2021

Goal Management in Multi-agent Systems

Venkatsampath Raja Gogineni
Wright State University

GOAL MANAGEMENT IN MULTI-AGENT SYSTEMS

A Dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

By

VENKATSAMPATH RAJA GOGINENI
M.S., Wright State University, 2017
B.Tech., Koneru Lakshmaiah University, 2015

2021
Wright State University

WRIGHT STATE UNIVERSITY
GRADUATE SCHOOL

January 7, 2022

I HEREBY RECOMMEND THAT THE DISSERTATION PREPARED UNDER MY SUPERVISION BY Venkatsampath Raja Gogineni ENTITLED Goal Management in Multi-agent Systems BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Doctor of Philosophy.

Dr. Michael T. Cox, Ph.D.
Dissertation Co-Director

Dr. Mateen M. Rizki, Ph.D.
Dissertation Co-Director

Dr. Yong Pei, Ph.D.
Computer Science & Engineering PhD Program Director

Barry Milligan, Ph.D.
Vice Provost for Academic Affairs
Dean of the Graduate School

Committee on
Final Examination

Dr. Mateen M. Rizki

Dr. Michael T. Cox

Dr. Matthew Mollineaux

Dr. Michael Raymer

Dr. Tanvi Banerjee

ABSTRACT

Gogineni, Venkatsampath Raja. P.h.D., Department of Computer Science & Engineering, Wright State University, 2021. *Goal Management in Multi-agent Systems.*

Autonomous agents in a multi-agent system coordinate to achieve their goals. However, in a partially observable world, current multi-agent systems are often less effective in achieving their goals. In much part, this limitation is due to an agent's lack of reasoning about other agents and their mental states. Another factor is the agent's inability to share required knowledge with other agents and the lack of explanations in justifying the reasons behind the goal. This research addresses these problems by presenting a general approach for agent goal management in unexpected situations. In this approach, an agent applies three main concepts: goal reasoning - to determine what goals to pursue and share; theory of mind - to select an agent(s) for goal delegation; explanation - to justify to the selected agent(s) the reasons behind the delegated goal.

Our approach presents several algorithms required for goal management in multi-agent systems. We demonstrate that these algorithms will help agents in a multi-agent context better manage their goals and improve their performance. In addition, we evaluate the performance of our multi-agent system in a marine life survey domain and a rover domain. Finally, we compare our work to different multi-agent systems and present empirical results that support our claim.

Contents

1 Introduction **1**
- 1.1 Motivation . . . 3
- 1.2 Contributions . . . 4
- 1.3 Assumptions . . . 7
- 1.4 Organization of the Dissertation . . . 7

2 Multi-agent Goal Management **9**
- 2.1 Research Problem and Example . . . 9
- 2.2 Technical Approach . . . 11
- 2.3 Memory Organization . . . 13
  - 2.3.1 Knowledge Acquisition . . . 15
  - 2.3.2 Storage and Retrieval . . . 16
- 2.4 Agent Coordination . . . 18
  - 2.4.1 Multi-agent Goal Reasoning . . . 18
    - 2.4.1.1 Goal Delegation . . . 19
    - 2.4.1.2 Goal Acceptance and Goal Rejection . . . 22
    - 2.4.1.3 Goal Sharing . . . 23
  - 2.4.2 Theory of Mind . . . 24
    - 2.4.2.1 Agent Selection . . . 24
    - 2.4.2.2 Knowledge Sharing . . . 26
  - 2.4.3 Explanation . . . 28
- 2.5 Conclusion . . . 31

3 Experimental Design and Evaluation **33**
- 3.1 Comparison with Different Multi-agent Systems . . . 34
- 3.2 The Marine Life Survey Domain . . . 36
  - 3.2.1 Experimental Design: Three agent scenario . . . 39
    - 3.2.1.1 Empirical Results . . . 40
    - 3.2.1.2 Statistical Significance . . . 45
  - 3.2.2 Experimental Design: Four agent scenario . . . 47
    - 3.2.2.1 Empirical Results . . . 48
    - 3.2.2.2 Statistical Significance . . . 53

3.3 The Rovers Domain 55
3.3.1 Experimental Design 56
3.3.2 Empirical Results 57

4 Related Research **60**
4.1 Multi-agent Systems 60
4.2 Theory of Mind 62
4.3 Goal Reasoning 63

5 Conclusions and Future Research **64**

References **66**

List of Figures

1.1 Unmanned maritime systems 4

2.1 Simulation of the underwater mine clearance domain in Moos IvP 10
2.2 Research approach 12
2.3 Memory classification 17
2.4 Goal delegation scenario in underwater mine clearance domain 21
2.5 HGN representation of the goal to clear mines in GA2. 23
2.6 Agent selection scenario in underwater mine clearance domain 26
2.7 Explanation pattern structure 29
2.8 Explanation pattern structure for justifying goals 30
2.9 Explanation scenario in underwater mine clearance domain 32

3.1 Gray's Reef National Marine Sanctuary and simulation of Marine life survey domain 38
3.2 Three agent scenario in the marine life survey domain 40
3.3 Percentage of goals achieved in the marine life survey domain (three-agent scenario) as a function of time 41
3.4 Percentage of hot-spots identified in the marine life survey domain (three-agent scenario) as a function of time 43
3.5 Percentage of goals achieved in the marine life survey domain (three-agent scenario) as a function of anomalies 44
3.6 Four agent scenario in the marine life survey domain 47
3.7 Percentage of goals achieved in the marine life survey domain (four-agent scenario) as a function of time 49
3.8 Percentage of hot-spots identified in the marine life survey domain (four-agent scenario) as a function of time 50
3.9 Percentage of goals achieved in the marine life survey domain (four-agent scenario) as a function of anomalies 52
3.10 Mars exploration rover 55
3.11 Agents distribution in the rovers domain 57
3.12 Percentage of goals achieved in the rover domain as a function of time . . . 58

List of Tables

2.1 Algorithm to recognize goal delegation 20
2.2 Algorithm to perform goal acceptance and goal rejection 22
2.3 Algorithm to perform agent selection 25
2.4 Knowledge sharing algorithm 27

3.1 Statistical distribution of agents' performance in the three-agent scenario of the marine life survey domain 45
3.2 Two-sample independent t-test in the three-agent scenario of the marine life survey domain 46
3.3 Statistical distribution of agents' performance in the four-agent scenario of the marine life survey domain 53
3.4 Two-sample independent t-test in the four-agent scenario of the marine life survey domain 54

Acknowledgments

I want to take this opportunity to express my gratitude to my advisor, **Dr. Michael Cox**, for playing a significant role in shaping this research. I cherish his enthusiasm and ideas, which significantly influenced me and motivated me throughout my Ph.D. program.

I would also like to thank my committee members for their immeasurable guidance and support. First, I want to thank **Dr. Matthew Molineaux** for guiding me in my research; his immense knowledge, insightful comments, and suggestions helped my research in every possible way. Second, I express my sincere gratitude to **Dr. Tanvi Banerjee**, who introduced me to research and encouraged me in my endeavors. Finally, **Dr. Michael Raymer** and **Dr. Mateen Rizki** served as wise committee members, and their suggestions are invaluable.

Apart from my committee, I am also grateful to **Dr. Pratik Parikh**, Professor in the Department of Industrial Engineering at University of Louisville, for motivating me to pursue Ph.D. in computer science.

Getting through graduate school required more than academic support, and I have my family to thank for listening to and, at times, having to tolerate me over the past six years. First, I thank my parents, **Butchi Babu Gogineni** and **Bhavani Gogineni**, for their support and patience. Second, I want to express my sincere thanks to my sister, **Dr. Monica Gogineni**, for standing by me at all times. Her love and enthusiasm are immeasurable. Finally, I express my deep gratitude to my wife, **Dr. Sravya Kondrakunta**, without whom this dissertation would only be a dream. I thank her for motivating me and helping me achieve my dreams.

This research is supported by National Science Foundation under grant 1849131, Air Force Office of Scientific Research under grant FA2386-17-1-4063, and the Office of Naval Research under grant N00014-18-1-2009.

I dedicate this thesis to three women of my life;

my grandmother, **Nirmala Valluripalli**, my mother, **Bhavani Gogineni** and my wife,

Sravya Kondrakunta for their endless love and encouragement.

Introduction

Humans work effectively in teams. They reason about people, share their knowledge and cooperate to achieve goals simultaneously. Similarly, autonomous agents in a multi-agent environment should work with each other to improve their efficiency and effectiveness. For our purposes, we define a multi-agent system as a combination of autonomous agents in an environment. Often, multi-agent systems function much better than a single agent system because of their ability to achieve assigned tasks. Besides task achievement, multi-agent systems can also share and delegate goals among themselves to overcome problems or quickly achieve existing goals. In a dynamic world, several problems can occur concurrently thus requiring autonomous agents to generate goals in response. However, given limited resources, a single agent cannot always respond to new problems and still achieve its current goals. Whereas in a multi-agent system, agents can delegate goals to others. To effectively delegate their goals, agents must reason about other agents' knowledge, select an agent to which the goal will be delegated, and share knowledge about the problem with the other agent. Acquiring such experiences improves the agents' own knowledge about other agents, which helps the agent better delegate goals in the future.

Answering when, what, whom and how to coordinate is central to understanding this research problem. **When** does an agent need to coordinate? In a partially observable multi-agent environment, several anomalous events might occur. These anomalous events might hinder the ability of an agent or its mission. Furthermore, in the real-world agents often have limited resources to respond to these anomalous events. Therefore, to achieve

an agent's mission with the limited resources available, it should coordinate with other agents to delegate its goals. **What** should an agent coordinate? When an agent does not have enough resources to achieve its mission. It must communicate the goals that achieve its mission effectively with other agents. Therefore, an agent must coordinate the goals it wants to delegate to other agents. **Whom** should an agent coordinate with? An agent should coordinate with a potential candidate that can successfully achieve its delegated goals at a minimum possible cost. **How** should an agent coordinate? An agent should coordinate in such a way that the other agent will understand the importance of the delegated goal and thus pursue it. Furthermore, it must minimize the amount of resources required to coordinate between the agents in the system. Therefore, an agent must explain the motivations behind the goal while minimizing its resources.

Much of the work in the multi-agent systems literature focuses on a centralized network of agents [41, 57]. The focus of this proposed research, however, will be to work toward a decentralized multi-agent system and to develop a communication framework between the agents to better manage their goals with little human intervention. The communication framework must assist the agents in coordinating with other agents that are not necessarily designed to work together.

Problem statement: How can a set of agents manage individual and shared goals in a dynamic world?

Autonomous agents must be able to function effectively even with the introduction of new agents into the world. The agents must be able to learn quickly about the new agent and must be able to share and delegate goals effectively to the newer agent with little human intervention. This proposed research focuses on developing a solution to the problem using the following three concepts:

- **Multi-agent Goal Reasoning:** helps an agent to determine what goals to share and pursue among all its goals. This choice helps an agent conserve its resources and improve its performance in the real world.

- **Theory of mind:** helps an agent infer other agents' beliefs, goals, and intentions. This capability gives an agent the ability to reason about sharing its goals with the right agent(s).

- **Explanation:** helps other agents understand the justification behind a shared goal request. This supports a rational negotiation of the request in a joint decision.

The hypothesis we put forth is that theory of mind and explanations are essential for effective goal management in multi-agent systems.

This research, with the help of the concepts above, will develop a computational framework which can be used in multi-agent systems research. The work will also make use of an existing cognitive architecture for implementation purposes. The next section will focus on the motivation behind the research idea.

1.1 Motivation

Current technology encourages different organizations to build multiple autonomous agents for different overlapping purposes. A need exists for these systems to interact and improve performance in an open ended world. An agent with the capability to share knowledge and goals with other agents allows it to improve its own adaptability, reliability, and reasoning in unforeseen situations. Consider figure 1.1 below. It shows different classifications of unmanned maritime vehicles as classified by the US Department of Defense. Some of them are for surveillance purposes, whereas others are for specific tasks like mine identification and clearance.

This research's intention to understand and improve the efficacy of multi-agent systems is motivated by a vision of using heterogeneous vehicles for different purposes within a common area. For the application of mine counter measures, their coordination will prove to be a asset to their missions, and it will help agents to better survey and identify mines

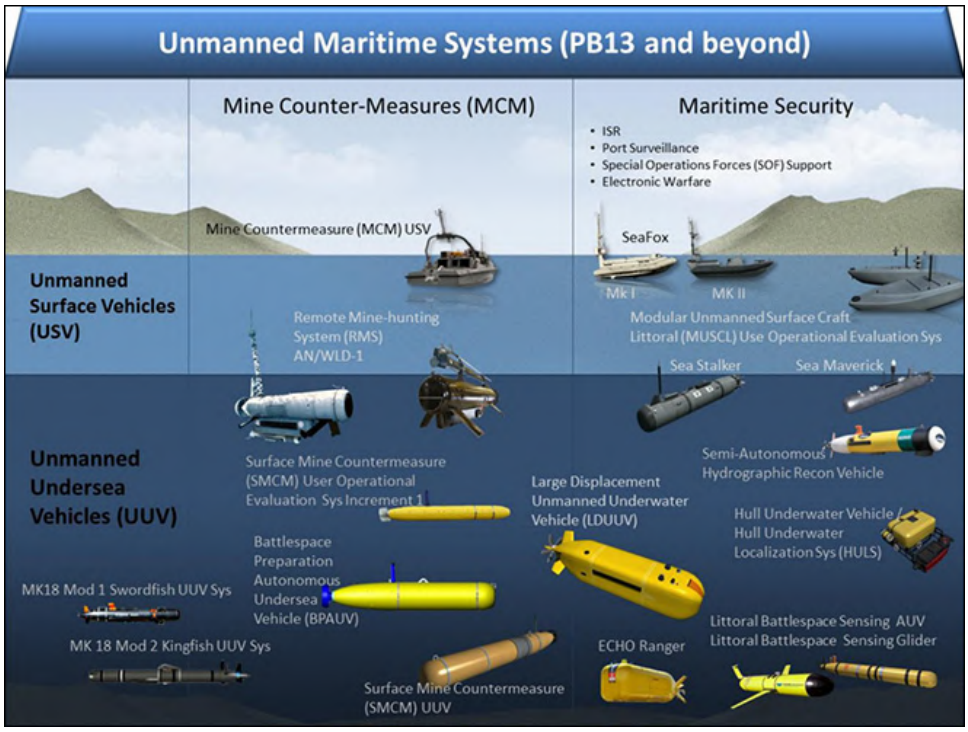

Figure 1.1: Classifications of Unmanned Maritime Systems, from U.S. Department of Defense [18]

within a specified region.

1.2 Contributions

Goal Driven Autonomy (GDA) [7, 8, 14, 15, 43, 44, 45] is an appropriate approach for autonomous agents to handle unexpected situations in a partially observable dynamic world. It reasons about the unexpected events encountered, explains the cause of the events, and performs operations on goals as a response [12]. In the event of a discrepancy (i.e., unexpected event), an agent can pursue different operations on goals: goal formulation, goal selection, goal change, goal delegation, goal monitoring, and goal achievement. The descriptions of these goal operations are as follows:

- **Goal formulation** (δ^*)**:** Whenever an agent encounters an unexpected event that poses a threat to the agent, it can choose to formulate a new goal to respond to the

problem.

- **Goal selection** (δ^{se})**:** An agent performs goal selection to select an active goal from the list of all its goals to pursue.

- **Goal change** (δ^{Δ})**:** An agent performs a goal change operation to change its goal to a similar goal.

- **Goal delegation** (δ^{de})**:** In unexpected situations, an agent can give its goal to other agents in the environment.

- **Goal monitoring:** An agent monitors its goals to see if it is still valuable and valid to pursue its goals.

- **Goal achievement:** An agent plans and executes its actions to achieve its selected goals.

While goal delegation is the significant contribution of our research, we leverage the concepts of GDA and the implementation of above goal operations through the MIDCA [11] cognitive architecture. Below is the specific list of contributions of this research,

1. We describe a novel theory of mind approach for goal delegation operations in distributed multi-agent systems (see Chapter 2). This approach will help autonomous agents delegate their goals to potential agents and identify required knowledge to share with the receiving agent.

2. We introduce a new algorithm, *DetectDelegation*, to recognize the need for the goal delegation operation in partially observable, dynamic environments. DetectDelegation algorithm is a general solution to determine the selection of delegated goals when there is the need for goal delegation (see Section 2.4.1.1).

3. We introduce the novel theory of mind methods, *AgentSelection* and *KnowledgeSharing*, to help an autonomous agent in a distributed multi-agent system choose potential

agents to delegate its goals and share the knowledge required to achieve its goals successfully. AgentSelection algorithm uses the delegating agent's knowledge about all other agents in a dynamic environment to select one or more potential agent(s) to delegate its goals (see Section 2.4.2.1). The KnowledgeSharing algorithm uses the delegating agent's knowledge about other agents to intelligently identify the required knowledge to share with the selected potential agent(s) (see Section 2.4.2.2).

4. We present an explanation framework to explain the motivations behind the goals to the receiving agent. Such motivations will help the agent understand the priority behind the delegated goals (see Section 2.4.3).

5. We present a general method, *GoalAcceptReject*, to help the receiving agents in a multi-agent system make an informed decision on whether to accept or reject the delegated goals (see Section 2.4.1.2).

6. We present experiments and empirical results demonstrating the performance improvements for DetectDelegation, AgentSelection, GoalAcceptReject, KnowledgeSharing, and Explanations in distributed multi-agent systems (see Chapters 3).

In this dissertation, we use experimental evidence to defend the following claims about the quality of the solutions found by the new techniques DetectDelegation, AgentSelection, GoalAcceptReject, KnowledgeSharing, and Explanations.

Claim 1: Goal delegation using a theory of mind approach causes the goal achievement performance (i.e., percentage of goals achieved successfully) in a distributed, multi-agent context to significantly improve relative to a traditional goal reasoning multi-agent system which does not delegate goals.

Claim 2: Explanations helps the receiving agent understand the priority behind the delegated goals, thereby improving the performance of goal achievement in a multi-agent system compared to a multi-agent goal reasoning system with only goal delegation.

1.3 Assumptions

With the current technology trends moving towards distributed nature of multi-agent systems, we believe that the following assumptions are reasonable and will enable us to focus on our research topic.

1. We assume that the operating multi-agent system is autonomous and distributed in nature. Individual autonomous agents in the system have their own goals to achieve, and without any unexpected situations, they all can achieve them.

2. We assume that every autonomous agent in a distributed multi-agent system has a communication framework like Knowledge Query Markup Language for communicating goals and information with other agents in the environment.

3. We assume that individual autonomous agents in a distributed multi-agent system know other agents' capabilities in the world (domain knowledge). However, they lack knowledge about other agents' current beliefs, states, and experiences.

1.4 Organization of the Dissertation

The remainder of this dissertation is organized as follows. Chapter 2 presents the concept of our research and introduces the technical approach to the research problem in detail. It begins by explaining the memory organization with aspects of storage, retrieval, and knowledge acquisition. The chapter then provides solutions to the question of When, what, whom, and how to delegate goals using the concepts of the theory of mind, goal reasoning, and explanations.

Chapter 3 describes the Marine Survey domain and the Rover domain used in this research. This chapter will describe the agent's capabilities, goals, and the unexpected events they might encounter in the respective domains. Furthermore, it also presents the experimental design and empirical results across the respective domains. It begins by introducing

different multi-agent systems to compare our approach. The chapter then demonstrates domain descriptions, experimental designs, statistical significance tests, and empirical results across the domains to test our hypothesis.

Chapter 4 presents the relevant work in related research fields. First, this chapter discusses the work in multi-agent systems and then moves on to the relevant research in the theory of mind. Finally, it discusses the work in goal reasoning research and the work related to explanations.

