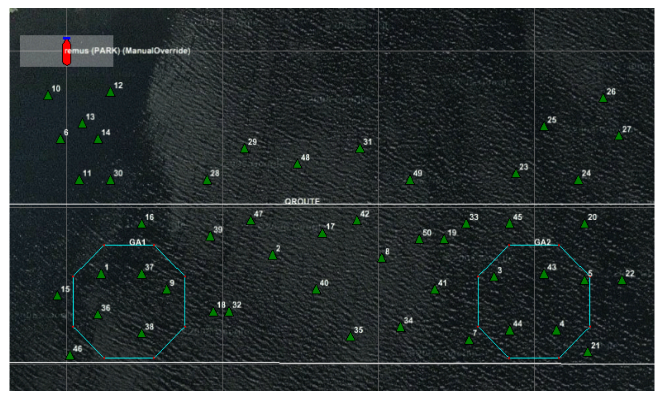

Figure 4: Simulation of the MOOS-UUV domain

Figure 4 shows the MOOS-UUV domain, the red cylindrical object in the top left corner of the image is the Remus 100, green triangles are the mines with their respective number, and the Qroute is the area between two parallel lines and the octagon on the left is GA1 and the octagon on the right is GA2. The mines are placed in a random distribution throughout the transit and the Qroute.

MIDCA and the MOOS-IvP simulation are connected via sockets and they communicate asynchronously. The MOOS will send the current location coordinates and speed of the Remus to the MIDCA and MIDCA guides the Remus by sending the MOOS the coordinates of the location where the vehicle is expected to go and also the speed of the agent. MIDCA is connected to Meta-AQUA through sockets synchronously. Meta-AQUA is an explanation generation module that takes the agents current actions and perceived states from MIDCA to explain an anomalous state/actions from its case-base of XP's and sends it to MIDCA, which reasons about the XP and formulates new goals.

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

4.1 Problem and explanation in the MOOS-UUV domain

In this section, our goal is to represent how the problem formalism, explanation and goal formulation defined in earlier sections fit in the MOOS-UUV domain. In the MOOS-UUV domain, the agents goals 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 are be that the mines are present only in the GA1. The plan (π_{gc}) would be comprising several steps to achieve its current goal (g_c) some of them would be the following:

π_{gc} = {
π_1 =*move_to_the_location (remus, home , location-a),*
π_2 =*move_to_the-location (remus, location-a , location-b),*
π_3 =*move_to_the_location (remus, location-b , ga1),*
π_4 =*survey_the_ga1_in (location-c),*
π_5 =*locate_the_mines_in_ga1 (location-c),*
π_6 =*clear_the_mines_in_ga1 (location-c)*}

Here location-a, location-b are the intermediate locations that the agent has to follow to reach GA1, i.e., these locations determine the path which should be followed by the agent to reach the GA1 and these locations are outside the Q-route, after that it should survey various locations in GA1 and locate/identify the mines and clear those mines. Location-c is in GA1. The initial state (s_0) of the agent is idle/do nothing. When the current goal is satisfied then the final state of the agent after it had achieved the goal would be,(s_g) mine free GA1. Let us now assume that the agent has started to work on its current goal and it has completed the π_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, the expected state (s_e) here would be that mine should not be in the path, $H_c = (\pi_c, g_c)$ has this anomaly is sent to the meta-aqua to determine if it is a problem or not, since the Pre-XP node of Qroute in the *PROBLEM-XP* pattern (discussed in section 4.2) remains false as the mine is outside of the Qroute, meta-aqua does not retrieve any explanation. So, in summary we would have all of the following in the problem definition, $P_{gda} = (s_c, \mathrm{Bk}, s_e, H_c, \pi_r, \chi, g_n)$, s_c is the state of the agent where it detectes mine at *location-b*. s_e is that there should not be mine at *location-b*. H_c contains the history of the plan that has been occurred i.e., $\pi_c = (\pi_1, \pi_2)$ and also the goal g_c = *cleared mines(remus,ga1),* Therefore H_c = {(π_1, *cleared mines(remus,ga1),* (π_2, *cleared mines(remus,ga1)*}. π_r contains the remaining steps i.e., $\pi_r = (\pi_3,...\pi_6)$. χ is the none which determines that it is not a problem. g_n would be a null set and Bk has the state transistion system Σ, that contains all the state information all the actions and their corresponsing state transition function, and the Δ would perform identity transformation i.e., choosing the same goal as before and continues to achieve the remaining part of the plan. So, the above anomaly is not considered a problem.

