Designing for Multi-Agent Collaboration:  
A Shared Mental Model Perspective

ABSTRACT

With the growing use of intelligent agents, it is important to know how to design artificial agents that would be capable of working in mixed teams of humans and artificial agents. In this thesis, formal models of teamwork and concepts successful in explaining effective teamwork in human teams are used to design artificial agents to enable the agents to be effective team members. For artificial agents to be effective teammates, they need to reason explicitly with concepts that achieve effective teamwork. To give artificial agents this reasoning ability, it is intuitive to explore teamwork concepts that work well with human teams. After all, humans will be an integral part of the human-agent system, and considering concepts that work well with human teams can provide useful insights when designing artificial agents.

Two important factors that bind human teams together are: 1) a shared understanding of the team, tasks and goals between members; and 2) the interdependence relationships between members. This thesis seeks to explore ways of representing the shared understanding, designing a computational model of the shared understanding, identifying links between the elements of the shared understanding and the interdependence relationships, and enabling artificial agents to make communication decisions using the shared understanding in the context of artificial agent teams.

This thesis uses shared mental models (SMM), which have been used to explain effective teamwork in human teams, as a way of establishing and maintaining the shared understanding between the team members, and proposes a computational model of SMM. The computational SMM consists of five components that represent the SMM knowledge and three types of processes that are required when designing and implementing artificial
agents capable of leveraging SMM in their decision-making processes. Empirical studies demonstrate the success of the proposed SMM in capturing the required knowledge.

A fine-grained analysis of various types of interdependence relationships results in the definitions of task, goal, and agent interdependence. These definitions clearly distinguish between the different forms of interdependence, and examples are used to show that such an analysis can identify the SMM related communication requirements. The results gathered using simulation-based experiments point out the sub-components of the SMM impacted by increasing levels of agent interdependence and provide guidance to agents on what to communicate given particular levels of interdependence. Examples also highlight that there are benefits to system designers as these definitions provide a formal framework in which to consider the different forms of interdependence relationships and resulting design options.

Agents need to know how to determine what, when, and to whom to communicate in order to establish SMM. This thesis proposes two novel and generalisable communication planning approaches, named Temporal-Projection based (TP) and Team-Model based (TM). These communication planning approaches enable agents to make their communication decisions using SMM. Empirical studies demonstrate the success of the two approaches in enabling agents to communicate the different SMM components, and illustrate that the two approaches perform better than hand-crafted communication protocols.
Declaration

This is to certify that:

(i) the thesis comprises only my original work towards the PhD;

(ii) due acknowledgement has been made in the text to all other material used;

(iii) the thesis is less than 100,000 words in length, exclusive of tables, maps, bibliographies and appendices.

Ronal Rajneshwar Singh  
17/05/2018

Date
Preface

This thesis includes materials adapted from the following publications and presentations. I declare that I am the primary author and have contributed more than 50% of each of these papers.

Chapter 3 is based on the paper:


Chapter 5 is based on the paper, which received the best paper award:


Chapter 6 is based on a 4-page short paper (unpublished), which was accepted and presented at the Workshop on Impedance Matching in Cognitive Partnerships at the International Joint Conference on Artificial Intelligence (IJCAI 2017):

To my wife,
Sajhneeta,
for her never ending love, support and encouragement.
Acknowledgments

My PhD journey has been an exciting, amazing and enriching experience, and this thesis would not have been possible without the support, guidance and encouragement of a number of individuals. I would like to extend my sincerest gratitude and appreciation to the following individuals who have guided, supported and encouraged me throughout my study.

First and foremost, I wish to thank my principal supervisor, Professor Liz Sonenberg, for giving me the valuable opportunity to work with her. I express my heartfelt gratitude for her patience, mentorship, and guidance that has enabled me to complete this PhD thesis. She had been an insightful, supportive, and patient mentor.