Chapter 5 reviews our claims and contributions and presents possible planned opportunities for future work.

Multi-agent Goal Management

Autonomous agents pursue goals that focus and direct their actions. However, when an agent recognizes that it cannot achieve all of them, it should delegate some of its goals to other agents. Our research focuses on how agents effectively coordinate with one another to improve the functionality of the system. In this chapter, we discuss the research problem in detail with an example followed by a technical approach to solve the problem. Furthermore, we discuss each component of the approach and finally walk-through the components using the example.

2.1 Research Problem and Example

Most of the current multi-agent systems [19, 21, 32, 55] take agents for granted in an environment. They often assume that the agents are either cooperative or competitive in solving problems (i.e., achieving goals). Cooperative agents (e.g., [31]) are designed to achieve their common goals. Although there is some disagreement about the definition of competitive agents, we can agree that they act selfishly to achieve their own goals. Such characteristics make coordination difficult when they are not designed to work together. Moreover, much of the multi-agent work deals with multiple agents that cooperatively search through the problem space in coming up with a plan to solve a problem [59, 58]. This approach requires combined effort and resources from all the agents to plan for a common goal. However, this is not ideal for every task. So, our research problem is to determine how an

autonomous agent in a multi-agent environment can coordinate with any agent to manage its goals while conserving resources. In this section, we demonstrate the research problem through an example in a complex naval domain.

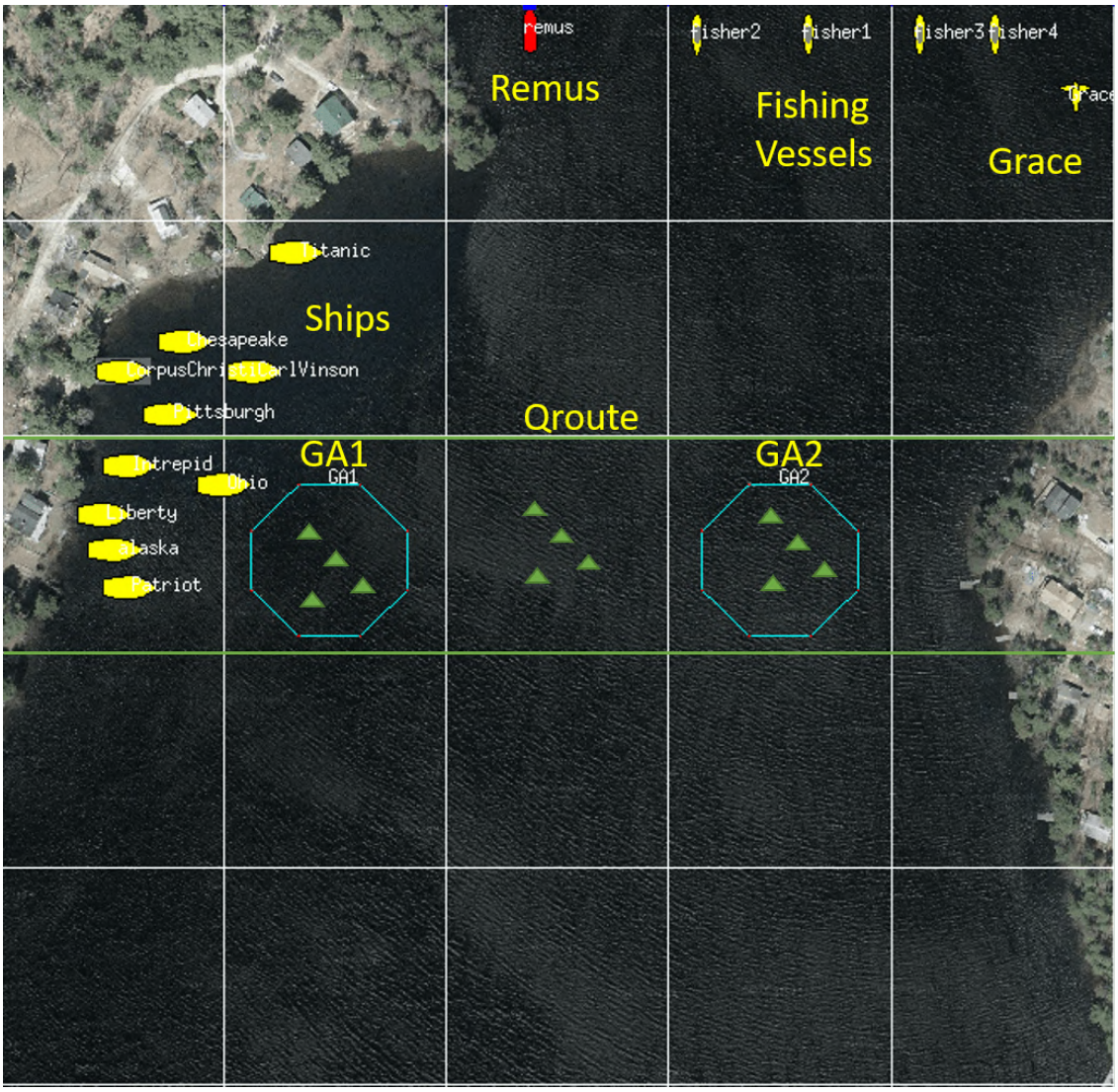

Figure 2.1: Simulation of the underwater mine clearance domain in Moos IvP. The Q-route extends from the left to the right side of the map and represents the path that ships (10 in yellow shown) will traverse. Grace (upper right in yellow) is a glider to survey the entire region for fish and the fishing vessels (4 in yellow shown) will catch fish in the region. The Remus AUV (red) must attempt to clear mines in green areas GA1 and GA2 to support the goals of ships reaching shore. Unexpected mines also exist within the route, and the AUV must delegate the goals.

Figure 2.1 shows an underwater mine clearance domain [25, 40]. The Q-route [42]

indicated by the parallel green lines in the figure, is an area through which ships transport their cargo. The blue octagons GA1 and GA2 (Green areas) are areas where mines are expected to be present and the green triangles represent these mines. The environment is also divided into a 5x5 grid consisting of 25 cells. Four different types of agents exist in this domain, and each have their own goals. Remus is an autonomous underwater vehicle whose goals are to survey certain regions (i.e., GA1 and GA2) to make the Q-route void of mines. Grace is a glider whose goals are to search certain cells in the grid to find concentrations of fish. Fishing vessels (3) have goals to catch fish; and ships (10) transport freight from the left end of the Q-route to the other end. While everyone pursues their individual goals, Remus encounters mines between GA1 and GA2 which it did not expect to be present. So it generates a new goal to search the area between GA1 and GA2 to find new mines. However, the Remus does not have enough resources (time) to achieve all these goals. If it cannot achieve all its goals within a certain time, ships traversing the Q-route might hit a mine and damage the shipments. Furthermore, underwater communication incurs a heavy cost.

In this scenario, it is an open question how the Remus can achieve all its goals. If it tries to delegate its goals to other agents, it is not clear with whom it should coordinate. Furthermore, it is unclear how to coordinate to keep the communication cost to a minimum. In critical missions such as these, addressing such issues is essential. We present an approach to address them in the next section and revisit this example at the end of the chapter.

2.2 Technical Approach

Figure 2.2 shows a hierarchical set of concepts that address the research problem. Agents in a multi-agent environment with limited resources must reason about one another and communicate effectively to achieve their goals. Such a task would require an agent to

store and acquire knowledge about other agents and use it for effective coordination. Such functionalities come under Memory Organization and Agent Coordination. In Memory Organization, the agent must represent the world's knowledge and other agents' capabilities, preferences, beliefs, and experiences. This further decomposes into acquiring new knowledge, representing, storing, and retrieving such knowledge. In Agent Coordination, the agent must use the knowledge of other agents in reasoning about both collaboration and communication, which breaks down into Multi-agent Goal Reasoning, Theory of Mind, and Explanations. In Multi-agent Goal Reasoning, the agent tries to answer when and what to collaborate with other agents and further makes an informed decision around the mode of collaboration (delegation or sharing). In Theory of Mind, the agent uses its knowledge to answer the question of whom it should collaborate with to share the required knowledge for successful collaboration. Furthermore, Explanations explain the motivation behind the collaboration to make the other agent understand the priority of the situation, thereby explicitly answering the question on how to collaborate. The following subsection covers each concept in some detail.

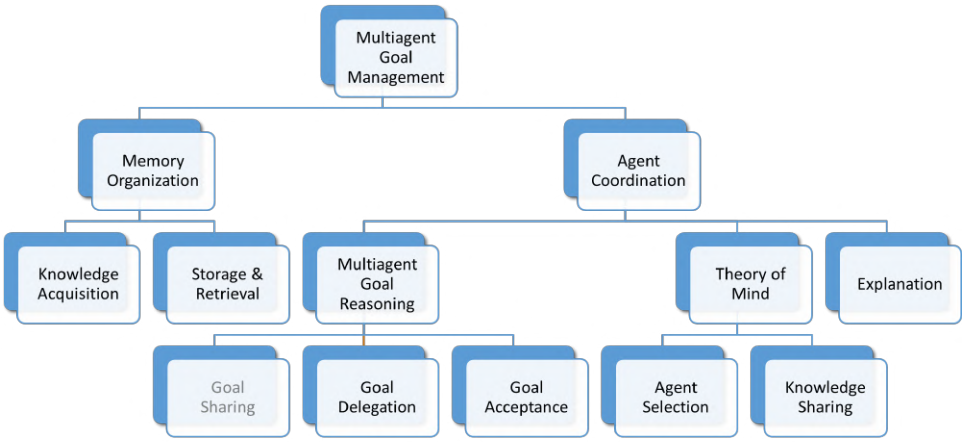

Figure 2.2: Major concepts that address multi-agent goal management.

2.3 Memory Organization

To be effective, an agent in a multi-agent environment must organize its knowledge about itself and other agents. In our research, such knowledge includes beliefs, goals, experiences, domain knowledge and a case-base of explanations which an agent could refer to while making a rational decision. In this section, we will discuss our view towards each of these categories of knowledge, their representation and their usefulness to an agent.

The **domain knowledge** of an agent a is represented formally as the state transition system $\Sigma^a = (S^a, A^a, \gamma^a)$. S^a is the set of all possible states, and A^a is the set of all possible actions. In addition, gamma is a state transition function $\gamma^a : S^a \times A^a \rightarrow S^a$ that returns the resulting state of an executable action given a current state s_c^a. The **Beliefs** $\hat{s}^a$ of an agent are the states that the agent holds to be true in the world $s^a \in S^a$, which is given by $\hat{s}^a = \hat{s_e}^a \cup \hat{s_c}^a$; where $\hat{s_e}^a$ and $\hat{s_c}^a$ are the expected and currently observed states of an agent. The **Goal Agenda** $G^a \subset S^a$ are the states that an agent desires to achieve. Since an agent cannot work on all the goals at the same time, it applies the goal selection operation $g_c^a \leftarrow \delta^{se}(G^a)$ to select a subset of current goals to pursue. Furthermore, an agent plans to achieve its current goals. A plan π^a is a sequence of actions $\pi^a = \langle \alpha_1^a, \alpha_2^a ... \alpha_n^a \rangle \in \Pi^a$, where $\alpha_i^a \in A^a$ which when executed achieves the current goal $g_c^a \leftarrow \gamma(\gamma(s_c^a, \pi^a[1]), \pi^a[2..n])$ [9]. Π^a is a set of all possible plans. To obtain a plan π^a, an agent uses a planning function $\phi : S^a \times G^a \times \Pi^a \rightarrow \Pi^a$ [9]. While an agent executes its actions, it gains **experiences** $\epsilon_c^a = \langle s_0^a, \alpha_1^a, ... s_{c-1}^a, \alpha_c^a \rangle$, where α_c^a is the current action of an agent [10].

Explanations help an agent when it encounters an anomaly; an anomaly occurs when expectations do not entail the observations of an agent i.e., $\hat{s_e}^a \not\models \hat{s_c}^a$. In such a situation the agent retrieves an explanation χ^a from the **case-base of explanations** L^a to understand the reasons behind the anomaly. The agent then formulates a goal in response to the anomaly, which is given by $g^a \leftarrow \delta^*$. We will now discuss each of these knowledge categories in detail.

Beliefs $\hat{s}^a$ are often represented in a first-order predicate logic and are useful to an

agent in planning for goals G^a, goal formulation δ^* and in goal delegation δ^{de}. While planning to achieve its current goals $g_c^a \in G^a$, an agent finds the best actions $\langle \alpha_1^a, \alpha_2^a, ...\alpha_n^a \rangle$ that when applied to the agent's current state s_c^a achieves g_c^a; in goal formulation δ^*, an agent generates its own goals as a response to a problem state; in goal delegation δ^{de}, an agent asks other agents to achieve some of its goals. In a dynamic multi-agent environment, an agent's belief states are updated in the following cases.

- Agent's own action results in a state change.
- Agent's perception of another agent's action results in a state change.
- Agent learns a state through coordination with other agent.
- Agent perceives an exogenous action (i.e., an agent less event such as a rain storm) or a resultant state after the exogenous action.
- Agent believes in a state that is responsible for the cause of an unexpected state while perceiving the world.
- Agent believes in a state that is responsible for the cause of an unexpected state while coordinating with another agent.

Goal agenda G^a are the states that an agent desires to achieve. Similar to beliefs $\hat{s}^a$, goals G^a are also represented in first-order predicate logic. In our research, an agent should distinctly store its current goals g_c^a, suspended goals, and already achieved goals $\{g_0^a, g_1^a, ...g_{c-1}^a\}$ with their resource consumption. Note that retrieving already achieved goals $\{g_0^a, g_1^a, ...g_{c-1}^a\}$ that are similar to the current goals g_c^a may help an agent estimate the resources required to achieve g_c^a. Such a resource estimate can help an agent select the goals it can achieve and suspend the others. Similarly, an agent should also store the information of other agents' past goals, current goals and its future goals. This will help the agent determine the goals that can be achieved by other agents.

Experiences ϵ_c^a represent the history in terms of state-action pairs of an agent in the environment. This will help the agent predict or adapt its behavior if a similar experience arises in the future. Similarly, experiences ϵ_c^o of other agents will help an agent determine the competency of other agents in achieving the delegated goals g_d^a.

Domain knowledge Σ^a includes the actions A^a and possible states S^a of an agent in an environment. Having this information will help an agent plan for its current goals g_c^a. Similarly, an agent having domain knowledge Σ^o of other agents will help it determine the capability of other agents during goal delegation δ^{de}.

Case-base of explanations L^a is a memory repository of generalized explanations of events that may happen in the domain [51, 13]. Such information enables an agent to understand the cause of an unexpected state. We assume that the initial cases in the case-base are prepared by experts in the domain.

2.3.1 Knowledge Acquisition

This research is not focused on developing learning algorithms to predict the categories of knowledge as discussed above. For example in multi-agent reinforcement learning (MARL) [5], the agents are trained across large amounts of training data to make them learn to work with each other to maximize the reward function. However, real world scenarios often lack such large amounts of training data. Moreover, we work in a distributed environment with different types of autonomous agents which makes the application of MARL challenging. So instead, we focus on obtaining such knowledge through the interaction between agents in the environment. For example, this knowledge can be obtained by the agent's observations, expectations and interactions with other agents in the world. This allows an agent to make a rational decision and to resolve its conflicting knowledge with the acquired new knowledge.

2.3.2 Storage and Retrieval

Similar to the previous section, this research does not develop optimized algorithms to store and retrieve information from specific knowledge categories. However, for each autonomous agent, we focus on storing each knowledge category separately in memory, including the knowledge categories of other agents. In our research, an autonomous agent possesses two types of memory for storing knowledge categories: working memory and long-term memory. Working memory is the short-term memory that the agent currently possesses, which includes its current beliefs, observations, goals, and recent experiences. Long-term memory contains the history of all the knowledge categories, domain knowledge, and the agent's case-base of explanations. This memory is helpful for an agent to look back and retrieve knowledge whenever necessary.

Retrieval is an essential component of our memory. In our research, an agent retrieves information from long-term memory for three critical tasks. In the first task, when an agent encounters an anomaly, it tries to retrieve an explanation from its case-base of explanations. This retrieved explanation will help an agent understand the anomaly and further formulate new goals as a response [23, 25]. In the second task, an agent often requires knowledge of other agents' capabilities for successful coordination. In such situations, the agent would retrieve other agents' domain knowledge from its long-term memory. Finally, in the third task, an agent should retrieve the knowledge of past interactions with other agents for successful coordination. In such situations, the agent would retrieve those interactions from the list of past experiences.

Figure 2.3 depicts the process of storage and retrieval in long term and working memory. We retrieve knowledge categories from long-term memory to working memory and store them the other way. In long-term memory, an agent stores knowledge of itself and knowledge about other agents. In an agent's knowledge of self, it stores the history of its own beliefs, goals and experiences along with domain knowledge and case-base of ex-

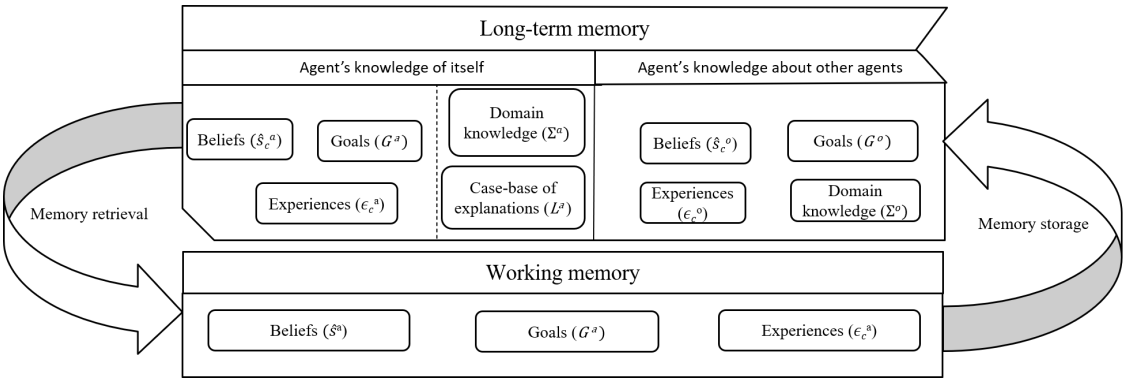

Figure 2.3: Long-term and working memory.

planations. Although the agent's domain knowledge and case-base can be updated, for the most part they remain constant. Similarly, within the agent's knowledge about other agents, it stores a history of beliefs, goals, experiences and domain knowledge about other agents. Here, there is a clear distinction between storing an agent's domain knowledge of itself and the domain knowledge of other agents. The former remains relatively constant while the latter changes over time and stores those changes in its history.

In the working memory, an agent maintains its current beliefs, goals and experiences. It also retrieves information from knowledge categories in long-term memory to make rational decisions and act in the world. For example, to achieve goals, an agent retrieves its domain knowledge from long-term memory to plan and perform a set of actions. Furthermore, to rationally delegate goals to other agents, an agent retrieves information from its long term memory storage about other agents and moves that knowledge to its working memory. Furthermore, an agent stores a history of each of their knowledge categories from working memory into long-term memory.

Each of these knowledge categories will help an agent in every possible task. In the next section, we will see how these knowledge structures will help an agent to successfully coordinate with other agents in the environment.

2.4 Agent Coordination

When an agent anticipates that it cannot achieve its goals in a dynamic environment, it should delegate its goals to other agents. For successful goal delegation, the agent needs to coordinate with other agents. To effectively coordinate, the agent needs to perform the following actions:

- Recognize *when* to delegate its goals.
- Decide *what* goals to delegate.
- Decide about *whom* the agent should request and share its knowledge.
- Decide *how* to delegate its goals.