Let us now assume that the g_c has been achieved successfully and the agent is traveling towards the GA2 to achieve its next goal. Assume that *location-d, location-e* are the intermediate locations to reach the GA2, and *location-e* has a mine in it. The history(H_c) now would be keeping track of all the changes plans and goals which the agent has achieved. The agent will now have a new g_c = *cleared_mines(remus,ga2).* The new plan (π_{gc}) would be comprising several steps to achieve its current goal (g_c) similar to the goal before, some of the steps would be:

π_{gc} = {
π_1 =*move_to_the_location (remus, ga1 , location-d),*
π_2 =*move_to_the-location (remus, location-d , location-e),*
π_3 =*move_to_the_location (remus, location-e , ga2),*
π_4 =*survey_the_ga1_in (location-f),*
π_5 =*locate_the_mines_in_ga1 (location-f),*
π_6 =*clear_the_mines_in_ga1 (location-f)*}

And 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 mine free GA2 if the current goal is satisfied. The expectations(s_e) of the agent would be that the mines are present only in the GA2. The agent encounters a similar anomaly when it reaches the *location-e,* but in this example, the mine at the Q-route is a problem for Remus100 because there might be ships traveling in the Qroute, while the mine during it's transit is not a problem to the agent. So, in summary we would have all of the following in the problem definition, $P_{gda} = (s_c, \mathrm{Bk}, s_e, H_c, \pi_r, \chi, g_n)$, s_c is the state of the agent where it detectes mine at *location-e*. s_e is that there should not be mine at *location-e*. H_c contains the history of the plan that has been occurred i.e., $\pi_c = (\pi_1, \pi_2)$ and also the goal g_c = *cleared mines(remus,ga2),* Therefore H_c = {(π_1, *cleared mines(remus,ga1),...* (π_1, *cleared mines(remus,ga2)*, (π_2, *cleared mines(remus,ga2)*}. π_r contains the remaining steps i.e., $\pi_r = (\pi_3,...\pi_6)$. χ is the retrieved *PROBLEM-XP* discussed in section 4.2 makes it the remus100's problem. g_n would be the new goal formulated, *cleared mines(remus, qroute)*, and Bk has the state transistion system Σ, that contains all the state information all the actions and their corresponsing state transition function, and the Δ would perform goal transformation and chooses the new goal, g_n, to be the current goal, g_c, while the history preserves the plan where the agent has taken up the new goal to come back to the abandoned goal and continue to achieve the remaining plan in future.

4.2 Explanation Pattern for the problem

Given the $P_{gda} = (s_c, \mathrm{Bk}, s_e, H_c, \pi_r, \chi, g_n)$ makes the anamoly worth being considered as problem. The current state (s_c) of observing mine at the Qroute by Remus100 helps retrieve an *PROBLEM-XP* pattern explaining "why the mine at Qroute is a problem", However the state of observing mine at the transit does not retrieve any XP from case-base because no XP structure is retrieved.

The *PROBLEM-XP* in Figure 5 explains "Why Mine at Qroute is a problem", it is comprised of two other XP's *MINE-XP* and *EXPLOSION-XP* which causes the observed state of mine at Q-route to be a problem.

MINE-XP explains why is a mine present at a physical location which is simillar to the one explained in section 3.

EXPLOSION-XP explains why there is an explosion event: because the mine and the Friendly ship are expected to be at the same location, which would caused the ship to be damaged. It contains the following nodes.