I would like to express my heartfelt gratitude to my co-supervisor, Associate Professor Tim Miller, for his mentorship, guidance, and patience. He had been equally insightful, supportive and patient.

I also thank the School Administrators for their help and support and taking care of administrative matters regarding my candidature. And special thanks to the fellow PhD colleagues for the coffee times, meals times and side discussions.

Last but not the least, I would like to thank my wife, Sajhneeta, for standing by me for the past four years and encouraging me all the way. This thesis would not have been possible without her understanding and support.
# Contents

1 Introduction

1.1 Motivation ...................................................... 1
1.2 Aim and Questions .......................................... 2
1.3 Contributions .................................................. 4
1.4 Thesis Outline ................................................ 5

2 Background and Related Work

2.1 Artificial Agents and Mental Models ........................................ 8
2.2 Existing Models of Human Agent Interaction (HAI) ......................... 14
2.3 Team Cognition and SMM for Human Teams ............................. 16
2.4 Formal Teamwork Models ........................................ 21
2.5 Conclusion ....................................................... 30

3 Interdependence in Multiagent Systems ................................. 31

3.1 Related Works .................................................. 33
3.2 Running Example ............................................... 38
3.3 Task Interdependence .......................................... 40
3.4 Agent Interdependence ......................................... 46
3.5 Goal Interdependence .......................................... 49
3.6 Example: Analysis of Situation using Interdependence Types ............ 51
3.7 Discussion ........................................................ 52
3.8 Conclusion ....................................................... 53

4 SMM for Artificial Agents: A Computational Model .................. 55

4.1 Related Works .................................................. 55
4.2 A Computational model of SMM .................................. 56
4.3 SMM State Information ......................................... 62
4.4 Task Model ....................................................... 66
## Listing of figures

2.1.1 The procedural reasoning system from Bordini et al. (2007) ........................................... 9  
2.3.1 SMM content proposed by Cannon-Bowers et al. (1993) .............................................. 18  
3.1.1 Interdependence forms from organisation psychology literature. ................................. 34  
3.1.2 Interdependence forms based on social dependence theory. ....................................... 36  
3.2.1 Sample BW4T environment. .......................................................................................... 40  
3.3.1 Four types of tasks form a hierarchy of increasing levels of interdependence. .............. 43  
4.1.1 Shared mental model components prevalent in studies concerning shared mental models for human teams ................................................................. 59  
4.1.2 Computational framework of SMM proposed by Scheutz et al. (2017) ........................ 60  
4.1.3 Simplified teamwork mental model of Scheutz et al. (2017). ........................................... 61  
4.2.1 Shared mental model data structures. ................................................................................ 63  
4.2.2 Artificial agent architecture with SMM components and key processes. ....................... 63  
4.8.1 Key processes required to establish, maintain and use SMM ......................................... 72  
5.1.1 Example of SMM for BW4T domain ................................................................. 82  
5.3.1 Basic joint tasks used to simulate different types of task interdependence. ................. 87  
5.3.2 Abstract decision cycle of an agent. .................................................................................. 91  
5.3.3 BW4T goal hierarchy ....................................................................................................... 94  
5.5.1 Performance and communication cost for different task interdependence types ......... 99  
5.5.2 Influence of SMM Components on Team Performance ............................................. 101  
5.5.3 Relationship between sharedness and performance for sequential tasks ................... 102  
5.5.4 Sharedness and delivery times for a 2-agent team engaged in a team task ................. 103  
5.5.5 Completion times for reciprocal and sequential tasks with joint actions .................... 104  
5.5.6 Sharedness for reciprocal and sequential tasks with joint actions for the two teams (represented by the two bars) ................................................................. 104
6.1 A graphical comparison of the planning and communication activities of the
temporal projection and team model based planning approaches ........................................... 108
6.4.1 Example BW4T problem ........................................................................................................... 121
6.4.2 Example of temporal projection used to identify the best search goals of two
agents in a sample BW4T task ........................................................................................................... 122
6.4.3 MAS Platform and Architecture. The architecture is shown for two agents
but is generic and scales to multiple agents ..................................................................................... 131
6.7.1 Number of messages exchanged by the agents in the BW4T domain ........................................ 136
6.7.2 Task completion times for the BW4T domain ........................................................................... 137
6.7.3 Task execution times for the BW4T domain. This is time for action execution only ................................................................. 138
6.7.4 Task completion times for the Medical Kit Delivery domain ..................................................... 139
6.7.5 Task execution times for the Medical Kit Delivery domain. This is time for action execution only ................................................................. 139
6.7.6 Number of messages exchanged by agents solving the Medical Kit Delivery scenario .................. 140
<table>
<thead>
<tr>
<th>Table Reference</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4.1</td>
<td>Temporal modal logic operators and modal operators for mental states</td>
<td>23</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Dependence or interdependence types and papers defining or explaining these types</td>
<td>38</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Experimental setups (S1 – S6) for each type of task interdependence</td>
<td>97</td>
</tr>
<tr>
<td>6.7.1</td>
<td>Averages of the three measures for the two domains with the standard deviations shown in brackets</td>
<td>135</td>
</tr>
</tbody>
</table>
Introduction