Instrumental to implementing these actions are the concepts of multi-agent goal reasoning, theory of mind, and explanations. Multi-agent goal reasoning will enable an agent to recognize when and what goals to delegate. Theory of mind will help an agent identify the best capable agent using its knowledge. Finally, explanations will explain the motivations behind the goals to the requesting agent, thereby explicitly deciding how to perform goal delegation. We will review each of these concepts and provide algorithmic solutions in the following sections.

2.4.1 Multi-agent Goal Reasoning

Goal reasoning is a viable computational component in managing an agent's goals [1, 50]. In the event of unexpected events, it enables an agent to perform different goal operations. Such goal operations will help the agent make rational decisions as a response to these exogenous events. In this research, we leverage the implementation of several goal operations but focus more on goal delegation. Furthermore, in a dynamic environment where unexpected situations occur, an agent should often respond to the developing problems. An

intelligent agent should generate new goals in response to the developing problems. However, with limited resources, an agent cannot always pursue its new goals and still achieve its current goals. So, to balance this conflict an agent can decide to delegate some of its goals to other agents. To perform goal delegation, an agent needs to address the following problems:

- Recognize *when* to delegate its goals. and
- Determine *what* goals to delegate.

An agent with goal reasoning ability can address the above mentioned problems. In our previous work [26], we presented an approach for an agent to select the goals it can achieve in a dynamic environment. Similarly, [34] also presented another goal selection strategy for an agent in dynamic environments. Such strategies, can help an agent determine the goals it can achieve and delegate the remaining goals. Furthermore, we will discuss when an agent should delegate and share goals in the following subsections.

2.4.1.1 Goal Delegation

Autonomous agents often pursue goals that are critical to the mission. When an agent anticipates that it cannot achieve its goals, goal delegation enables the agent to delegate its goals to other agents in the environment. In doing so, the agent can still achieve its goals and improve its performance towards the mission. The primary step involved in a goal delegation process is to recognize the need for delegation. We use goal specific resource estimation and priority functions [26] to select the goals an agent can achieve, and delegate the remaining goals in the order of maximum priority.

Table 2.1 shows an algorithm that returns a set of delegated goals g_d when there is a need for goal delegation. *DetectDelegation* takes the following inputs: goal agenda of the current agent ($\hat{G}_c = \{g_1, g_2, ...g_c, ...g_m\}$) and current state of agent's perception (s_{cc}). While there are enough estimated resources $\hat{R}(s_{cc})$ in the world (line 2), the agent tries to compute

the order of goals it can achieve $g_{achievable}$. Such a goal ordering is possible by applying goal-specific priority $\hat{P}(g, s_{cc})$ and resource estimation functions $\hat{R}(g, s_{cc})$ [26]. The agent then selects a set of goals g_s which has maximum priority (line 3) and minimum resource consumption (line 4). Later, it computes the estimated resources to achieve selected goals g_s (line 5). Furthermore, the agent adds the selected goals to its achievable goals $g_{achievable}$ (line 6). The agent then continues executing these steps until all the goals in its goal agenda are ordered (line 7). Finally, it computes the goals it must delegate (g_d) by subtracting the goals in its agenda with the goals it can achieve (line 9). The function finally returns the set of delegated goals (line 10).

Table 2.1: A method for detecting goal delegation by the current agent ($agent_c$). Parameter $\hat{G}_c$ is the goal agenda of the current agent, and s_{cc} is the known observed state of $agent_c$.

	DetectDelegation $(\hat{G}_c, s_{cc})$	
1.	$g_{achievable} \leftarrow \emptyset$	
2.	$\hat{r} \leftarrow \hat{R}(\hat{G}_c, s_{cc})$	*// Estimation of agent's resources*
3.	$while\ \hat{r} > 0\ do$	*// Loop until the agent has enough resources*
4.	$g_s \leftarrow \underset{g \in (\hat{G}_c - g_{achievable})}{argmax} \hat{P}(g, s_{cc})$	*// Select goals with maximum priority*
5.	$g_s \leftarrow \underset{g \in g_s}{argmin}\ \hat{R}(g, s_{cc})$	*// From the goals with the same priority, select goals with minimum resources*
6.	$\hat{r} \leftarrow \hat{r} - \hat{R}(g_s, s_{cc})$	*// Estimate the remaining resources*
7.	$g_{achievable} \leftarrow g_{achievable} \cup g_s$	*// Add selected goals to agent's achievable goals*
8.	$if\ (\hat{G}_c - g_{achievable})\ is\ \emptyset\ then$	*// Break, if agent can achieve all its goals*
9.	$break$	
10.	$g_d \leftarrow (\hat{G}_c - g_{achievable})$	*// Goals agent cannot achieve*
11.	$return\ g_d$	*// Return delegated goals*

To understand the application of the above goal delegation algorithm, let us revisit the example scenario from section 2.1. In this example, Remus has achieved its goal to clear mines in GA1. While pursuing the goal to clear mines in GA2 by transiting from GA1 to GA2, it encounters mines between the two areas and formulates a new goal to clear mines in GA3, as shown in figure 2.4.

Following the algorithm, the agent anticipates that it does not have enough resources

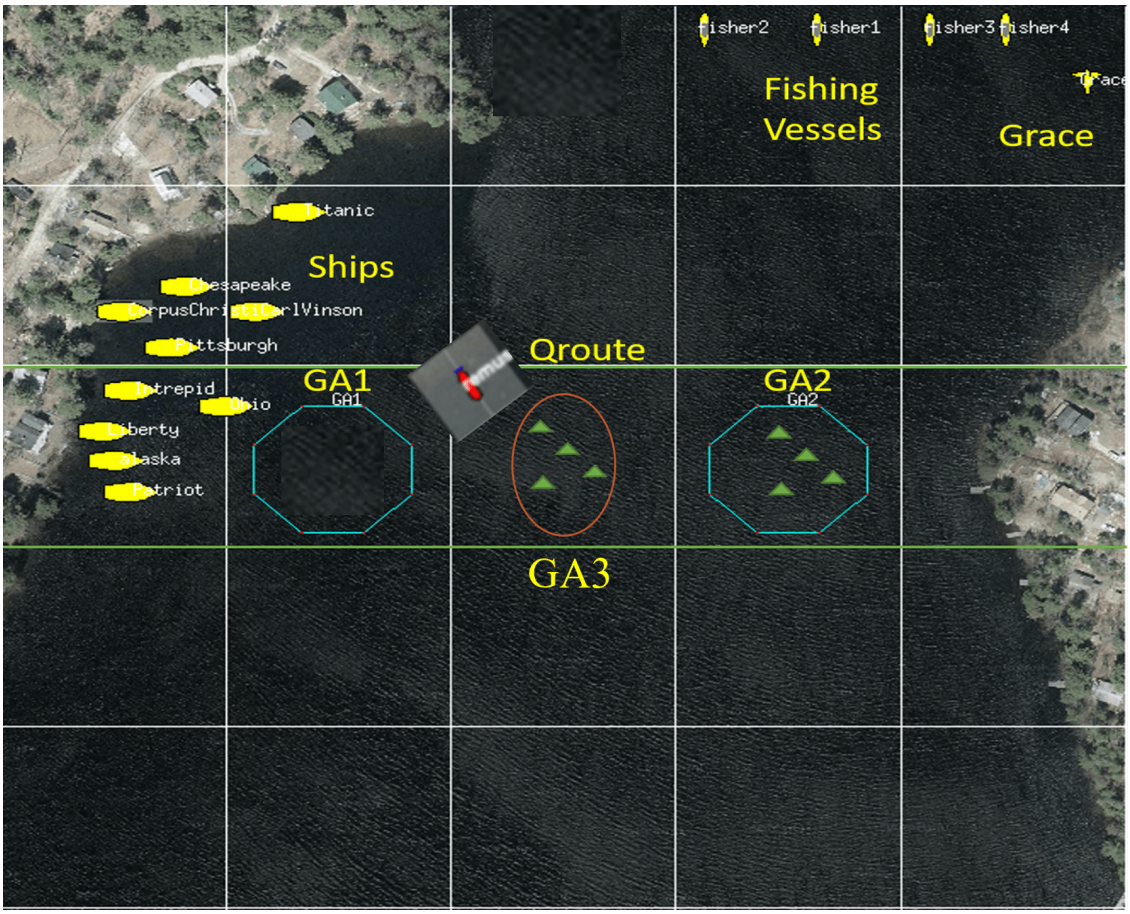

Figure 2.4: Simulation of the underwater mine clearance domain. Remus (red) choose to pursue clearing mines in GA3 and decides to delegate its goal to clear mines in GA3 to other agents in the environment.

(time) to achieve its goals to clear mines in GA2 and GA3. Although both goals have the same priority, clearing mines in GA3 takes less resources than in GA2. So, the agent decides to clear mines in GA3 and delegates the goal to clear mines in GA2 to other agents in the environment.

In this section, we have seen how the agent uses DetectDelegation algorithm to determine when and what goals to delegate. The following section describes how a receiving agent should respond to the goal requests.

2.4.1.2 Goal Acceptance and Goal Rejection

Often an autonomous agent in a multi-agent system cannot accept all requested goals. It is often limited by its resources and must reason about its current goals while accepting or delegating the remaining ones. We use DetectDelegation algorithm from section 2.4.1.1 to identify the goals the agent cannot pursue. Then, the agent rejects the requested goals that are unachievable and accepts the remaining goals.

Table 2.2 shows an algorithm that returns the goals the agent can achieve given the agent's goal agenda $\hat{G}_i$, current state s_{ci} and delegated goals g_d. A requested agent $agent_i$ estimates the goals it cannot achieve $g_{unachievable}$ using the *MonitorDelegation* algorithm from section 2.4.1.1 (line 1). Later $agent_i$ adds delegated goals g_d to its goal agenda $\hat{G}_i$ if they are not in unachievable goals $g_{unachievable}$ (line 2 and line 3). Finally this returns the list of goals $agent_i$ can achieve (line 4) while rejecting the remaining goals.

Table 2.2: A method to accept/reject a delegated goal by the selected agent $agent_i$. Parameter $\hat{G}_i$ is the goal agenda of the selected agent, s_{ci} is the known observed state of $agent_i$, and g_d is the set of goals delegated by the current agent $(agent_c)$ to the selected agent $(agent_i)$.

	GoalAcceptReject $(\hat{G}_i, s_{ci}, g_d)$	*// $agent_i's$ goal agenda, its current state and delegated goals*
1.	$g_{unachievable} \leftarrow DetectDelegation(\hat{G}_i \cup g_d, s_{ci})$	*// Obtain unachievable goals*
2.	$if\ g_d \notin g_{unachievable}$	*// If delegated goals are achievable*
3.	$\quad \hat{G}_i \leftarrow \hat{G}_i \cup g_d$	*// Add delegated goals to goal agenda*
4.	$return\ \hat{G}_i$	

From the example in the previous section, let us assume that Remus delegates its goals to clear mines in GA2 to Grace. Since Grace is an underwater glider and cannot clear mines, following the GoalAcceptReject algorithm, the requested goal remains in the list of unachievable goals. Therefore, Grace rejects the requested goal. In the next section, we will see how goal sharing can help Remus and Grace achieve all the goals.

2.4.1.3 Goal Sharing

When an agent cannot successfully delegate its goals to another agent, it should at least share the goals between multiple agents. One approach to split goals is through *Hierarchical Goal networks (HGN)* [54]. HGN is a method where a higher level goal is subdivided into smaller constructs which, when all achieved, will achieve the higher level goal. Using these HGNs an agent could share the subdivided goals with multiple agents in the environment to complete a higher level goal which one agent could not achieve on its own. Note that this work on goal sharing is in the preliminary stage, and we intend to pursue it in the future.

To understand the application of Goal Sharing, let us revisit the example from the previous section. Remus delegates the goal to clear mines in GA2, and Grace rejects it. Figure 2.5 shows the HGN for the delegated goal. Following the approach of HGN's, Remus would pursue the goal to neutralize mines in GA2 and delegate the goal to identify mines in GA2 with Grace.

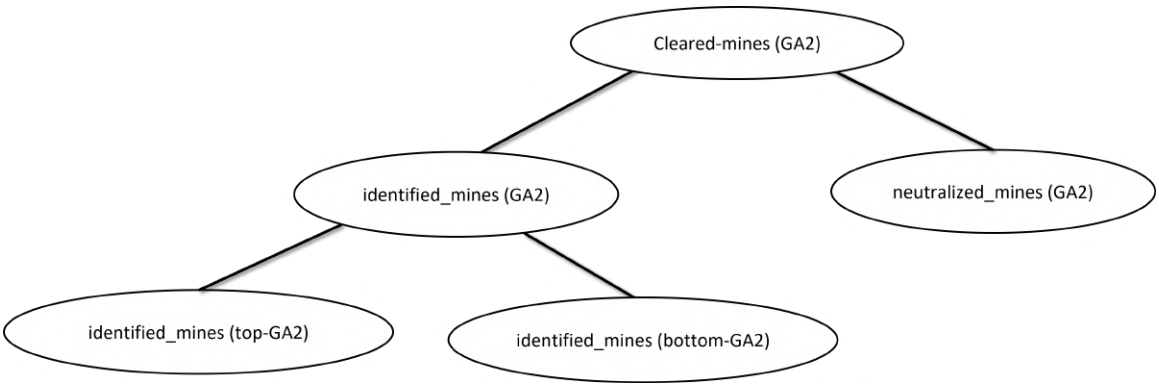

Figure 2.5: HGN representation of the goal to clear mines in GA2.

In this section, we have seen using DetectDelegation algorithm, when an agent should delegate its goals and what it should delegate. Furthermore, using GoalAcceptReject, we have also seen how the receiving agent should respond to the requested goals. In the following section, we will look at the theory of mind approach to whom an agent should delegate its goals.

2.4.2 Theory of Mind

There are two major competing theories in psychology on the theory of mind: Theory theory and Simulation theory. Theory theory, states that an agent holds a naive algorithm to infer beliefs, goals and intentions of others and, eventually, improves its algorithm as it gains more experience. Simulation theory states that an agent simulates the actions of other agents to predict their behavior. For our purposes, theory of mind will be grounded in the Simulation theory.

In a distributed multi-agent environment, an individual agent often acts on its own. It is not feasible for the agent to know all the information about other agents' goals and current states. So to delegate goals, the agent is limited by its knowledge about other agents. Theory of Mind refers to reasoning about other agents' mental states using the agent's knowledge about other agents.

In our multi-agent system, we developed a theory of mind approach to select an agent among all the other agents in the environment for goal delegation. This approach is represented in section 2.4.2.1. Furthermore, following this approach also aids the delegating agent in sharing required knowledge with the selected agent. Such knowledge can help the selected agent to successfully achieve the delegated goals. This knowledge sharing algorithm is represented in the section 2.4.2.2

2.4.2.1 Agent Selection

A delegating agent must select another agent capable of achieving its delegated goals among all the other agents in the environment. Selecting an agent must use its knowledge about other agents' states, goals, and domain knowledge to simulate their ability to achieve the delegated goal. To perform such an agent selection, we use landmarks [30] for the delegated goals and select the agent that takes minimum cost to achieve these landmarks. Landmarks are the states that exist in all possible plans to achieve the goal.

Table 2.3 represents the algorithm that the current agent $agent_c$ applies to select an

$agent_i$ to delegate its goals g_d. The *AgentSelection* algorithm takes the following inputs: all the candidate agents in the environment CA = $\{agent_1, agent_2, ...agent_k\}$, their domain knowledge $(\Sigma_1, \Sigma_2, ...\Sigma_k)$ where Σ is a state transition system represented as (S, A, γ), $agent_c$'s own domain knowledge Σ_c and set of goals to delegate g_d. $Agent_c$ then computes the landmarks $L[1..m]$ to achieve the delegated goals g_d (line 1). Later, it selects $agent_i$ that has the minimum estimated cost to reach the first landmark (line 2 and line 3). Finally, $agent_c$ makes a delegation request to $agent_i$.

Table 2.3: A method for selecting an agent ($agent_i$) by the current agent ($agent_c$) to delegate its goals g_d. Parameter CA is the set of all candidate agents in the environment, $(\Sigma_1...\Sigma_k)$ are corresponding candidate agents' domain knowledge known to the current agent ($agent_c$), Σ_c is the current agent's domain knowledge, s_{cc} is the known observed state of $agent_c$, and g_d is the set of goals to delegate by $agent_c$.

AgentSelection$((CA, (\Sigma_1...\Sigma_k), \Sigma_c, s_{cc}, g_d)$	
1. $L[1, 2, ..m] \leftarrow Landmarks(\Sigma_c, s_{cc}, g_d)$	*// Obtain landmarks to achieve delegated goals*
2. $agent\_index \leftarrow \underset{j \in CA}{argmin}$ cost$(\hat{\Pi}(\Sigma_j, s_{cj}, L[1])$	*// Select agent with minimum cost // to achieve the first landmark*
3. $agent_i \leftarrow CA\,[agent\_index]$	
4. $return\ agent_i$	

To better understand the algorithm, let us revisit the example from the underwater mine clearance domain as shown in the figure 2.6. Remus decides to delegate the goal to clear mines in GA2. For simplicity, let us make an assumption that Grace is now equipped with mine clearing equipment. To delegate its goals, Remus follows the AgentSelection algorithm to obtain the landmarks for the goal. These landmarks include the state of an agent being at the location and identifying mines at GA2. Remus then estimates the planning cost of every agent based on its knowledge about others and selects the agent Grace to delegate its goals. Grace not only has the capability to identify mines but also takes less time to reach GA2 compared to all the other agents in the environment.

Now that the agent has decided on whom to delegate its goals. In the next section, we will see our theory of mind approach on what knowledge would it decide to share with the

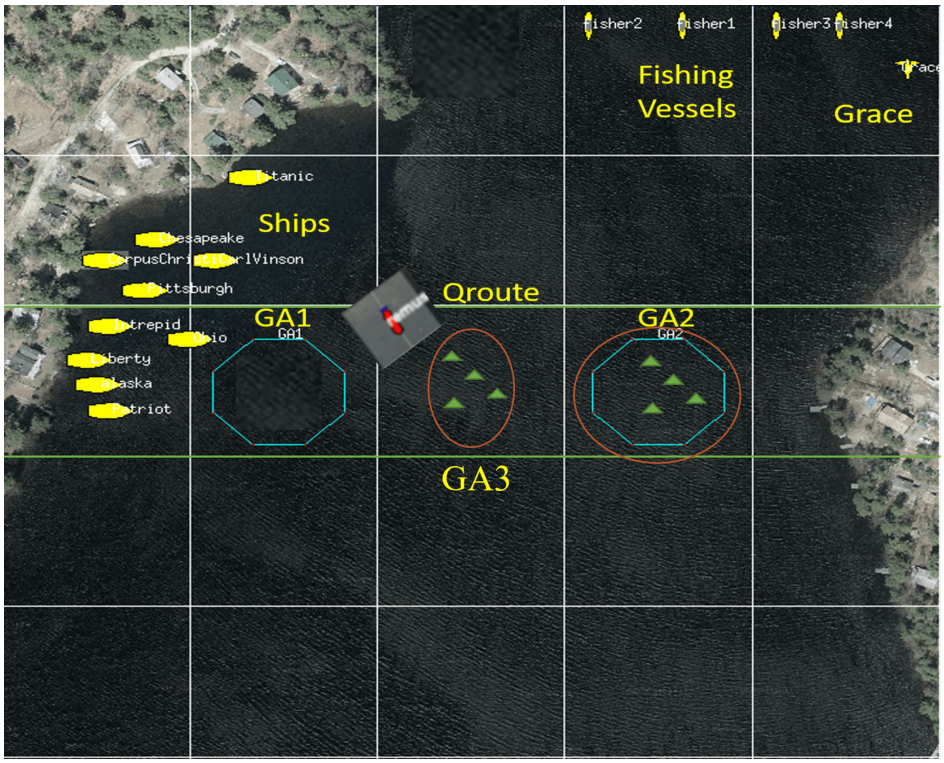

Figure 2.6: Simulation of the underwater mine clearance domain. Remus (red) choose to pursue clearing mines in GA3 and decides to delegate its goal to clear mines in GA2 to other agents in the environment.

receiving agent.