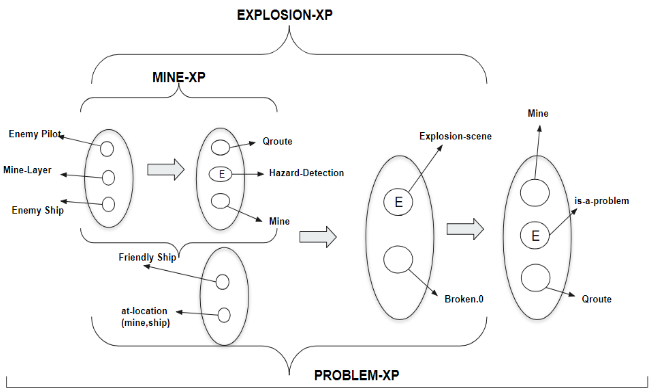

Figure 5: PROBLEM-XP

Explains Node:

- Explosion-scene (frame definition that contains the action of a friendly ship being broken into pieces)

Pre-XP Node:

- Broken.0 (frame definition for broken pieces).

XP-Asserted Nodes:

- Friend Ship (frame definition for existence of the ship)
- at_location (definition for mine and ship to be at the same location")
- MINE-XP

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

Explains Node:

- Is-a-problem (boolean value)

Pre-XP Node:

- Qroute
- Mine

XP-Asserted Nodes:

- EXPLOSION-XP

The anamoly worth being considered a problem makes the explains node (is-a-problem) to be true, as the Pre-XP nodes are satisfied help us retrieve the *PROBLEM-XP*, it is reasoned to obtain the cause of the problem through antecedents and if there exists any unknown information the agent generates the learning goals like did the explosion happen? Will the ship be at the Qroute? .

The antecedents will help the agent determine the possibility of the explosion and thereby formulates a new goal $g_n = cleared\_mines(remus, Qroute)$ and with the help of $Bk = (\Sigma, \Delta)$, the agent can now transform its current goal to a new goal.

5 Evaluation of the implementation in the MOOS-UUV domain

To perform the evaluation we have introduced two other agents along with our GDA agent, those two are the Eager agent and the Baseline agent. The GDA agent detects the mines that are problems and only clears those which are perceived as problems. The Eager agent clears all the mines that it comes across, i.e., it works on each and every anomaly that it comes across. The Baseline agent clears only the mines which are inside of the two green areas, i.e., it ignores all the anomalies detected and would only perform the task assigned to it.

To assess the functioning of these three agents we have assigned scores for clearing the mines, for clearing the mines outside the Qroute the agent would achieve a score of 0. Clearing the mines within GA1 and GA2, for each mine, the agents would obtain a score of 1. Clearing the mines within the Qroute and outside of GA1 and GA2, for each mine, the agents would obtain a score of 2. Finally, all of these scores are added up to see which agent got the highest score. The results of these evaluations have been presented in the next subsection 5.1.

5.1 Results

The results obtained here are for three different deadlines, one is when the deadline is 35 Seconds, the second one is for the deadline of 70 seconds, the third one is for the deadline of 80 seconds and other case is when the deadline is varied from 0 seconds to 80 seconds with a variation of 5 seconds and the scores of all the three agents for different deadlines is observed.