With growing use of intelligent agents, it is important to know how to design artificial agents that would be capable of working in mixed teams of humans and artificial agents. This thesis leverages formal models of teamwork and concepts successful in explaining effective teamwork in human teams as the basis of designing artificial agents to enable the agents to be effective team members. I believe that artificial agents that behave more like teammates than simple tools will be more widely trusted and accepted by humans. This chapter discusses the motivation, aim and questions, contributions, and finally the outline of the thesis.

1.1 Motivation

With the growing use of artificial agents for military and civilian purposes, such as for performing search and rescue tasks, space exploration, and military operations (Chen and Barnes, 2014), teamwork between humans and artificial agents will most certainly continue to increase. Artificial agents may be required to perform tasks that range from independent tasks that do not require interactions with others to highly interdependent tasks requiring close and continuous interactions and a lot of coordination, which in turn requires at least:

- Shared understanding: as has been observed in human teams, when teams are on-the-same-page regarding the team and task, members are able to coordinate better and the team performance is improved (Mohammed et al., 2010).

- Communication: When used effectively, communication enables human teams to achieve a shared understanding of their task and team, which in turn results in effective teamwork (DeChurch and Mesmer-Magnus, 2010b, Mohammed et al., 2019).
One of the concepts that has been successful in explaining effective teamwork in human teams is shared mental models (SMM). SMM has been defined by Cannon-Bowers et al. (1993) as:

“knowledge structures held by members of a team that enable them to form accurate explanations and expectations for the task, and, in turn, coordinate their actions and adapt their behaviour to demands of the task and other team members”.

Given the success of shared mental models in explaining effective teamwork in human teams, it is intuitive to explore it when designing systems for artificial agent teams and human-agent teams. The long-term goal of this research is to design artificial agents that can work in a team with humans as teammates, and not simply for humans. It is thus intuitive to explore teamwork concepts that work well with human teams. After all, humans will be an integral part of the human-agent system, and considering concepts that work well with human teams can provide useful insights when designing human-agent systems requiring teamwork between the members. Research on formal models of teamwork is also informative and provide useful insights on concepts that are necessary for accomplishing effective teamwork. A number of formal models of teamwork addressing the fundamentals of effective teamwork have been proposed (Grant et al., 2005). These models highlight the necessary elements required for effective teamwork, such as beliefs and other propositional attitudes (see for example Bratman (1992), Tuomela and Miller (1988)), as well as commitments (see for example Castelfranchi (1995), Gilbert (2008), Jennings (1993)). In this thesis, I take inspiration from research in these two streams.