2.4.2.2 Knowledge Sharing

Given the selected agent $agent_i$ from section 2.4.2.1 and the goals to delegate g_d from 2.4.1.1, the delegating agent $agent_c$ should share required knowledge to achieve the goal. However, due to partial observability and the distributed nature of the multi-agent environment, $agent_c$ cannot know what knowledge the other agents necessarily require. However, $agent_c$ should reason using its knowledge to determine some information required by $agent_i$ to achieve the delegated goals. To obtain such information, the agent computes expectations by planning to achieve the first landmark using its own knowledge. These expectations are the future states of the requesting agent when it tries to achieve the goal. Furthermore, the agent then chooses to share all the knowledge related to the attained ex-

pectations that it believes $agent_i$ does not know. Such knowledge will help the receiving agent understand the problems it should avoid and the benefits from available opportunities.

Table 2.4 formally represents the knowledge sharing algorithm which outputs the states (s_{share}) that $agent_c$ should share with $agent_i$. KnowledgeSharing algorithm takes the following input: $agent_i$'s domain knowledge $\Sigma_i = (S_i, A_i, \gamma_i)$, $agent_c$'s known state of $agent_i$ (s_{ci}), $agent_c$'s current state of the world (s_{cc}) and the first Landmark $L[1]$. $agent_c$ plans $\langle a_1, a_2, ...a_n \rangle$ to achieve the first landmark (line 1). $agent_c$ then computes expectations for every action of the plan (line 5). These expectations comes from the preconditions and the effects of the actions. Later $agent_c$ then tries to retrieve all the states (s_r) that are abstractly related to the computed expectations (s_{ei}) and its current state (s_{cc}) (line 6). Abstractly related states are those states that have similar predicates or contain arguments that exists in both the given state representations. Later, $agent_c$ adds it to the states s_{share} to share with $agent_i$ while removing the states it has already shared s_{ci} (line 7). Finally $agent_c$ updates its knowledge about $agent_i$ (line 8) and returns the states s_{share} to share (line 9).

Table 2.4: A method for computing the knowledge the current agent ($agent_c$) must share with the delegated agent ($agent_i$). Parameter Σ_i is the known domain knowledge of $agent_i$, s_{ci} is the currently observed state of $agent_i$, s_{cc} is the known observed state of $agent_c$, and $L[1]$) is the first landmark.

KnowledgeSharing $(\Sigma_i, s_{ci}, s_{cc}, L[1])$	
1. $\langle a_1, a_2, ...a_n \rangle \leftarrow \hat{\Pi}(\Sigma_i, s_{ci}, L[1])$	*// Estimated actions to achieve first Landmark*
2. $s_{share} \leftarrow \emptyset$	*// Knowledge states to share with* $agent_i$
3. $s_{ei} \leftarrow \emptyset$	*// The expected states of* $agent_i$
4. $for\ a\ in\ \langle a_1, a_2, ...a_n \rangle$	
5. $s_{ei} \leftarrow s_{ei} \cup pre(a) \cup a^+ - a^-$	*// Expectations of* $agent_i$ *stemming from the results of action* a
6. $s_r \leftarrow AbstractRelatedStates(s_{ei}, s_{cc})$	*// Related states of* $agent_c$ *to expectations*
7. $s_{share} \leftarrow s_{share} \cup (s_r - s_{ci})$	*// Add related states to share with* $agent_i$
8. $s_{ci} \leftarrow s_{ci} \cup s_{share}$	*// Update knowledge of* $agent_i$
9. $return\ s_{share}$	

To understand the KnowledgeSharing algorithm, let us revisit the example from the previous section. Remus chooses Grace to delegate its goal (i.e., to clear mines in GA2).

Now it uses the KnowledgeSharing algorithm to share the required knowledge. First, Remus using its own knowledge about Grace, obtains expectations from its plan to achieve the first landmark (i.e., being at GA2 to identify mines). These expectations include the future state of Grace being at the Q-route (area between the two parallel lines in 2.6. Since Remus already knows the location of some mines in the Q-route (GA3), it shares those locations with Grace. This information will help Grace traverse those regions carefully or avoid them altogether to stay safe and continue pursuing the delegated goal. In the next section we will see how an agent should delegate its goal using explanations.

2.4.3 Explanation

After the requesting agent selects an agent for delegation, it should coordinate with the selected agent. We use explanations to coordinate, and an explanation from a requested agent helps the selected agent understand the justifications behind the delegated goals. In this subsection, we will discuss the structure of the explanation and its representation in the goal delegation process.

We represent these explanations in terms of an *explanation pattern (XP)* [13, 23, 52] structure. An XP (Figure 2.7) is a data structure that represents a causal relationship between multiple states and/or actions. 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. These must be true for the XP to apply.
- **XP-asserted node:** An action, state, or XP that contributes to the cause of the Explains node.
- **Internal node:** Optional nodes between the antecedent and the consequent.

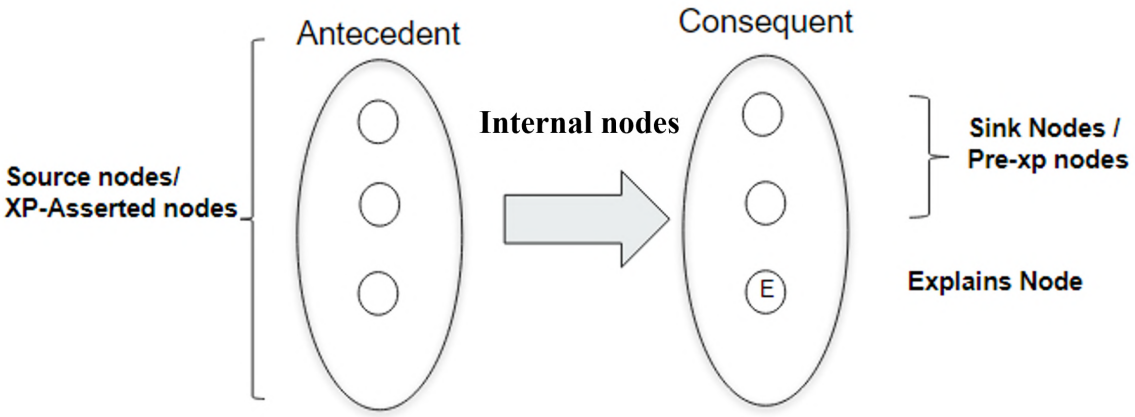

Figure 2.7: XP structure.

An explanation pattern represents a causal structure in which XP-asserted nodes form an antecedent and Pre-XP nodes form the consequent. Note an XP also contains internal nodes which links the antecedent of the XP with the consequent. In the following subsection we will discuss how a receiving agent coordinates with the selected agent.

To request a selected agent to achieve its goals, the requesting agent should explain the motivations $M = \{m_1, ..., m_s\}$ behind the delegated goal. Such motivations will help the selected agent make a rational decision to accept or reject the delegated goals. To fit into the context of coordinating goals to other agents, we adapt the definition of nodes in the XP structure as follows:

- **Explains node:** (g_x) state desired by the requesting agent (i.e., delegated goal).
- **Pre-XP node:** (ρ_x) state/action that should be true to achieve the desired state (i.e., conditions for a goal to be valid).
- **XP-asserted node:** (m_x) Motivated action, state, or XP that contributes to the reason behind the request.

Figure 2.8 shows the tweaked explanation structure to justify a delegated goal (g_x). The antecedent includes the agent's motivations $\{m_1, ..., m_s\}$ to pursue the delegated goal

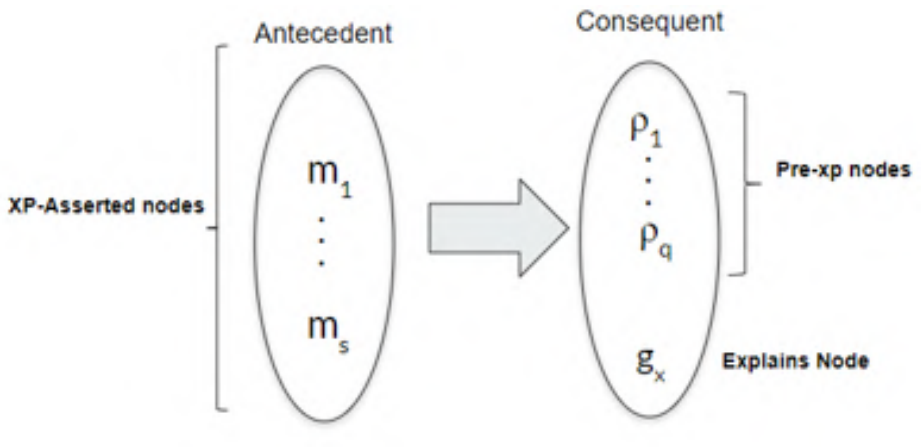

Figure 2.8: XP structure for justifying a delegated goal.

(g_x) while the consequent contains Pre-XP nodes and an explains node. Pre-Xp nodes are the conditions $\rho_1, ..., \rho_q$ for a delegated goal to be valid while the explains node is the delegated goal (g_x) itself. The above explanation structure will help the selected agent understand the importance of the delegated goal among its own goals. Such an understanding will aid the selected agent to reason about its commitment to the delegated goal.

To understand the importance of explanations in the goal delegation process, let us revisit the underwater mine clearance domain example from the previous section. Until now, Remus decides to delegate the goal to clear mines in GA2 to Grace and decides to share the knowledge of existing mines in the Q-route. Now, Remus uses an explanation pattern to convey the goal's motivations and establish conventions. The convention[1] for this goal is the future state of ships traversing the Q-route while the motivation behind the delegated goal is to make a safe passage for the cargo-carrying ten ships. This information will help Grace understand the priority of the delegated goal and thereby plan accordingly to achieve it. The following section will show how an agent performs the delegation operation by putting together the outputs from DetectDelegation, AgentSelection, KnowledgeSharing, and Explanation algorithms.

[1]For instance, if the ships change their passage to a different Q-route, Grace no longer needs to achieve the goal as it is not valid. Thus conventions help agents determine if the goal is still valid to pursue or not.

2.5 Conclusion

After the requesting agent decides on *when*, *what*, to *whom*, and *how* to delegate its goals, the next step is to coordinate with the receiving agent for successful delegation. To perform this operation, the agent generates the following goal $requested(agent_c, agent_i, g_d, s_{share}, \chi)$ where $agent_c$ is the requesting agent, $agent_i$ is the receiving agent, g_d is the delegated goal, s_{share} is the information to share with $agent_i$ and χ is the explanation.

The requesting agent $agent_c$ will plan to achieve the goal. It then executes its plan to successfully make the request. Furthermore, the agent waits for an acknowledgement form $agent_i$ before continuing to pursue its own goals.

In the underwater mine clearance example. Remus will generate the following goal *requested (Remus, Grace, cleared_mines(GA2),* s_{share}*,* χ_r*)* where s_{share} contains the following set of states (*mine-at(GA3, mine1) ... mine-at(GA3, mine4)*). It then plans to achieve the goal, executes the plan and receives an acceptance form Grace. Thus, both Remus and Grace creates a successful passage to the ships and heads to their initial locations as shown in the figure 2.9.

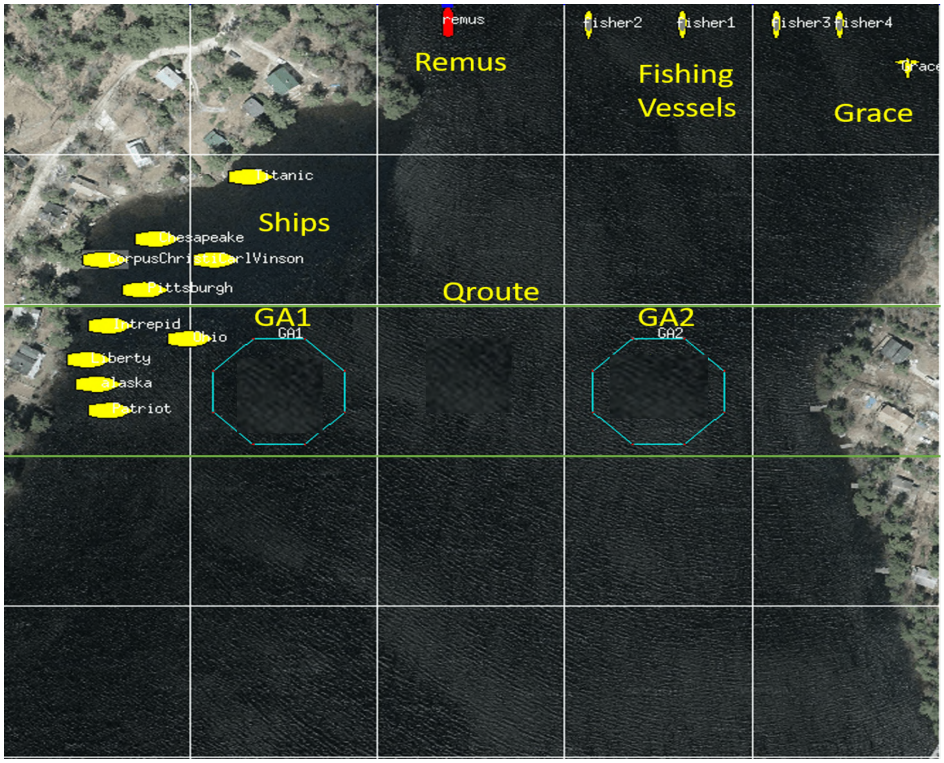

Figure 2.9: Simulation of the underwater mine clearance domain. Remus (red) and Grace (yellow) clears mines in GA2 and GA3 and makes a safe passage for the Ships to transit.

Experimental Design and Evaluation

We have implemented our work in the marine life survey domain and the rover domain to test our claims. The marine life survey domain is a simulated version of surveying underwater marine life in the real world. The agents in this domain are underwater vehicles that possess similar capabilities to find oceanic life concentrations. The Rover domain is one of the classic planning benchmark domains introduced as a simple representation of the NASA Mars Exploration missions. The agents in this domain are equipped with different but possibly overlapping sensors to perform experiments on a planet's surface. While both environments are partially observable and distributed in nature, they are different in many ways. For example, they differ in the goals they pursue, the observations made, the problems that occur, and the challenges they encounter.

This chapter describes each of these domains in detail and provides experimental evidence to our claims presented in chapter 1. Here, we have performed experiments with different design parameters across both domains. In the following sections we will look at these experimental designs and the empirical results in detail.

- ***Claim 1:*** *Goal delegation using a theory of mind approach causes the goal achievement performance (i.e., percentage of goals achieved successfully) in a distributed, multi-agent context to significantly improve relative to a traditional goal reasoning multi-agent system which does not delegate goals.*

- ***Claim 2:*** *Explanations helps the receiving agent understand the priority behind the*

delegated goals, thereby improving the performance of goal achievement in a multi-agent system compared to a multi-agent goal reasoning system with only goal delegation.

3.1 Comparison with Different Multi-agent Systems

To evaluate the performances of each of our algorithms introduced in Chapter 2 and to compare them with external multi-agent systems we introduced eight variations. They are as follows,

1. **Ideal:** Unexpected events (i.e., anomalies) do not occur in the ideal multi-agent system's environment. Everything goes according to the plan, and the agents can achieve their own goals without goal delegation. Having this type of multi-agent system implemented in a domain will help us understand the highest performance the system can perform at any given time. As such, it provides and upper bound on performance.

2. **Traditional goal reasoning:** In the traditional multi-agent goal reasoning condition, agents encounter anomalies and, in response, they perform relevant goal operations (except goal delegation). These goal operations include goal selection, goal formulation, goal change, and goal achievement. Comparing this multi-agent system with others will enable us to understand the importance of delegating goals to other agents using our theory of mind approach.

3. **Delegation (*DetectDelegation*, *AgentSelection*):** In the multi-agent delegation condition, agents perform different goal operations in response to the encountered anomalies. Furthermore, they decide when and what goals to delegate using the *DetectDelegation* algorithm (see 2.1) and finally coordinate them to the agent(s) determined by the *AgentSelection* algorithm (see 2.3 . Here that the receiving agent

will always accept the goals delegated to it. However, it prioritizes achieving its own goals first and only later achieves the delegated goals. This multi-agent condition will show us the impact of the algorithms mentioned above on a traditional multi-agent goal reasoning condition.

4. **Accept_reject (*DetectDelegation*, *AgentSelection*, *GoalAcceptReject*):** The accept_reject multi-agent condition enhances goal delegation with an additional decision-making ability for the receiving agent. It can either accept or reject the delegated goals as determined by the *GoalAcceptReject* algorithm (see 2.2. Note that if the receiving agent(s) denies the given goal(s), the requesting agent will then delegate its goals to the next best agent provided by the *AgentSelection* algorithm.

5. **Knowledge sharing (*DetectDelegation*, *AgentSelection*, *GoalAcceptReject*, *KnowledgeSharing*):** Agents in the knowledge sharing delegation condition do everything similar to the agents in the Accept_reject multi-agent condition. In addition, they share the required knowledge with the receiving agent determined by the *KnowledgeSharing* algorithm (see 2.4). This multi-agent condition will enable us to evaluate the impact of sharing such knowledge with the receiving agent. Note that the receiving agent(s) will prioritize achieving their own goals first and later pursue the delegated goals.

6. **Explanation (*DetectDelegation*, *AgentSelection*, *GoalAcceptReject*, *KnowledgeSharing*, *Explanation*):** Multi-agent explanation condition is very similar to the knowledge sharing condition. However, in this condition, requesting agent(s) will also share motivations behind the delegated goals through explanations. Such reasons will enable the receiving agent(s) to understand the priority of the given goals and achieve them accordingly. The performance of the explanation multi-agent condition represents our complete approach towards the goal delegation process using theory of mind.

7. **Random (*DetectDelegation*, *RandomAgentSelection*):**

   The multi-agent aandom condition shares the same capabilities as the traditional goal reasoning multi-agent system. However, agents in this multi-agent system will decide when and what goals to delegate using the *DetectDelegation* algorithm and randomly choose an agent to pursue its goals. Comparing this multi-agent system with the goal delegation multi-agent system will help us understand the impact of the *AgentSelection* algorithm on the agents' performance.

8. **Auction mechanism (external multi-agent system):**

   The auction mechanism is the most standard agent-coordination approach in the multi-agent systems community. In this multi-agent condition, requesting agents will make their goals available for bidding. Other agents will take the goals and provide their cost to achieve them. The requesting agent will then delegate the goals to the receiving agent with the lowest bid. Note that the agents still use the *DetectDelegation* algorithm to determine when and what to delegate. Furthermore, the receiving agent will still prioritize achieving its own goals over the given goals.

Evaluating the performance of each process in our goal delegation approach (2-6) while comparing them to some external multi-agent systems (7-8) is the primary intent behind the introduction of these eight multi-agent conditions. In the following sections, we will see the implementation of these multi-agent systems in different experimental scenarios of the marine life survey domain and the rover domain.