Figure 6 presents the scores achieved by the three different

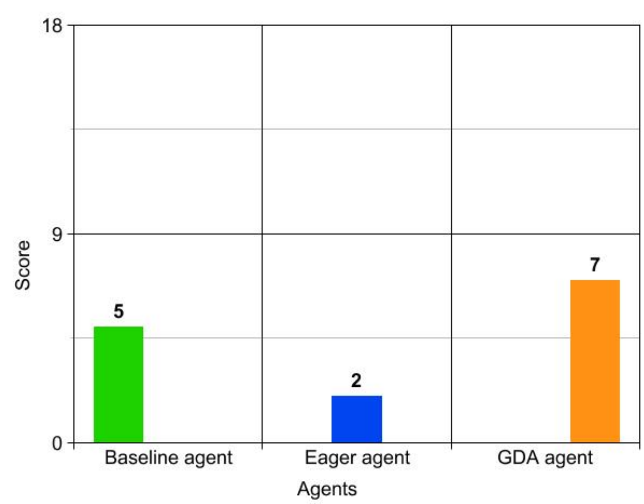

Figure 6: Score obtained by the three agents when the Deadline is 35 Seconds

agents when the deadline is 35 seconds. The X-axis depicts the agents and the Y-axis is the score achieved by the agent, here the eager achieved a score 2, the GDA agent achieved a score 7 and the Baseline agent achieved a score 5. This is because the eager agent clears every mine which is in its path and wastes a huge amount of time to get to the Qroute. The GDA agent clears the mines within the GA1 and some within the Qroute outside of the green areas, so it has achieved the highest score among the three agents. The Baseline agent just performs the actions that it is assigned, so it clears the mines in GA1 and is headed towards the GA2. One interesting thing to note while comparing the three agents is that the Eager agent would perform as good as the GDA agent provided that there are no mines in its path, the Eager agent when provided with sufficient amount of time will reach the score attained by Baseline agent and with a little more time can perform better than baseline agents provided there are mines outside of the green areas and within the Qroute. Note that all of these deductions are true for the world where the mines in the world is static.

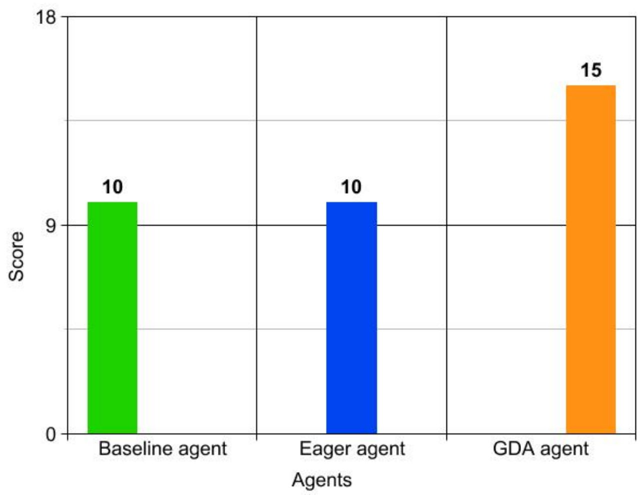

Figure 7: Score obtained by the three agents when the Deadline is 70 Seconds

Figure 7 represents the case when the deadline is 70 seconds. The X-axis depicts the agents and the Y-axis is the score achieved by the agent. Here the Eager agent scored 10, the GDA agent scored 15, and the Baseline agent scored 10. In this case the Eager agent had enough time to clear the mines within the green area and also some in the Qroute, whereas the Baseline agent had time to complete the mines in both the green areas and without clearing any more mines it left for its initial location even when it has a large amount of time to clear other mines. Even here the GDA agent outperforms both the agents because of the advantage that it started early than the eager agent but along with the baseline agent but the baseline agent did not work on clearing any extra mines after achieving its goals, but the GDA agent identified the problems and has worked on those and achieved the highest score.

Figure 8 depicts the score attained by three agents when the deadline is 80 seconds. The X-axis depicts the agents and the Y-axis is the score achieved by the agent. In this case the Eager agent achieved a score 11, the GDA agent achieved 17 points and the baseline agent achieved 10 points. In this