1.2 Aim and Questions

While the long-term goal of this research is to design human-agent systems, the scope of this thesis has been limited to identifying the data structures and algorithms that enable artificial agents to use SMM for decision-making while engaged in collaborative activities. Identifying these components (the data structures and algorithms) is an important first step in designing artificial agents that can leverage SMM for decision-making. The empirical studies conducted using teams of artificial agents provide an initial validation of the proposed computational SMM.
The over-arching goal of this thesis is:

**GOAL:** To provide a computational model of SMM that artificial agents can use as a decision-making tool when engaged in interdependent tasks with other artificial agents.

This thesis aims to answer the following questions:

1. What are the core components of SMM that are required when designing and implementing artificial agents and what representational approach for these components enables artificial agents to use SMM for decision-making?

   Researchers have identified a number of components that form the SMM for human teams ([Cannon-Bowers et al., 1993](#)). In addition, various formal models of teamwork ([Grant et al., 2005](#)) identify elements important to artificial agent teams. When designing artificial agents that use SMM as the basis of decision-making, it becomes important to identify the key components of the SMM and to choose a potential representation for the knowledge structures. These knowledge structures form the input to the decision-making algorithms.

2. How does the level of interdependence between agents impact the development and use of SMM?

   Some tasks require very close and continuous interactions and concurrent execution of individual actions, that is, joint actions, while other tasks may not require concurrent execution of actions. This indicates that the agents may have varying levels of interdependence between them. As such, one study explores interdependence in more detail and another explores the relationship between the level of agent interdependence and at least two components of SMM in terms of the team performance.

3. How can SMM be injected into the communication planning process to enable artificial agent teams to establish SMM?

   SMM can have prior knowledge that comes from organisational policies or experience working together with other agents. Knowledge is also accrued by teams as the members engage in on-going team activities. This thesis explores communication algorithms that enable teams to establish SMM. This question is focused on exploring how the agents can use communication to establish SMM.
1.3 Contributions

This thesis makes contributions to the field of multiagent teamwork, and has the potential of extending these contributions to human-agent teamwork. The key contributions are:

1. **A computational model of SMM:** While the two streams of research have the same goal, that is, effective teamwork, this thesis is one of the first to consider both perspectives during the design of a computational model of SMM. That is, there is an effort to compare and contrast the two perspectives and reconcile the two accounts with the aim of proposing a computational model of SMM. Five key components of the SMM to capture the SMM knowledge have been identified: 1) SMM State; 2) Domain Model; 3) Interaction Model; 4) Task Model; and 5) Member Model. In addition, there are three types of processes that are required to establish and maintain SMM. These are: 1) processes that handle initialisation of SMM; 2) processes that handle update and maintenance of SMM; and 3) processes that allow utilisation of SMM in decision-making. While research over the years attempted to integrate SMM into multiagent teamwork, the research presented in this thesis is novel in two ways. First, the SMM proposed in this thesis is computational rather than a semantic model used to understand the concepts of SMM. The research presented in this thesis is augmented with a computational model, making it one of the few works that propose a computational model and provides an implementation. Second, this thesis models and implements all of the key components of the SMM as opposed to various other studies that utilise only some of the key components.

2. **Analysis of interdependence types:** This thesis builds on existing perspectives on interdependence and performs a fine-grained analysis of the different types of interdependence relationships. This endeavour lead to a set of semi-formal definitions of task, goal, and agent interdependence. Analysis of real world situations demonstrate that the fine-grained analysis of interdependence is helpful in identifying the potential communication requirements and that these communication requirements form part of the SMM. This illustrates that there are benefits in performing a fine-grained analysis of interdependence. Empirical studies establish that different levels of interdependence between agents requires agents to communicate different components of SMM in order to achieve effective team performance. This provides guidance to agents on what to communicate given particular levels of interdependence. Also, examples demonstrate that there are benefits to system designers when they are equipped with
a formal framework within which to consider the different forms of interdependence in terms of weighting out different design options.