3.2 The Marine Life Survey Domain

Surveying marine environments using *autonomous underwater vehicles (AUV*s) is often time-limited and challenging. These autonomous vehicles typically collect readings of temperature, salinity, and pressure throughout the survey region and investigate key aquatic

life features. One such critical element is to detect the areas of high fish concentrations underwater, which we refer to as fish *hot spots* in our research. Finding these locations of major fish hot spots is essential for researchers to study aquatic behavior to maintain ecological balance. In addition, these studies significantly help conserve endangered species in oceanic environments. Therefore discovering locations of major hot spots is the prime mission of these AUVs. However, several barriers exist in marine environments that make this mission challenging to achieve. For example, sea creatures may attach themselves to the AUVs and hinders their movement. Tides and currents in water also make the underwater traversal more challenging. Furthermore, several obstacles such as coral reefs or underwater rock formations may appear, requiring a change in the course of these AUVs.

Figure 3.1 shows the region of Gray's Reef National Marine Sanctuary located on the inner shelf of the South Atlantic Bight off the coast of Savannah, GA. The red area represents the research area and is restricted for public access. In this research-only region, our team regularly deploys AUVs such as custom robotic fish and Slocum gliders to evaluate and test new platforms to perform missions of detecting areas of fish hot spots. During these missions, AUVs typically surface to communicate regularly or respond to forced interruptions. In this area, marine scientists have previously tagged the fish with acoustic transmitters that constantly ping at a pre-determined frequency. These acoustic signals help the AUVs detect fish using their acoustic detection sensors. These AUVs can also classify unique pings, ignoring multiple pings from the same fish.

Before actual deployment, we simulated the marine life survey domain [39] using the Moos-IvP [3] framework. This simulator replicates the underwater environment close to the real world. In addition, simulated environments enable us to empirically evaluate our mechanisms before testing them out in the real world. For simulation, we have divided a portion of the research area (see lower right of Figure 3.1) into twenty-five cells. The red dots represent the thousand fish that pings at every seventeen time steps. The highlighted region around the agent is the acoustic fish tag detection sensor. At Gray's Reef, the detec-

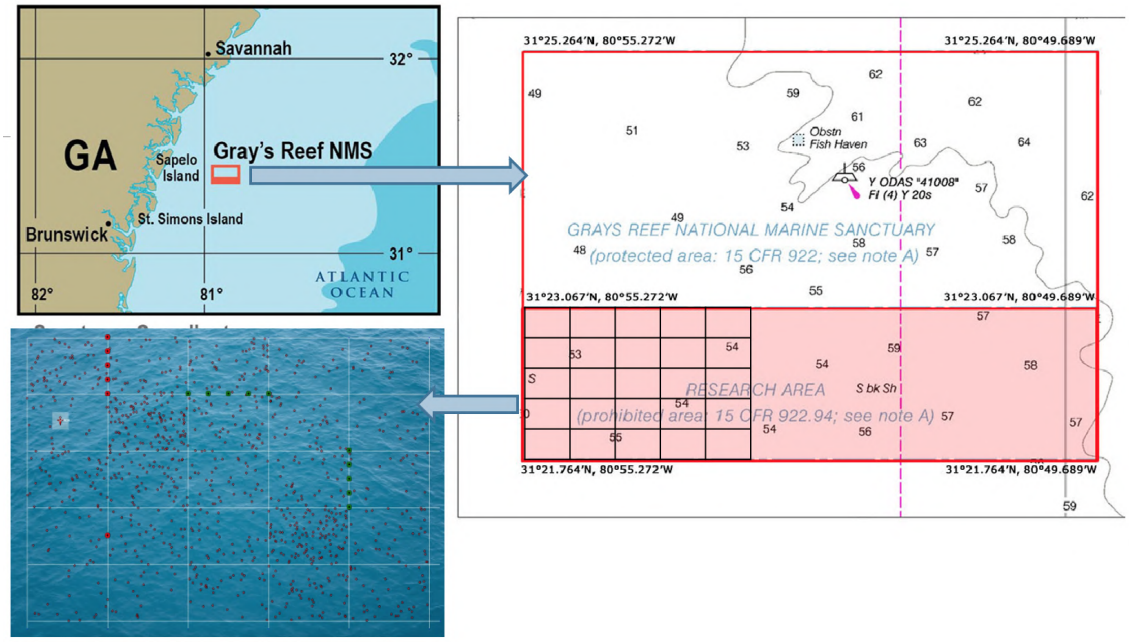

Figure 3.1: Gray's Reef National Marine Sanctuary is located off the coast of Georgia and contains a research area shown in the insert shaded in pink. Within this, we represent a 5x5 subsection in simulation. This grid contains fish hot-spots that are of interest to marine scientists. The highlighted square around the agent indicates the sensor range for detecting acoustic fish tags (the small red dots).

tion radius varies with environmental conditions, but currently, the simulator assumes it to be ten meters. As mentioned, an agent can identify hot spots based on the number of pings.

3.2.1 Experimental Design: Three agent scenario

We have introduced two naturally occurring discrepancies in the marine life survey domain: Remora attacks and flow events. Remora attacks are attachments to the AUV made by sea creatures that hinder movement. They act randomly with a specific rate (0.007) of occurrence. These Remora attacks negatively affect an agent's speed, and enough attacks could disable an individual platform from moving. An agent can successfully respond to a Remora attack by formulating a goal to be free from the organism by gliding backward. Flow events occur at a specific location in the marine life survey domain. These events disable the agent and push them out of the region. In such a case, an agent can only delegate its goals and ask the operator for help to initiate a rescue operation. Note that the agents can only communicate when they surface, and they surface at regular intervals of 10 units in simulation time.

Figure 3.2 shows the 5x5 region of the marine life survey domain. Three agents exist in this multi-agent system, namely Grace, Franklin, and Remus. All agents are provided with initial goals to survey a specific part of the 5x5 region. For example, Grace is responsible for achieving nine survey goals represented as the blue region in the figure. Similarly, Franklin and Remus have eight goals to survey the green and red areas. Furthermore, as shown in the figure, two specific flow-affected cells are at (2,0) and (0,2). When agents survey these flow-affected cells, they are pushed far from the area and are disabled. However, they can delegate goals and call for help. Similarly, there are randomly occurring Remora attacks, as mentioned above. In addition, nine fish hot spots exist in this scenario, represented as the cells with yellow borders in the figure.

Each trial in the above-mentioned experimental setting has different initial starting

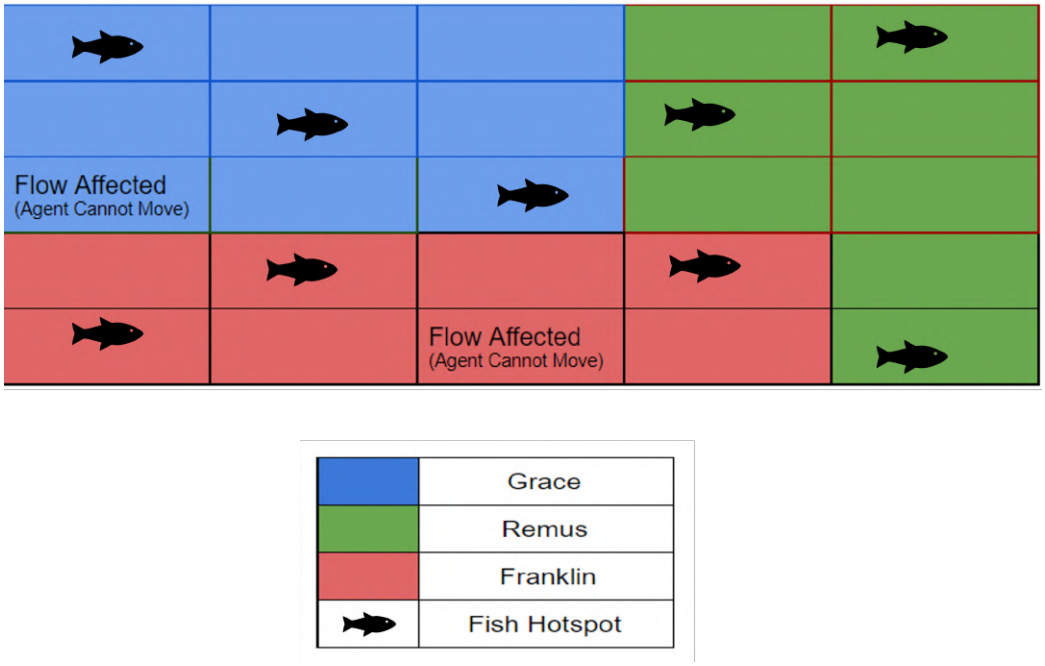

Figure 3.2: Experimental design to evaluate eight different multi-agent conditions in the marine life survey domain [24]. There are three underwater gliders (Franklin, Grace, and Remus) and nine hot spots in the scenario. The hot spot location are evenly distributed between the 3 agents to survey.

positions for the agents. Therefore, we have performed two experiments with varying randomness for each multi-agent condition, and an experiment contains a hundred separate trials. This randomness affects the occurrence of Remora attacks, thereby changing the performance of the multi-agent system as a whole. This experiment aims to measure the performance of a multi-agent system in terms of the percentage of goals achieved and the indentification of fish hot spots. Section 3.2.1.2 presents the statistical significance of the results, followed by the empirical results in section 3.2.1.1.

3.2.1.1 Empirical Results

Figure 3.3 depicts the results of the eight different multi-agent conditions. The X-axis represents the time taken by the agents in these multi-agent conditions to achieve their goals, while the Y-axis represents the percentage of goals achieved. As mentioned in the previous section, an experiment is run twice with different seed values for every condition

and there are hundred initial positions for each experiment. Therefore, every point in the graph averages two hundred trails. Furthermore, every trial has a simulation time limit of six hundred seconds. Note that forty times one simulation time gives the actual time in the real world. If an agent following our approach (goal delegation using theory of mind) is affected by the flow, it decides to delegates its goals to an agent with the minimum planning cost, shares the affected region's knowledge, and explains the motivations behind the goals.

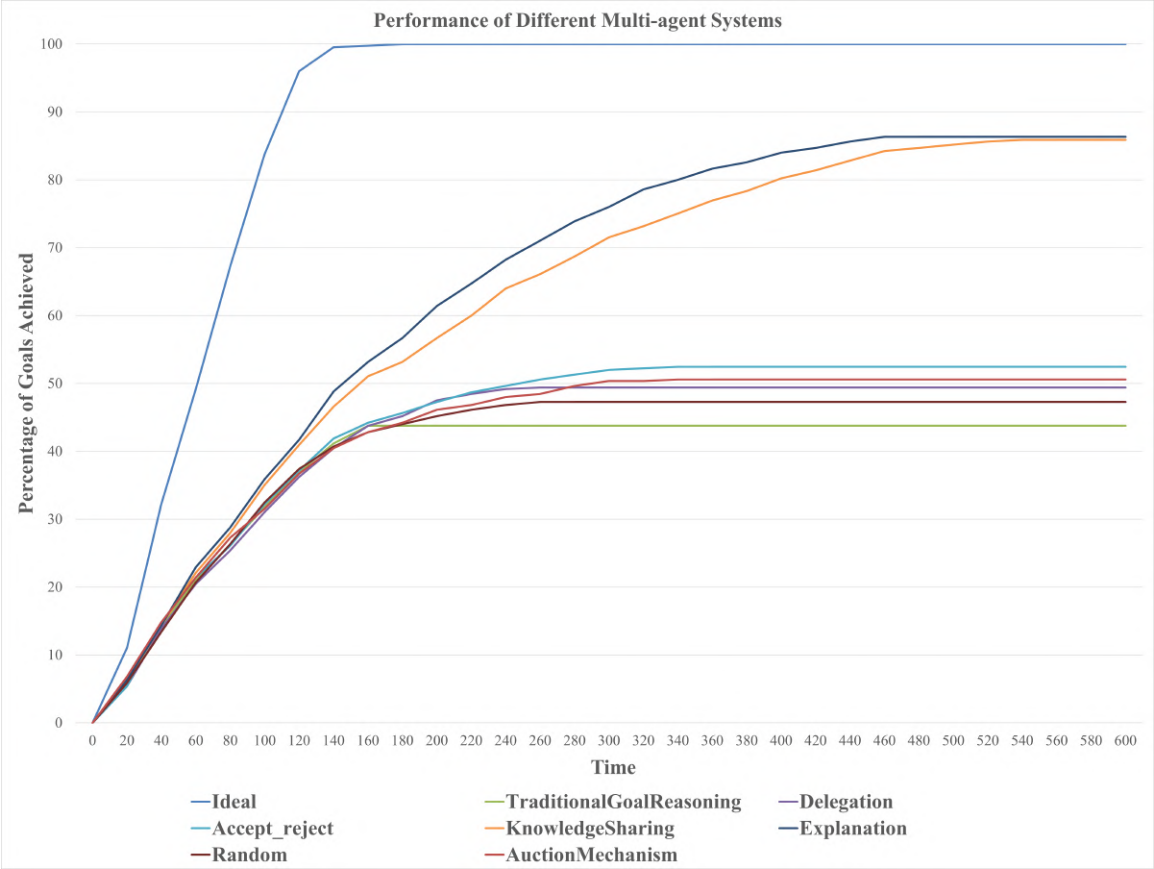

Figure 3.3: Percentage of goals achieved by agents in different multi-agent conditions in the marine life survey domain. On the X-axis is the time to achieve the goals, and on the Y-axis is the percentage of goals achieved.

Overall, the results show that the agents in the multi-agent explanation condition performs better and achieves goals quicker than all the others in the environment except for the agents in an ideal multi-agent condition, which operates in perfect conditions. The crucial function that gave the edge to the explanation condition is the receiving agents ability to

provide motivations behind the delegated goals. In this experiment, the basis is to find the hot spots. When a receiving agent understands this motivation, it prioritizes the given goals in search of the highest fish densities.

Next to explanation is the performance of agents in the multi-agent knowledge sharing condition. Agents in this multi-agent condition can perform as many goals as in explanation given enough time. However, this is not the case with the real world where the missions are often time-limited. Nevertheless, the significant performance of the agents in these multi-agent conditions compared to other multi-agent conditions shows that our approach towards goal delegation using theory of mind has proven effective. Note that our experimental setup does not allow agents in a practical multi-agent conditions to achieve one hundred percent of the goals. Because trials exist where all agents start at the flow-affected regions, no agent will achieve any goals.

Next comes the performance of agents in the multi-agent accept_reject condition, which gives the receiving agent the decision-making ability to accept or reject delegated goals. In this experiment, if the requested agent is also at a flow-effected region, it refuses the goal, thereby helping the requesting agent give its goals to the subsequent agents. However, not sharing the requesting agent's knowledge of flow-affected areas keeps it significantly lower than Knowledge sharing.

Agents in the multi-agent auction mechanism performs close to the agents in the accept_reject condition given enough time. Minimum cost bids help the requesting agent(s) determine the most capable agents and avoid agents affected by the flow. However, it takes at least twenty units of time for successful bidding. The communication overhead thus keeps its performance significantly lower in the interval 160 to 260 time units.

Finally, the observation that the agents in the delegation multi-agent system (with agent selection capability) performing better than the agents in the random and traditional goal reasoning conditions is self-explanatory. Note that at time 140, agents in the multi-agent traditional goal reasoning condition performs better than random and auction mech-

anism because of the communication overhead in delegating goals (agents communicate only when they surface). However, it soon fades out as time increases.

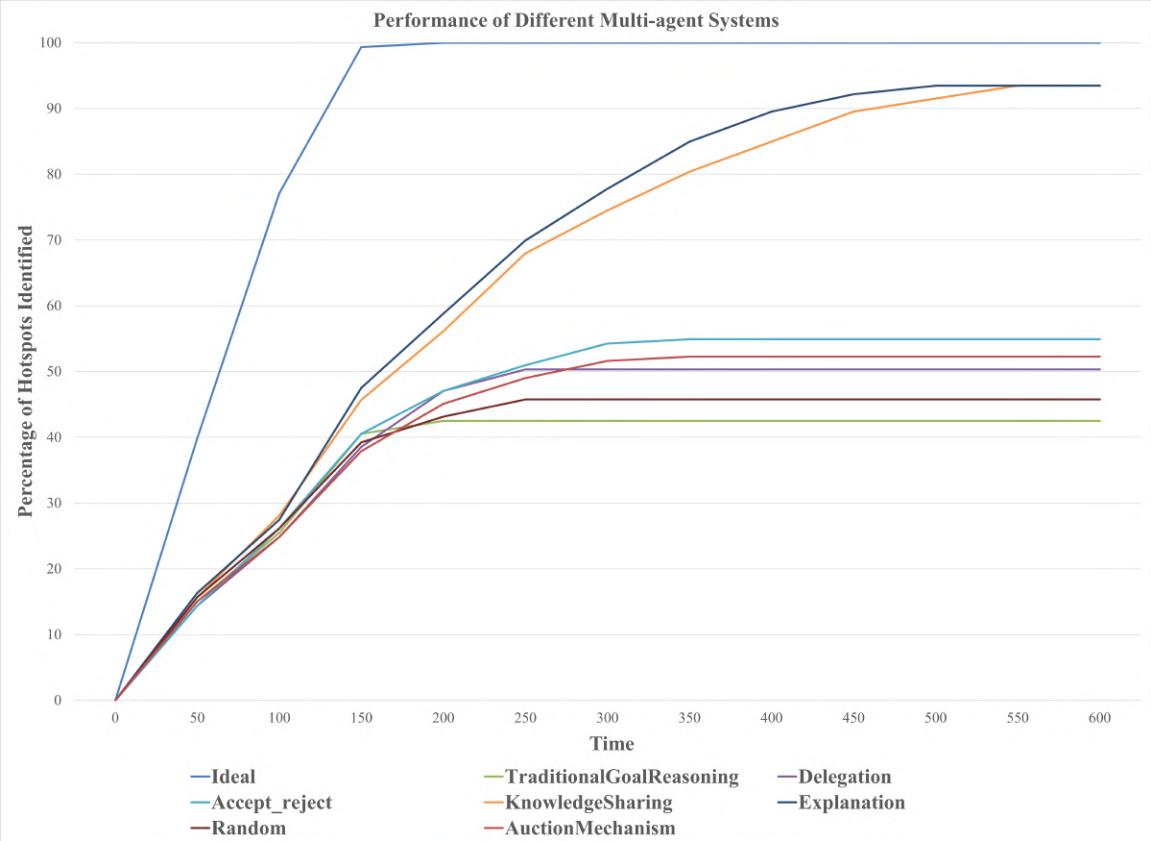

Figure 3.4: Percentage of hot spots identified by agents in different multi-agent conditions in the marine life survey domain. On the X-axis is the time to identify the hot spots, and on the Y-axis is the percentage of hot spots identified.

Figure 3.4 depicts the performance of agents in eight different multi-agent conditions in identifying the percentage of fish hot spots. The X-axis represents the time taken by the agents in the multi-agent conditions to identify the hot spots, while the Y-axis represents the percentage of hot spots identified. As mentioned in the previous section, there are nine hot spot locations.

Similar to the previous graph, the general pattern remains the same. Agents in the multi-agent explanation and the knowledge sharing conditions achieve 93% of the goals thus proving the effectiveness of goal delegation using theory of mind. At the same time,

agents in the agent_select condition reaches 55% of hot spots followed by the agents in the auction mechanism at around 52%. Agents in the delegation condition achieves 50% of hot spots proving that delegation alone is not sufficient. At the lower tier, we have random at 45%, and agents in the traditional goal reasoning condition identifies 41% of the hot spots for 600 units in simulation time.