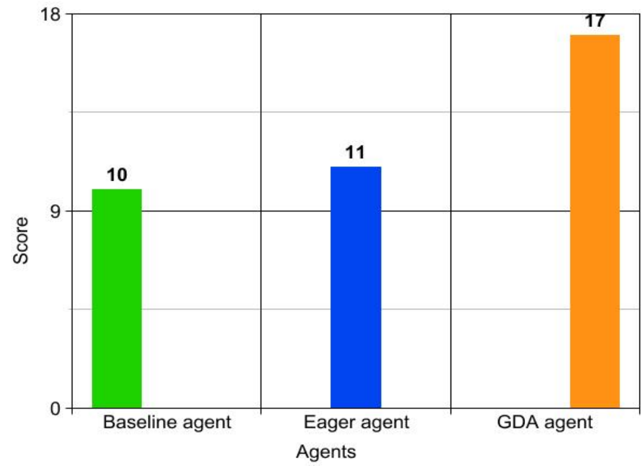

Figure 8: Score obtained by the three agents when the Deadline is 80 Seconds

scenario, the Eager agent performed better than the baseline agent because of the fact that the eager agent clears every mine which it encounters so that also includes the ones within the Qroute and outside of the green areas, whereas the baseline agent just clears the mines within the green areas and remains idle.Here the GDA agent performd far better than the eager agent because it started to clear the mines which are important way before than the eager agent while it was struck with clearing all the unnecessary ones. So, in terms of the efficiency, the GDA agent is better than the Eager agent but they will coincide when clearing every mine would be important for the mission.

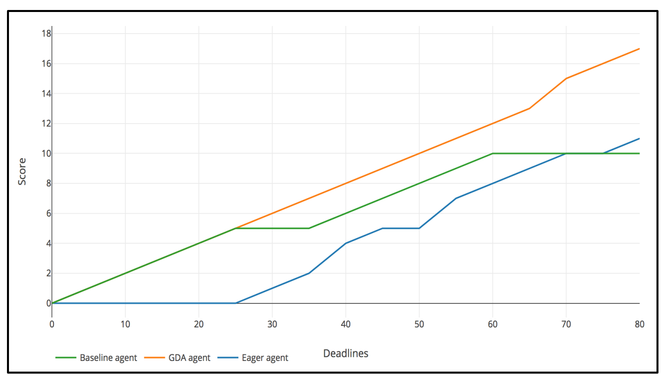

Figure 9: Score obtained by the three agents for the deadlines varying from 0 Seconds to 80 seconds

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. Initiallly from the deadline of 0 to 25 seconds the GDA agent and the Baseline agent performed similarly because they directly start working on the mines within the green areas and the Qroute, whereas the Eager agent was busily clearing the mines outside the Qroute until that time and it has attained a score of 0. After that the 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 has identified the problems and was working on clearing the mines within the Qroute and outside of green areas, so the GDA agent was increasing it score even when travelling from GA1 to GA2, whereas the Eager agent has entered the Qroute and has started to clear all the mines that it encounters. At the deadline of 60 seconds the Baseline agent has cleared all the mines within the GA1 and GA2 and remained idle, so the score achieved by the Baseline agent remained constant, whereas the Eager agent was working on clearing all the mines that it has encountered and has reached the score of Baseline agent by the Deadline of 70 seconds, and as usual the GDA agent was smart enough to start first and has maintained the highest score by clearing the mines that are problems. 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.

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 described a formalism for agents capable of distinguish between those anomalies that the agent must deal with from those that do not. The crucial point is the use of *explanation patterns* so that an agent can formulate its own goals to be adaptable for unexpected events/situations that require the agent's attention.

Real world scenarios often deal with deadlines and it is practically not feasible for an agent to worry about all the anomalies it comes across, reason, react and achieve its prime goals within a 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 agent.

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 can help an agent prioritize anomalies that are detected as a problem with the goals it possesses. Moreover, given an agents constraints prioritizing can help an agent delegate goals to other agents. Second, adding Goal Monitors [Dannenhauer and Cox, 2018] after formulating goals can help an agent adapt to the world changes. For example, in our experimental setting if the Qroute changes from one location to other then we may not need to clear mines because ships no longer travel at the old Qroute. Finally, numerical occurrence of similar problems greater than a certain threshold can help generate an agent a goal whose scope is greater than an individual goal formulated by understanding a single problem. For example, our experiment setting has around ten mines in the Qroute, instead of clearing just the mines on the agents' path from GA1 to GA2, the agent can generate or delegate 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.

[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, 2013] 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, 1996] Michael T. Cox. Introspective multistrategy learning: Constructing a learning strategy under reasoning failure. PhD thesis, Georgia, Atlanta, 1996.

[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 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.

[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.

[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.

[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.