3. Communication models based on SMM: This thesis proposes two novel planning-based communication approaches: 1) the temporal-projection based; and 2) team-model based. Both approaches employ SMM to decide when, what, and to whom to communicate to. This is, to the best of my knowledge, novel approaches to communication planning. Empirical studies illustrate the success of the two approaches in reducing the amount of communication between agents when compared with a hand-crafted communication policy, and this illustrates the ability of the two approaches in shifting the responsibility for communication planning from the designer to the agents. Empirical studies demonstrate that it is possible for agents to integrate communication planning with planning for other actions via the team-model based planning approach, and I argue that the team-model based planning approach aligns better with SMM.

1.4 Thesis Outline

In Chapter 2, a review of existing teamwork models and shared mental models literature is presented. The aim of this chapter is to discuss teamwork concepts, such as team cognition and its various forms, teamwork requirements, review the relevant aspects of the two perspectives of teamwork and enable a comparison of the two perspectives. Additionally, the basic components of the artificial agents designed and implemented in this thesis are discussed.

Chapter 3 reviews the term interdependence, and its various forms. The aim of this study is to provide not only clear definitions of interdependence and its types but also to provide a formal framework within which to consider the different forms of interdependence relationships. An attempt has been made to unify various perspectives on the concept. The benefits of a fine-grained analysis of interdependence are demonstrated using an example, which is also used to demonstrate the link between SMM and interdependence types. The chapter discusses semi-formal definitions of various types of interdependence, such as task interdependence, goal interdependence, and agent interdependence.

Chapter 4 proposes a computational model of SMM based on existing teamwork models and shared mental models literature. The key data structures and processes are discussed. The data structures enable agents to capture SMM data while the processes enable agents to use SMM to make decisions.
Chapter 5, building on the model proposed in Chapter 4 and the analysis of interdependence in Chapter 3, investigates the relationship between SMM and levels of interdependence between agents. The chapter discusses a way of inducing interdependence between agents via tasks of varying levels of interdependence. The implementation of the agents is discussed together with the simulation environment, scenario, and the results.

Chapter 6 discusses the communication protocols based on SMM. Two approaches to communication planning have been implemented: 1) temporal-projection (TP) based; and 2) team-model (TM) based. The chapter discusses the two planning-based approaches, and presents the details and results of the empirical studies.

Chapter 7 summarises the key findings of this thesis, discusses some limitations, discusses the practical implications, and provides potential future directions for this research.
This chapter reviews literature on SMM related research in the context of human teams, formal models of teamwork, existing models of human-agent systems, and the fundamentals of artificial agents. There are at least two streams of research investigating effective teamwork. The first is by researchers considering human teams, and the other by researchers who have proposed formal models of teamwork. In this chapter, an attempt is made to review literature on teamwork resulting from research in the two streams. As would be evident from the discussions in this chapter, there is considerable overlap of the fundamentals of teamwork between these two streams. Given that the long-term goal is to be able to design artificial agents that can collaborate with humans, a significant amount of content in this chapter comes from SMM related research in the context of human teams. While many concepts discussed in this chapter are taken from theories of human reasoning, the following discussions are meant to be in the context of artificial agents. A thread of research concerning SMM has been conducted using artificial agent teams and human-agent teams. However, discussion of these works will appear in Chapter 4, which proposes the computational model of SMM employed by empirical studies in this thesis.

The design of systems in which humans and agents work together is not a new one. This chapter provides an account of the principles that guided the development of these systems and their limitations, and highlights why these models cannot be used for designing systems that require human-agent teamwork. In addition, existing concepts that guided the design and implementation of artificial agents in this thesis will be discussed. These concepts provide a starting point for implementing artificial agents capable of leveraging SMM for decision-making.

This chapter is organised as follows. Section 2.1 details the fundamentals of the artificial agents; Section 2.2 discusses the existing models of human-agent systems; Section 2.3 reviews literature on SMM for human teams; Section 2.4 discusses the formal teamwork
models; and Section 2.5 concludes the chapter and highlights the similarities between formal teamwork models and SMM for human teams.