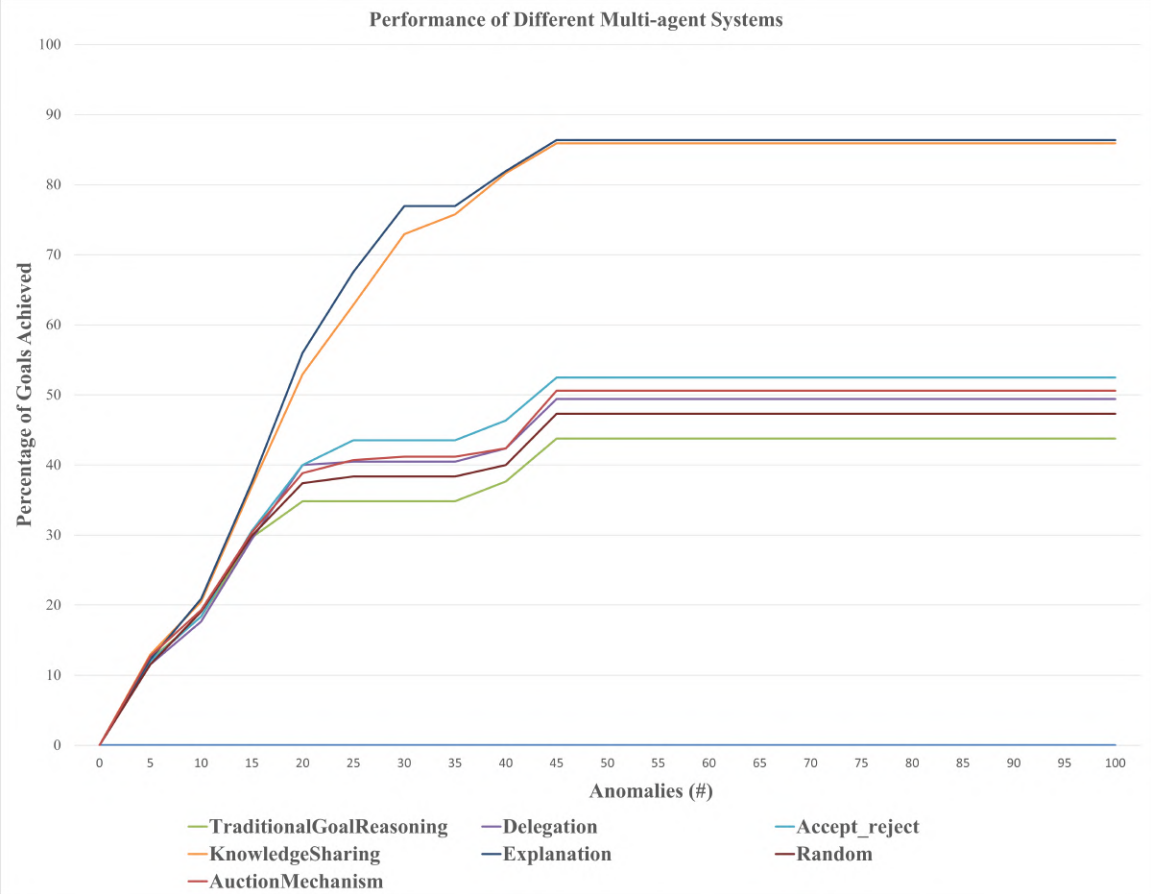

Figure 3.5: Percentage of goals achieved by the agents in different multi-agent conditions in the marine life survey domain. On the X-axis is the number of anomalies and on the Y-axis is the Percentage of goals achieved.

Figure 3.5 depicts the performance of eight different multi-agent conditions in identifying the percentage of goals over anomalies. The X-axis represents the number of anomaly occurrences in the environment, while the Y-axis represents the percentage of goals achieved. Anomalies in this domain include remora attacks and flow events. An

agent formulates the goals to be free from the remora in response to the anomaly and delegates goals in the case of the flow event.

Similar to the graph from 3.3, the general pattern remains the same. Agents in the multi-agent explanation and the knowledge sharing multi-agent conditions achieve 88% of the goals proving the effectiveness of goal delegation using theory of mind. At the same time, agents in the agent_select condition reaches 53% of goals followed by the auction mechanism agents' performance at around 51%. Agents in the delegation condition achieves 50% of hot spots proving that goal delegation alone is not sufficient. At the lower tier, we have the performances of agents in multi-agent random condition at 47%, and traditional goal reasoning at 43% respectively.

3.2.1.2 Statistical Significance

We have performed a two sample t-test to compare our complete approach (multi-agent explanation condition) with all the other multi-agent conditions at two different time points (three hundred and six hundred) in the experiments for statistical significance. Table 3.1 shows the statistical performance of different multi-agent conditions (introduced in section 3.1) over two hundred trials.

Table 3.1: The table shows the statistical distribution of agents' performance in different multi-agent conditions for the three agent scenario of the marine life survey domain for 200 trials.

Multi-agent System	Time	N	Mean	Std. Deviation
Traditional goal reasoning	300	200	43.8	19.68
	600	200	43.8	19.68
Delegation	300	200	49.47	24.47
	600	200	49.47	24.47
Accept_reject	300	200	51.95	22.93
	600	200	52.41	23.41
Knowledge sharing	300	200	71.57	18.77
	600	200	86	17.9
Explanation	300	200	76.12	17.85
	600	200	86.47	16.71
Random	300	200	47.32	24.27
	600	200	47.32	24.27
Auction mechanism	300	200	50.37	24.07
	600	200	50.61	24.47

We have chosen a 95% confidence interval (i.e., we can reject the null hypothesis if the $p - value < 0.05$). The null hypothesis and alternate hypothesis for the independent samples t-test are as follows,

H_0 : There is no significant performance change between the agents in the multi-agent explanation condition and the agents of the other multi-agent condition in comparison.

H_a : Significant statistical difference does exist between our approach and the agents operating in the other multi-agent condition.

Table 3.2: Results from the two-sample independent t-test, shows that the performance of the agents in the multi-agent explanation condition significantly differs from the agents' performances in other multi-agent conditions for the three agent scenario.

Multi-agent System	Time	Explanation	
		p-value	t-value
Traditional goal reasoning	300	<0.0001	17.2
	600	<0.0001	23.37
Delegation	300	<0.0001	12.44
	600	<0.0001	17.65
Accept_reject	300	<0.0001	11.76
	600	<0.0001	16.74
Knowledge sharing	300	0.0134	2.48
	600	0.786	0.27
Random	300	<0.0001	13.5
	600	<0.0001	18.78
Auction mechanism	300	<0.0001	12.15
	600	<0.0001	17.11

Table 3.2 shows the list of p-values between agents in the multi-agent explanation condition and other multi-agent conditions. We can see that the $p - value < 0.0001$ for all other multi-agent conditions except in the knowledge sharing multi-agent condition. Thus, we can reject the null hypothesis. Now, in the case of multi-agent knowledge sharing condition, at time six hundred, agents in both conditions have similar performance in achieving the goals. However, in the multi-agent explanation condition, agents achieve goals much quicker than in knowledge sharing condition, which we can see from the p-

value of 0.0134 at time three hundred. Thus, our data suggest a statistical significance in performance between our approach and the performance of the agents in the rest of the multi-agent conditions.

3.2.2 Experimental Design: Four agent scenario

This experimental design is similar to the setup from the three agent scenario in section 3.2.1.1, but here there are four agents. Figure 3.6 shows the 5x5 region of the marine life survey domain, shared by the agents Grace, Franklin, Remus and Neo to survey their given goals. For example, Neo is responsible for achieving seven survey goals represented as the yellow region in the figure. Similarly, Franklin, Remus and Grace share six goals each to survey the red, green and, blue areas.

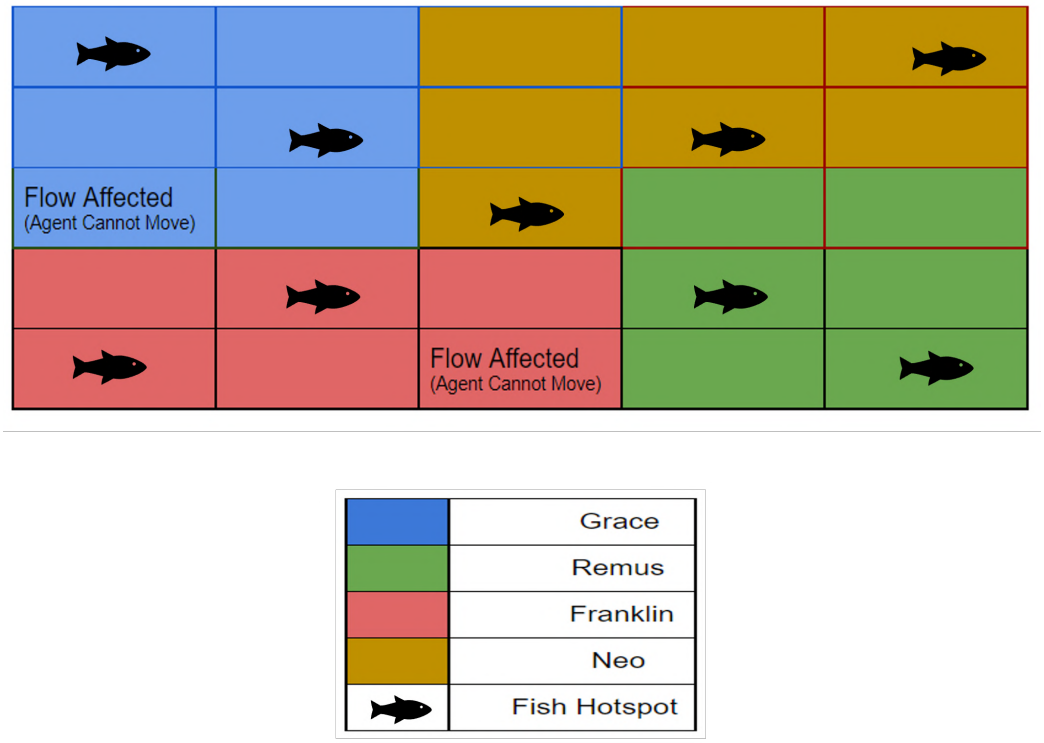

Figure 3.6: Experimental design to evaluate eight different multi-agent systems in the marine life survey domain. There are four underwater gliders (Franklin, Grace, Remus, and Neo) and nine hot spots in the scenario.

Furthermore, as shown in the figure, two specific flow-affected cells are at (2,0) and (0,2). When agents survey these flow-affected cells, they are pushed far from the area and are disabled. However, they can delegate goals and call for help. Similarly, there are randomly occurring Remora attacks, as mentioned above. In addition, similar to the three agent scenerio there are nine fish hot spots.

The main advantage of having the four agent scenario is to see if any disparities exist between the performances of agents in the eight different multi-agent conditions. Experiments, trials, and anomalies remain unchanged from the three agent scenario experiments in the previous section. The following sections discuss the statistical significance and their empirical results in detail.

3.2.2.1 Empirical Results

Figure 3.7 depicts the results of the eight different multi-agent systems. The X-axis represents the time taken by the agents in the various multi-agent conditions to achieve their goals, while the Y-axis represents the percentage of goals achieved. As in the previous section, an experiment is run twice with different seed values for each condition. There are one 100 different initial positions for the agents. Therefore, every point in the graph averages two hundred trails. Furthermore, every trial has a simulation time limit of six hundred seconds.

Overall results show a similar pattern of results as the three agent scenarios. However, because there are four agents, they can achieve the goals much quicker and with a little higher percentage of goals achieved than the three-agent scenario.

Agents in the multi-agent explanation and the knowledge sharing conditions achieve greater than 90% of the goals thus proving goal delegation's effectiveness using theory of mind. At the same time, agents in the accept_reject reaches 70% of goals followed by the performance of agents in auction mechanism that achieves around 63%. Finally, agents

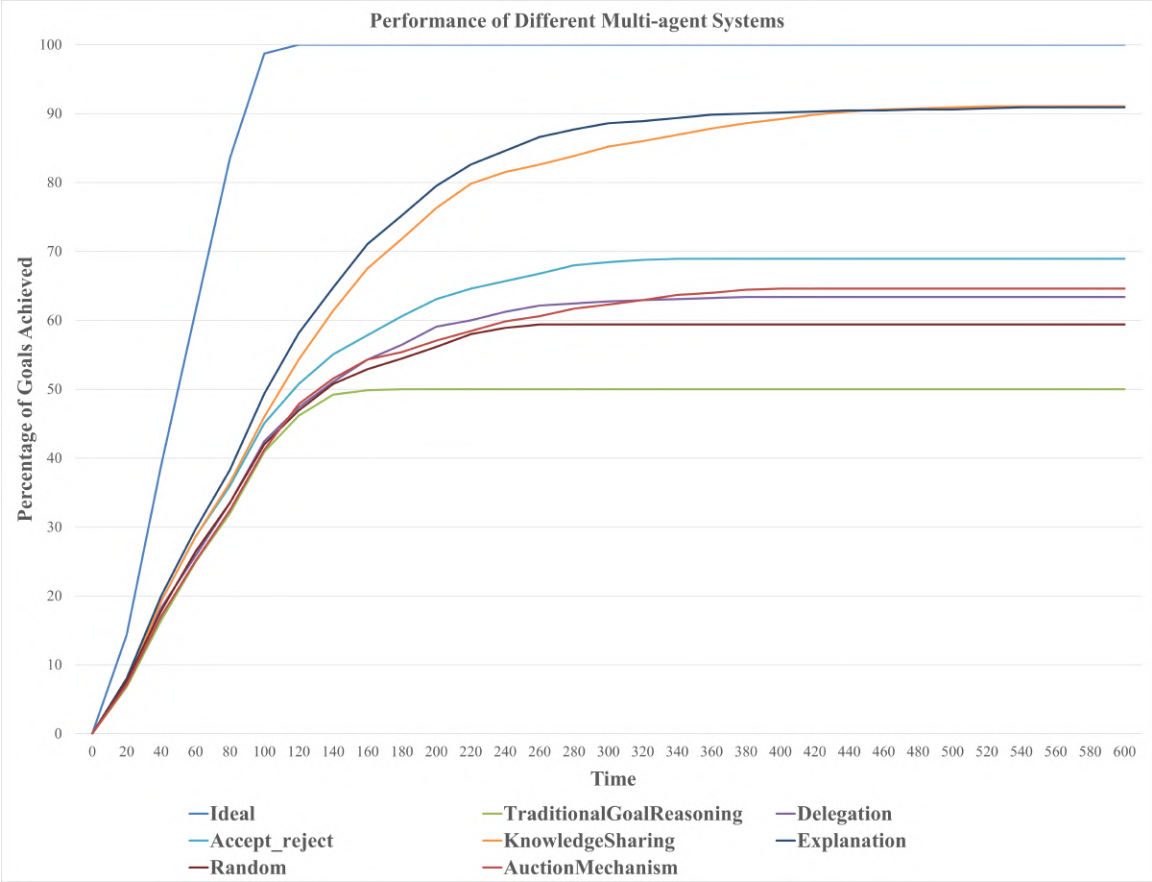

Figure 3.7: Percentage of goals achieved by agents in different multi-agent conditions for the four agent scenario of the marine life survey domain. On the X-axis is the time to achieve the goals and on the Y-axis is the percentage of goals achieved.

in the delegation condition achieves 62% of goals proving that goal delegation alone is insufficient. As the number of agents increases, the disparity between the performances of the agents in the multi-agent delegation and Auction mechanism conditions decreases. At the lower tier, we have the agents in the random condition achieving 61% of goals and traditional goal reasoning condition performing around 50% respectively.

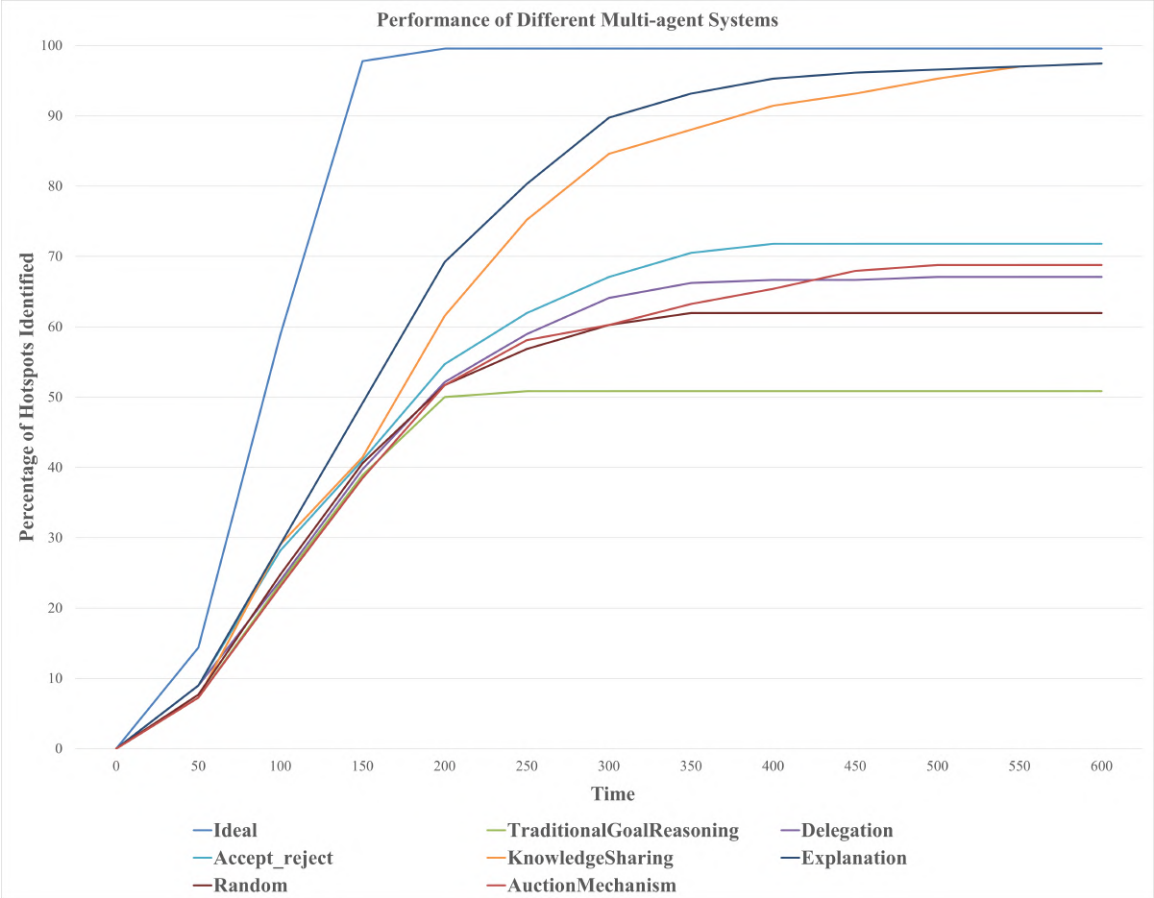

Figure 3.8: Percentage of hot spots identified by the agents in the different multi-agent conditions for the four agent scenario of the marine life survey domain. On the X-axis is the time to identify the hot spots, and on the Y-axis is the percentage of hot spots identified.

Figure 3.8 depicts the performance of agents in different multi-agent conditions in identifying the percentage of fish hot spots. The X-axis represents the time taken by the agents in the multi-agent conditions to identify the hot spots, while the Y-axis represents the percentage of hot spots identified. As with the previous section, nine hot spot locations

exist.

Similar to the previous graph, the pattern remains the same. Agents in the multi-agent explanation and the knowledge sharing conditions identify 98% of the hot spots, once again proving goal delegation's effectiveness using theory of mind. At the same time, agents in the accept_reject condition discovers 72% of hot spots followed by performance of agents in the auction mechanism reaching 70% mark. Agents in the delegation condition identifies 68% of hot spots thus proving that goal delegation alone is not sufficient. At the lower tier, we have agents in the random condition identifying hot spots at 62% and the performance of agents in the traditional goal reasoning condition at 50% respectively for the given time.