2.1 Artificial Agents and Mental Models

In order to realise SMM for artificial agents, each component of the artificial agent needs to be identified, modelled, and implemented. This section describes the fundamental components of the artificial agents implemented in this thesis. These components form the basic building blocks of the artificial agents that leverage SMM in their decision-making processes. As stated earlier, while some of these concepts are taken from theories of human reasoning, the following discussion is meant to be in the context of an artificial agent as far as this thesis is concerned.

Mental models are simplified representations used by individuals to explain and predict their surroundings (Johnson-Laird, 1983, 2010, Jones et al., 2011, Rouse and Morris, 1986). These models have content and structure or relationships between the content. Some examples of the content, which may be prior or gathered through perceptions (including messages exchanged with or by other agents), are beliefs about the state of the environment and other agents, the agent’s and/or others’ goals, intentions, plans, knowledge, and reasoning rules that help an agent make decisions and handle events such as perceptions and communication. A possible representation of mental models is symbolic, for example, by using predicates. Predicates are used to define domain related concepts and these are used by the agent to represent the beliefs and the decision making rules. Previous works (Johnson-Laird, 1983, 2010, Jones et al., 2011, Rouse and Morris, 1986) have accounted for and discussed the representation of the mental models. While the strengths and limitations of the representational approach taken in this thesis is known, the symbolic form used in this thesis is not only a potential common approach in prior literature, but also common in the artificial agents community.

2.1.1 Artificial Agent Characteristics

An artificial agent is a system that is situated in some environment, and has some capacity to sense and act in that environment. These agents have three key abilities: 1) event processing; 2) knowledge representation; and 3) reasoning or decision-making. They have the ability to handle events, such as perceiving the environment (percepts) and communication. The knowledge representation gives the agent the ability to maintain a model of the environment, including other agents, and reason with this knowledge to make decisions,
such as choosing an action.

Such agents have certain characteristics as well, such as, being autonomous, social and proactive. While there is not an agreed upon definition of autonomy, it basically means to have the ability of making decisions to achieve the assigned goals independently. Agents are also expected to be social, that is, cooperate, coordinate and collaborate with other agents. Finally, an agent is expected to try to achieve the assigned goals, that is demonstrate goal-directed behaviour.

A common model that is used as the basis of design of agents that embody the characteristics outlined above is known as the belief-desire-intention or BDI model (Bratman, 1987, Rao et al., 1995). The behaviour of such an agent can be explained in terms of the agent’s beliefs, desires, and intentions. This is commonly known as an intentional system and this approach was inspired by folk psychology. A simplified BDI architecture is shown in Figure 2.1.1. The mental state of a BDI agent contains beliefs, desires (goals), plans, and intentions. Inspiration is taken from the BDI architecture to design the kind of artificial agents required in this thesis. BDI agents are known as intentional agents because the agents maintain an intentional stance (Dennett, 1989) towards their environment. Intentional agent’s behaviour is explained and predicted by means of mental states such as
beliefs, desires, goals, intentions and commitments. Thus, agents that maintain mental states make it straightforward to design and implement SMM related constructs.

The behaviour of a BDI agent is generated by practical reasoning performed by an interpreter. The overall reasoning cycle, which involves both the deliberation and means-ends reasoning (explained later in Section 2.1.1), usually involves many steps. It first involves perceiving the environment, updating beliefs and goals accordingly. Then based on the current beliefs and goals, selecting applicable intentions (and plans) (and pushing them onto the intention stack), and finally, selecting an intention to execute. If the intention is another goal rather than an atomic action, the agent may find appropriate plans for these (and push these on to the intention stack). The practical reasoning process usually ends when the agent has no more goals.