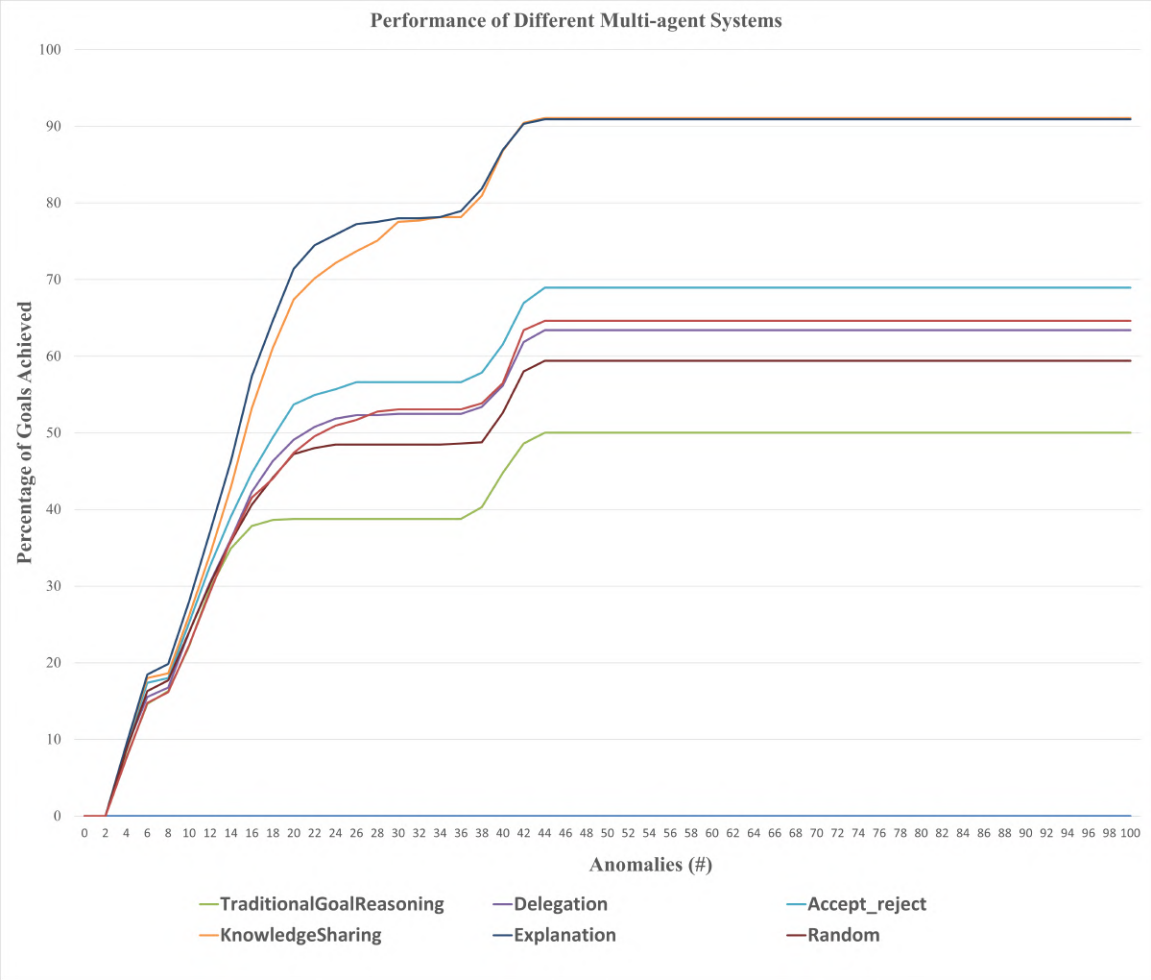

Figure 3.9: Percentage of goals achieved by agents in the different multi-agent conditions for the four agent scenario of the marine life survey domain. On the X-axis is the number of anomalies and on the Y-axis is the percentage of goals achieved.

Figure 3.9 depicts the performance of agents in different multi-agent conditions in identifying the percentage of fish hot spots. The X-axis represents the number of anomaly occurrences in the environment, while the Y-axis represents the percentage of goals achieved. Anomalies in this domain include remora attacks and flow events. An agent formulates goals in response to the remora anomaly and delegates goals in the case of the flow event.

Similar to the graph from 3.7, the pattern remains the same. Agents in the explanation and the knowledge sharing multi-agent condition achieve greater than 90% of the goals proving the effectiveness of goal delegation using theory of mind. At the same time, agents

in Accept_reject condition reaches 70% of goals followed by the performance of agents in the auction mechanism condition at 63%. Furthermore, agents in the delegation condition achieves 62% of goals proving that goal delegation alone is not sufficient. At the lower tier, we have random at 60% goal achievement and agents in the traditional goal reasoning achieving less than 50% of goals.

3.2.2.2 Statistical Significance

Similar to the previous section, we have performed a two sample t-test to compare our complete approach (Explanation) with all the other multi-agent conditions at two different time points (three hundred and six hundred) for statistical significance. Table 3.3 shows the statistical performance of agents in different multi-agent conditions (see section 3.1) over two hundred trials.

Table 3.3: The table shows the statistical distribution of the performance of agents in different multi-agent conditions for the four agent scenario of the marine life survey domain for 200 trials.

Multi-agent System	Time	N	Mean	Std. Deviation
Traditional goal reasoning	300	200	49.88	22.79
	600	200	49.88	22.79
Delegation	300	200	62.93	27.98
	600	200	63.57	28.45
Accept_reject	300	200	68.48	24.08
	600	200	68.96	24.1
Knowledge sharing	300	200	85.32	11.78
	600	200	90.88	4.74
Explanation	300	200	88.64	7.69
	600	200	91.04	4.71
Random	300	200	59.4	26.9
	600	200	59.4	26.9
Auction mechanism	300	200	62.36	23.98
	600	200	64.68	25.81

We have chosen a 95% confidence interval (i.e., we can reject the null hypothesis if the $p - value < 0.05$). The null hypothesis and alternate hypothesis for the independent samples t-test are as follows,

H_0 : There is no significant performance change between the agents in the multi-agent explanation condition and the others in comparison.

H_a : Significant statistical difference does exist between our approach and the performance of agents in other multi-agent conditions.

Table 3.4: Results from the two-sample independent t-test, showing that the agents in the multi-agent explanation condition significantly differs from all the other multi-agent conditions for the four agent scenario.

Multi-agent System	Time	Explanation	
		p-value	t-value
Traditional goal reasoning	300	<0.0001	22.78
	600	<0.0001	25.01
Delegation	300	<0.0001	12.53
	600	<0.0001	13.47
Accept_reject	300	<0.0001	11.27
	600	<0.0001	12.71
Knowledge sharing	300	0.0009	3.33
	600	0.1600	0.33
Explanation	300		
	600		
Random	300	<0.0001	14.78
	600	<0.0001	16.38
Auction mechanism	300	<0.0001	14.75
	600	<0.0001	14.2

Table 3.4 shows the list of p-values between the performance of agents in the multi-agent explanation condition and agents operating in the other multi-agent conditions. We can see that the $p - valueis < 0.0001$ for all the multi-agent conditions except for the multi-agent knowledge sharing condition. Thus, we can reject the null hypothesis. Now, in the case of knowledge sharing condition, at time six hundred, they both have similar performance in achieving the goals. However, in the explanation condition, agents achieve goals much quicker than the agents in the knowledge sharing condition, which we can see from the p-value of 0.0009 at time three hundred. Thus, our data suggest a statistical

significance in performance between the agents in our approach and the rest of the agents in the multi-agent conditions.

3.3 The Rovers Domain

The rovers domain is a simplified problem representation of the NASA Mars Exploration Rover missions launched in 2003. As the name suggests, agents in this mission are a collection of rovers equipped with different sensors. They must travel between various waypoints collecting data and transmitting it back to a lander. Typical goals of the agent involve traversing the planet's surface to perform science gathering experiments such as soil sample analysis, rock sample analysis, capturing photographs of different objectives. Figure 3.10 shows the conceptual diagram of the Opportunity rover equipped with various tools. These tools include cameras with different resolutions, soil sample equipment, and rock sample equipment.

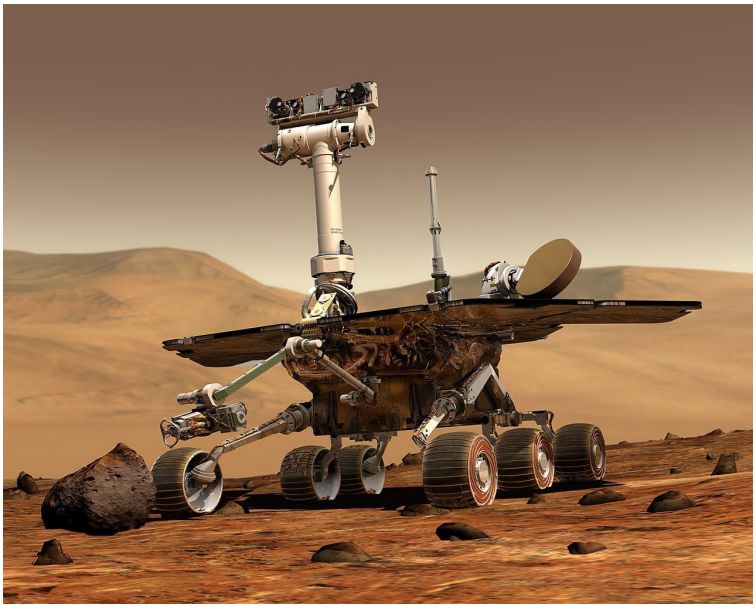

Figure 3.10: Artistic concept of a Mars Exploration Rover (MER) from December 2002, designed to perform geological experiments on planet Mars.

These rovers have different capabilities to perform the mission, sometimes overlap-

ping with each other. Several challenges make their goals difficult to achieve. For example, some rovers cannot traverse specific waypoints. Similarly, not all rovers possess the tools to achieve their mission, thus making it a good problem domain for multi-agent coordination. In addition, exogenous events exist that damage the rovers' sensors. Furthermore, data transmission is also constrained by the visibility of the lander from the waypoints. Because of the complexity in the domain, it is one of the benchmark problem domains in the planning competitions held at International Conference on Automated Planning and Scheduling (ICAPS)[1] every two years [6].

3.3.1 Experimental Design

For the Rover domain, we have obtained hundred problem sets from the Competition of Distributed and Multi-agent Planners (CoDMAP)[2] track held at the international planning competition. From the hundred, forty problem sets have been actually in the competition to evaluate different planners, while the remaining sixty are from the problem generator used by the competition. Each problem contains a different number of goals and agents. At the start of each trial, we have uniformly and randomly allocated these goals to the agents. However, since every agent does not possess the required tools to achieve their assigned goals, they can delegate them to other agents. In addition, we have introduced events that randomly damage the equipment of these agents. Equipment in this domain includes high and low-resolution cameras and tools to perform soil/rock sample analysis.

Figure 3.11 shows the distribution of agents from the problem set. On the X-axis are the number of agents and on the Y-axis are the number of problems/trials containing these agents. Our trials include a maximum of ten distributed agents and a minimum of one agent operating in the domain.

For every multi-agent condition described in section 3.1, we have performed three

[1] https://www.icaps-conference.org/competitions/
[2] http://agents.fel.cvut.cz/codmap/

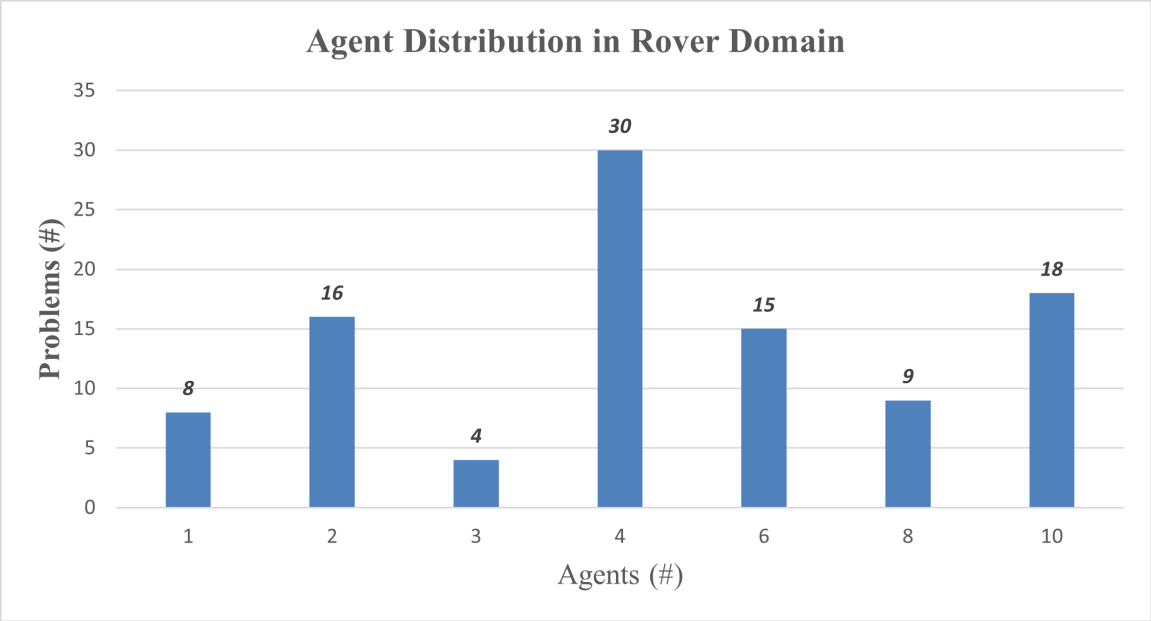

Figure 3.11: Distribution of agents across the 100 benchmark problems sets in the rovers domain.

experiments with the same hundred trials mentioned above. However, the experiments vary in the randomness of the initial goals allocated and the occurrence of exogenous events that damage tools. In the next section, we will see how the results obtained are similar to the results from the marine survey domain.

3.3.2 Empirical Results

Figure 3.12 depicts the results of the agents in the eight different multi-agent conditions. The X-axis represents the time taken by agents in the the multi-agent conditions to achieve their goals, while the Y-axis represents the percentage of goals achieved. As mentioned in the previous section, a trail is run three times with different seed values for agents in every condition. There are a hundred trials for each experiment, and each trial contains n different number of goals and agents. Therefore, every point in the graph averages three hundred trials. Note that the units for time here are MIDCA cycles. As mentioned previously, MIDCA is a cognitive architecture providing the goal reasoning framework for

implementing goal delegation. If an agent following our approach (goal delegation using theory of mind) cannot pursue its assigned goals due to the reasons mentioned above, it decides to delegates its goals to an agent with the minimum planning cost. It shares the knowledge that it lost its sensor traversing a waypoint, and explains the motivations behind the goals.

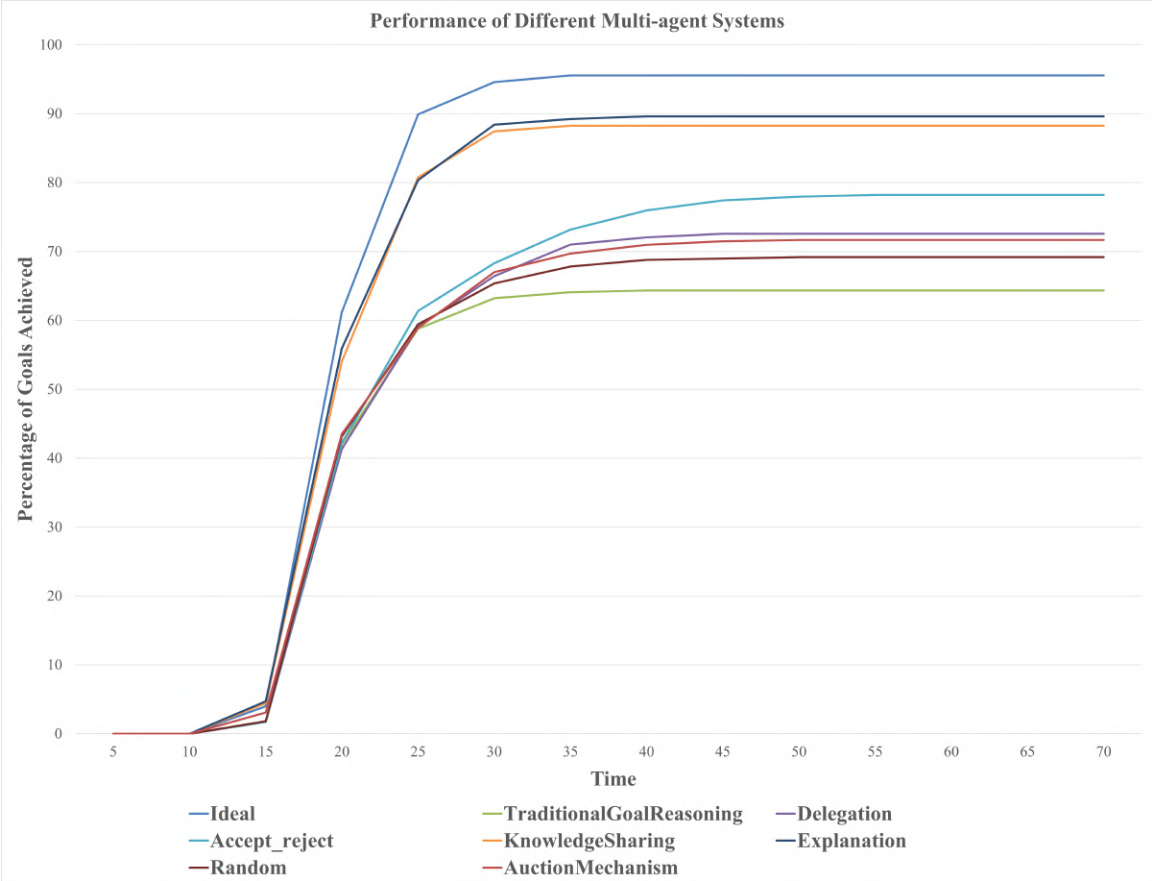

Figure 3.12: Percentage of goals achieved by agents in different multi-agent conditions in the rovers domain. On the X-axis is the time to achieve the goals, and on the Y-axis is the percentage of goals achieved.

All agents are provided with initial goals at ten simulation time units (See figure 3.12) and will remain in operation until the deadline of hundred time units. Although the results show a similar pattern compared to the results from the marine life survey domain, a few key observations are to be made.

Agents in the multi-agent ideal condition do not achieve all their goals. Since these

benchmark problems evaluate planners, some goals are too hard to solve. In our work, we use the fast-downward planner [29] to plan for the goals. Note that some problems are too easy to solve for the agents, which is why we see agents in the traditional goal reasoning condition achieving almost 63% of goals in the given time. Furthermore, agents in the delegation condition perform slightly higher than the agents in the auction mechanism condition because of the communication overhead during agent coordination.

Irrespective of the disparities between complex and straightforward problems, agents in the multi-agent explanation and the knowledge sharing conditions achieve close to 90% of the goals proving again the effectiveness of goal delegation using theory of mind. At the same time, agents in the accept_reject condition reaches 79% of goals followed by the performance of agents in the delegation at around 73% of goals proving that goal delegation alone is not sufficient. Finally, agents in the multi-agent auction mechanism achieve around 72% of the goals, and at the lower tier, we have the performances of agents in the random conditions at 67% goal achievement and traditional goal reasoning condition at 63%.

In summary, the results here and those from 3.2 support the main hypothesis (page 3) that theory of mind and explanations are essential for effective goal management in multi-agent systems.

Related Research

In this chapter, we offer a literature review of related concepts and algorithms that either stand pivotal in shaping this research or stand similar in solving problems that resemble our own. We review the following topics: multi-agent systems, which covers concepts and implementation of different types of multi agent systems in addition to the different coordination mechanisms involved; theory of mind, which discusses several existing theories to develop reasoning about other agents from both psychological and computational perspectives; goal reasoning, which covers the idea of where goals come from and methods of goal management; and explanation, which discusses the concept of explanations in understanding unknown events.

4.1 Multi-agent Systems

There are three different classifications of systems in which multiple autonomous agents work together. They are: centralized, decentralized and hybrid [56]. Centralized multi-agent systems [41] represent a central architecture between agents, which implies that the agents are assumed to be cooperative and benign during the problem solving process. In contrast, decentralized multi-agent systems represent [60] autonomous control of agents, which are assumed to be independent, cooperative or competitive, depending on the situation they experience. Hybrid multi-agent systems represent agents following a combination of both decentralized and multi-agent systems. We are more interested in decentralized

multi-agent systems, as they allow other agents to be added to the system easily. For the purpose of this discussion, when we refer to multi-agent systems we are exclusively referencing decentralized systems. Wooldridge and Jennings [60] presents a formalized approach for multi-agent systems to pursue goals in a Cooperative Problem-Solving process (CPS). To recognize and achieve a multi-agent problem, this approach follows a four-step process. Namely: (1) recognition, to recognize if an agent cannot solve a problem individually; (2) team formation, to select a group of agents that can be expected to solve the problem; (3) plan formation, to come up with a plan that is agreed between the agents to solve the problem (involves negotiation for agreement between agents); and (4) team action, to perform the actions by the agents. However, multi-agent coordination remains a central problem in the steps following recognition.

There are broadly two main coordination mechanisms in multi-agent systems. One is called direct coordination mechanism and the other is indirect coordination. In the direct coordination mechanism, an agent deliberately interacts with other agents. In indirect coordination mechanism, an agent does not directly interact with other agents. However, it either uses environmental cues or specific sign signals to coordinate. Fierro and colleagues [21] presented an experiment for indirect coordination mechanism, in which a group of mobile robots maintain a specified formation to perform search and rescue operations. Whenever an obstacle arose these agents observed their neighboring agents to reorder and maintain one of the specified formations. Stone and Veloso [55] also presented the idea of locker room agreements where the agents agree on certain set of rules before acting on the environment. While acting, the agents follow rules based on the observation of environmental cues. This idea is implemented in the Robocup domain, where agents coordinate to play soccer.

Direct coordination mechanisms can be further classified into signal broadcasting, auction mechanisms, negotiations and argumentation. In signal broadcasting [32, 47], agents send messages to any agents within a specified range (e.g., announcements in an

airport). In an auction mechanism [19, 22], a single agent or a group of agents can bid on either a single goal or multiple goals. In negotiations [20, 53], agents interact with each other and can go back and forth on bids. Argumentation [49], is similar to negotiations where agents provide reasons for their negotiations. This thesis has shown the importance of such justifications and reasons when coordinating, but we have also demonstrated the limitation of auction mechanisms alone.

4.2 Theory of Mind

Theory of mind is an ability of an agent to infer other agents' goals, beliefs, emotions and intentions. There are mainly two broad theories which are widely accepted from a psychological stand point. Theory Theory [28], states that children hold a naïve psychological theory to infer goals, beliefs and emotions of others. This knowledge is used to predict behaviors of others. Moreover, as children grow up they develop their psychological awareness to better infer mental states of others. The Simulation Theory of Empathy [27], states that children simulate the actions of others to predict the behavior of others. Furthermore, there are several hybrid theories that include a part of Theory Theory and the Simulation Theory of Empathy, some of which are the intentional stance and Structure-Mapping theory of analogy. Intentional stance [4, 17] is a concept of theory of mind that states that an agent can predict other agent's beliefs and desires given the other's purpose and place in the world. *Structure-Mapping theory (SMT)* [2] of analogy is a theory of analogy and similarity. SMT focuses on humans' ability to see structural similarities across dissimilar cases. From a computational standpoint, Rabkina and colleagues [48] propose that an agent can better recognize goals of other agents by externally observing their actions using an implementation of the Analogical Theory of Mind. This implementation involves retrieving mapped structures of internal knowledge about the observable actions, thereby predicting the goal using the retrieved structures. The internal knowledge is therefore trained.

4.3 Goal Reasoning

Goal Driven Autonomy (GDA) is introduced in INTRO [7]. This work was motivated to find the reason behind the origin of goals. This system was implemented using the Wumpus world domain. The goal of the agent is to reach a destination while avoiding the pits in the world. Later on, the concept of GDA was incorporated into a cognitive architecture called MIDCA [46] and subsequently the work was extended into the arsonist domain [45]. Goal management has been a key focus of goal reasoning research, hence GDA allows the agent to dynamically perform certain goal operations: selection, monitoring, transformation, formulation and several others (see [39, 38]). The work of Kondrakunta [34, 35, 36] presents a goal selection strategy to look at the cost-benefit ratio during goal selection. This work closely aligns with one of our recent selection strategies [26]. Our recent work also tries to improve the selection strategy with the help of resource estimation and priority functions for goals.

Similarly, Dannenhauer and colleagues [16] 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 or drops the goal. This paper did not consider the problem of selecting goals when there are multiple goals to achieve however. Moreover, the preconditions are mostly rule based rather than any kind of functional estimation, which is often a problem when the agent has very limited resources at hand. The work in [34] also presents an initial implementation of goal change using predicate transformations. The authors Cox and colleagues [12] presented the idea to implement other goal operations like change and formulation. A formalism of how to implement the two operations was outlined in the paper. Finally, for the work on agents generating their own goals when problems are encountered, see [40, 37], Each of these publications distinguish between an anomaly and a problem and they come up with a process for the agent to generate its own goals. GDA is also implemented in a second architecture called ARTUE [33], which presents the performance variation of the ARTUE architecture with both benefits and limitations.

Conclusions and Future Research

In our research, we presented an approach for a distributed multi-agent system. The agents in the system work together when unexpected events happen. They follow a theory of mind approach to delegate goals and share the required knowledge. This thesis explicitly develops algorithms to approach goal delegation, agent selection, knowledge sharing, and goal acceptance rejection, and a framework for explanation. These algorithms improve the performance of a multi-agent system when uncertain events are bound to occur, which is often the case in the real world. Furthermore, to support our claims of robustness and generality, we introduced our approach in two different research domains. The data supports our claims that a multi-agent system following our approach outperforms all the real-world multi-agent systems introduced in chapter 3.

We intend to extend this research to incorporate different kinds of explanations and the concepts of usurpation and goal sharing. Explanations can not only help the delegating agent to justify the reasons behind the delegated goals to the selected agent but can also play a major role in the agent coordination process. For example, in the case of a selected agent rejecting the delegated goals, explaining the reasons behind the rejection will help improve the delegating agent's agent selection process.

In this research, we have only talked about uncertain events that negatively affect agent's resources. However, real-world opportunities can also occur, which positively affects the agent's resources. In such a case, an agent that can quickly achieve its goals can also volunteer to achieve other agents' goals, thereby improving the multi-agent system's

performance Overall. Such a process is called goal usurpation. This also occurs when after delegating a goal, an agent decides to take it back.

Moreover, in this research, we assumed that an individual agent is capable enough to achieve its own goals. However, in the real-world an individual agent often requires the help of several other agents to perform a given goal. Such a process is called goal sharing. We want to incorporate goal sharing within our approach by leveraging the concept of *Hierarchical goal networks (HGN)*. HGNs can split a high-level goal into several sub-goals to share with capable agents. Such goal sharing could further improve the performance of a multi-agent system.

In summary, results obtained from both the domains suggest that agents operating in a distributed multi-agent environment can effectively manage their individual and shared goals by following our approach. They decide when and what goals to delegate using our *DetectDelegation* algorithm, whom to delegate using *AgentSelection* algorithm, and *how* to delegate using both *KnowledgeSharing* algorithm and Explanation frameworks.

References

[1] David W Aha. Goal reasoning: Foundations, emerging applications, and prospects. *AI Magazine*, 39(2):3–24, 2018.

[2] Theodore Bach. Structure-mapping: Directions from simulation to theory. *Philosophical Psychology*, 24(1):23–51, 2011.

[3] Michael R Benjamin, Henrik Schmidt, Paul M Newman, and John J Leonard. Nested autonomy for unmanned marine vehicles with moos-ivp. *Journal of Field Robotics*, 27(6):834–875, 2010.

[4] Robert Burke, Damian Isla, Marc Downie, Yuri Ivanov, and Bruce Blumberg. Creature smarts: The art and architecture of a virtual brain. In *Proceedings of the Computer Game Developers Conference*, pages 147–166. Citeseer, 2001.

[5] Lucian Buşoniu, Robert Babuška, and Bart De Schutter. Multi-agent reinforcement learning: An overview. *Innovations in multi-agent systems and applications-1*, pages 183–221, 2010.

[6] Amanda Coles, Andrew Coles, Angel García Olaya, Sergio Jiménez, Carlos Linares López, Scott Sanner, and Sungwook Yoon. A survey of the seventh international planning competition. *AI Magazine*, 33(1):83–88, 2012.

[7] Michael T Cox. Perpetual self-aware cognitive agents. *AI magazine*, 28(1):32–32, 2007.

[8] Michael T Cox. Goal-driven autonomy and question-based problem recognition. In *Second Annual Conference on Advances in Cognitive Systems 2013, Poster Collection*, pages 29–45. Cognitive Systems Foundation, 2013.

[9] Michael T Cox. A model of planning, action, and interpretation with goal reasoning. In *Proceedings of the fourth Annual Conference on Advances in Cognitive Systems*, pages 57–76, 2017.

[10] Michael T Cox. The problem with problems. *To appear in Advances in Cognitive Systems*, in press.

[11] Michael T Cox, Zohreh Alavi, Dustin Dannenhauer, Vahid Eyorokon, Hector Munoz-Avila, and Don Perlis. MIDCA: A metacognitive, integrated dual-cycle architecture for self-regulated autonomy. In *Thirtieth AAAI Conference on Artificial Intelligence*, pages 3712–3718. AAAI Press, 2016.

[12] Michael T Cox, Dustin Dannenhauer, and Sravya Kondrakunta. Goal operations for cognitive systems. In *Thirty-First AAAI Conference on Artificial Intelligence*. AAAI Press, 2017.

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

[14] Dustin Dannenhauer and Hector Munoz-Avila. Raising expectations in gda agents acting in dynamic environments. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, page 2241–2247, 2015.

[15] Dustin Dannenhauer, Hector Munoz-Avila, and Sravya Kondrakunta. Goal-driven autonomy agents with sensing costs. In *Proceedings of the 5th Goal Reasoning Workshop*, 2017.

[16] Zohreh Dannenhauer, Matthew Molineaux, and Michael T Cox. Explanation-based goal monitors for autonomous agents. In *Advances in Cognitive Systems*, pages 1–6. Cognitive Systems Foundation, 2019.

[17] Daniel C Dennett. Taking the intentional stance seriously. *Behavioral and Brain Sciences*, 6(3):379–390, 1983.

[18] Department of Defense. Unmanned systems integrated roadmap fy 2013-2038, 2013. [Online; accessed July 8, 2020].

[19] Bernardine M Dias, Robert Zlot, Marc Zinck, Juan P Gonzalez, and Anthony Stentz. A versatile implementation of the traderbots approach for multirobot coordination. In *Proceedings of the International Conference on Intelligent Autonomous Systems (IAS)*, 2004.

[20] Peyman Faratin, Carles Sierra, and Nicholas R Jennings. Using similarity criteria to make negotiation trade-offs. In *Proceedings Fourth International Conference on MultiAgent Systems*, pages 119–126. IEEE, 2000.

[21] Rafael Fierro, Peng Song, Aveek Das, and Vijay Kumar. Cooperative control of robot formations. In *Cooperative control and optimization*, pages 73–93. Springer, 2002.

[22] Brian P Gerkey and Maja J Mataric. Sold!: Auction methods for multirobot coordination. *IEEE transactions on robotics and automation*, 18(5):758–768, 2002.

[23] Venkatsampath Raja Gogineni, Sravya Kondrakunta, Danielle Brown, Matthew Molineaux, and Michael T Cox. Probabilistic selection of case-based explanations in

an underwater mine clearance domain. In *International Conference on Case-Based Reasoning*, pages 110–124. Springer, 2019.

[24] Venkatsampath Raja Gogineni, Sravya Kondrakunta, and Michael T. Cox. Multi-agent goal delegation. In *Proceedings of the 9th Goal Reasoning Workshop*, 2021.

[25] Venkatsampath Raja Gogineni, Sravya Kondrakunta, Matthew Molineaux, and Michael T Cox. Application of case-based explanations to formulate goals in an unpredictable mine clearance domain. In *Proceedings of the ICCBR Workshop on Case-Based Reasoning for the Explanation of Intelligent Systems*, pages 42–51, 2018.

[26] Venkatsampath Raja Gogineni, Sravya Kondrakunta, Matthew Molineaux, and Michael T Cox. Case-based explanations and goal specific resource estimations. In *Proceedings of Thirty-Third International Flairs Conference*, pages 407–412. AAAI, 2020.

[27] Alvin I Goldman. In defense of the simulation theory. *Mind & Language*, 7(1–2):104–119, 1992.

[28] Alison Gopnik and Henry M Wellman. The theory theory. *Mapping the mind: Domain specificity in cognition and culture*, page 257–293, 1994.

[29] Malte Helmert. The fast downward planning system. *Journal of Artificial Intelligence Research*, 26:191–246, 2006.

[30] Jörg Hoffmann, Julie Porteous, and Laura Sebastia. Ordered landmarks in planning. *Journal of Artificial Intelligence Research*, 22:215–278, 2004.

[31] Pieter Jan't Hoen, Karl Tuyls, Liviu Panait, Sean Luke, and Johannes A La Poutre. An overview of cooperative and competitive multiagent learning. In *Proceedings of the First International Conference on Learning and Adaption in Multi-Agent Systems*, pages 1–46. Springer, 2005.

[32] Nidhi Kalra, Dave Ferguson, and Anthony Stentz. Hoplites: A market-based framework for planned tight coordination in multirobot teams. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1170–1177. IEEE, 2005.

[33] Matthew Klenk, Matthew Molineaux, and David W Aha. Goal-driven autonomy for responding to unexpected events in strategy simulations. *Computational Intelligence*, 29(2):187–206, 2013.

[34] Sravya Kondrakunta. Implementation and evaluation of goal selection in a cognitive architecture. Master's thesis, Wright State University, 2017.

[35] Sravya Kondrakunta and Michael T Cox. Autonomous goal selection operations for agent-based architectures. In *Fifth Goal Reasoning Workshop*, 2017.

[36] Sravya Kondrakunta and Michael T Cox. Autonomous goal selection operations for agent-based architectures. In *Proceedings from Advances in Artificial Intelligence and Applied Cognitive Computing*, Las Vegas, NV, 2021. Springer.

[37] Sravya Kondrakunta, Venkatsampath Raja Gogineni, Danielle Brown, Matt Molineaux, and Michael T Cox. Problem recognition, explanation and goal formulation. *Advances in Cognititve Systems*, 2019.

[38] Sravya Kondrakunta, Venkatsampath Raja Gogineni, and Michael T. Cox. Agent goal management using goal operations. In *Proceedings of the 9th Goal Reasoning Workshop*, 2021.

[39] Sravya Kondrakunta, Venkatsampath Raja Gogineni, Michael T. Cox, Demetris Coleman, Xiaobo Tan, Tony Lin, Mengxue Hou, Fumin Zhang, Frank McQuarrie, and Catherine R. Edwards. The rational selection of goal operations and the integration of search strategies with goal-driven autonomy. In *Proceedings of the Ninth Annual Conference on Advances in Cognitive Systems*. Cognitive Systems Foundation, 2021.

[40] Sravya Kondrakunta, Venkatsampath Raja Gogineni, Matthew Molineaux, Hector Munoz-Avila, Martin Oxenham, and Michael T Cox. Toward problem recognition, explanation and goal formulation. In *Sixth Goal Reasoning Workshop at IJCAI/FAIM-2018*, 2018.

[41] Hector J Levesque, Philip R Cohen, and José HT Nunes. On acting together. In *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90)*, pages 94–99, 1990.

[42] Pei-Chieh Li. Planning the optimal transit for a ship through a mapped minefield. Master's thesis, Naval Postgraduate School Monterey, 2009.

[43] Matthew Molineaux, Matthew Klenk, and David Aha. Goal-driven autonomy in a navy strategy simulation. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[44] Héctor Munoz-Avila, David William Aha, Ulit Jaidee, Matthew Klenk, and Matthew Molineaux. Applying goal driven autonomy to a team shooter game. In *Twenty-Third International FLAIRS Conference*, 2010.

[45] Matt Paisner, Michael T Cox, Michael Maynord, and Don Perlis. Goal-driven autonomy for cognitive systems. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, pages 2085–2090, 2014.

[46] Matt Paisner, Michael Maynord, Michael T Cox, and Don Perlis. Goal-driven autonomy in dynamic environments. In *Goal Reasoning workshop from the ACS*, pages 79–94, 2013.

[47] Lynne E Parker. Lifelong adaptation in heterogeneous multi-robot teams: Response to continual variation in individual robot performance. *Autonomous Robots*, 8(3):239–267, 2000.

[48] Irina Rabkina, Pavan Kathnaraju, Mark Roberts, Jason Wilson, Kenneth Forbus, and Laura Hiatt. Recognizing the goals of uninspectable agents. In *Proceedings of the Eighth Annual Conference on Advances in Cognitive Systems*, 2020.

[49] Iyad Rahwan, Sarvapali D Ramchurn, Nicholas R Jennings, Peter Mcburney, Simon Parsons, and Liz Sonenberg. Argumentation-based negotiation. *Knowledge engineering review*, 18(4):343–375, 2003.

[50] Mark Roberts, Daniel Borrajo, Michael Cox, and Neil Yorke-Smith. Special issue on goal reasoning. *AI Communications*, 31(2):115–116, 2018.

[51] Roger C Schank, Alex Kass, and Christopher K Riesbeck. *Inside Case-based Explanation*. Psychology Press, 1994.

[52] RP Schank. *Explanation patterns: Understanding mechanically and creatively*. Psychology Press, 1986.

[53] Nathan Schurr, Steven Okamoto, Rajiv T Maheswaran, Paul Scerri, and Milind Tambe. Evolution of a teamwork model. *Cognition and multi-agent interaction: From cognitive modeling to social simulation*, pages 307–327, 2005.

[54] Vikas Shivashankar, Ron Alford, Ugur Kuter, and Dana Nau. Hierarchical goal networks and goal-driven autonomy: Going where ai planning meets goal reasoning. In *Goal Reasoning Workshop from the ACS*, pages 95–110, 2013.

[55] Peter Stone and Manuela Veloso. Towards collaborative and adversarial learning: A case study in robotic soccer. *International Journal of Human-Computer Studies*, 48(1):83–104, 1998.

[56] Piotr Szymak. Comparison of centralized, dispersed and hybrid multiagent control systems of underwater vehicles team. In *Solid State Phenomena*, volume 180, pages 114–121. Trans Tech Publications, 2012.

[57] Milind Tambe and Weixiong Zhang. Towards flexible teamwork in persistent teams. *Autonomous Agents and Multi-Agent Systems*, 3(2):159–183, 2000.

[58] Alejandro Torreño, Eva Onaindia, Antonín Komenda, and Michal Štolba. Cooperative multi-agent planning: A survey. *ACM Computing Surveys (CSUR)*, 50(6):1–32, 2017.

[59] Alejandro Torreño, Eva Onaindia, and Oscar Sapena. Fmap: Distributed cooperative multi-agent planning. *Applied Intelligence*, 41(2):606–626, 2014.

[60] Michael J Wooldridge and Nicholas R Jennings. The cooperative problem-solving process. *Journal of Logic and Computation*, 9(4):563–592, 1999.