The key components of this architecture are explained next. The objective of the following definitions and explanations is to make these concepts unambiguous. The basic representational unit that captures the agents mental state and knowledge is a predicate. Each predicate represents a single belief and a set of these beliefs forms the agent’s belief base. Agents also maintain a goal base, their plans, and so on. These are explained below.

**Beliefs**

An agent may maintain a belief base (\(\Psi\)), which is a set of predicates (\(\psi\)). A belief expresses a property of an entity. For example, in the case of the classical Blocks World domain, \(\text{on}(\text{b1}, \text{b2})\) could be used to represent that block \(\text{b1}\) is on top of block \(\text{b2}\). These beliefs represent what the agent considers to be true at the moment and can be dynamic, that is, the beliefs change based on what the agent perceives and the interaction that the agent has with other agents in the environment. The following provides definitions of these concepts.

**Definition 2.1.1 (Term).** A term, denoted generically as \(\tau\), is a variable \(x, y, z\) (with or without subscripts); a constant \(a, b, c\) (with or without subscripts); or a function \(f(\tau_0, \ldots, \tau_n)\) where \(f\) is a n-ary function symbol applied to (possibly nested) terms \(\tau_0, \ldots, \tau_n\).

**Definition 2.1.2 (Predicate).** A predicate (or a first-order atomic formula), denoted as \(\psi\), is any construct of the form \(p(\tau_0, \ldots, \tau_n)\), where \(p\) is an n-ary predicate symbol applied to terms \(\tau_0, \ldots, \tau_n\).

**Definition 2.1.3 (Belief and Belief Base).** A belief is the information believed to be true by an agent, that is, it is a ground predicate. A belief base, denoted as \(\Psi\), is a set of predicates, that is, \(\{\psi_0, \ldots, \psi_m\}\). Therefore, a belief base is all information that the agent believes to be true at any given point in time.
Desires, Goals and Intentions

Desires are possible states of the world that the agent may want to bring about. A goal, which is treated as a kind of desire, represents the state of the world an agent would like to bring about, and an agent maintains a goal base comprising of potential goals (\(\Phi\)). Intentions (\(\theta\)) form the subset of the goals that an agent is presently trying to achieve. Intentions have two parts: 1) the goal; and 2) the means to the goal - the plan. The agents explained in this thesis differentiate between the two. That is, intentions are a subset of goals, and an agent generates plan(s) to achieve the intentions and usually commits to achieving the intentions until the agent believes that the intention can no longer be achieved or the intention has been achieved. A plan (\(\alpha\)) consists of a sequence of actions that an agent takes to potentially go from the current state of the world to the goal state. The plans considered in this thesis are sequential plans. There are other alternatives such as policies (Ghallab et al. 2004), which are not considered in this thesis. The environment may be dynamic, so the agent cannot always ensure that the plan will in fact result in the goal state. Note that the definitions of the plan and action are in essence STRIPS (Fikes and Nilsson, 1971) definitions, and have been presented here for the coherency of the discussions.

Definition 2.1.4 (Goal and Goal Base). A goal is the state of the world that the agent wants to bring about. A goal base, denoted as \(\Phi\), is a set of predicates, that is, \(\{\phi_0, ..., \phi_g\}\). Therefore, a goal base is the set of world states that the agent may want to achieve.

Definition 2.1.5 (Action). An action is a tuple \(\langle name(a), pre(a), del(a), add(a)\rangle\). The name or identifier, \(name(a)\), of the action can be expressed as a predicate of the form \(p(\tau_0, ..., \tau_n)\), where \(\tau_0, ..., \tau_n\) are the parameters (variables). The precondition, \(pre(a)\), specifies predicates that must be true for the action to be applicable. The \(add(a)\) and \(del(a)\) are add and delete effects, which specifies the beliefs that are added and removed to/from the belief base following successful execution of action \(a\). Each action can optionally have a cost, \(c\), associated with it.

Definition 2.1.6 (Plan). A plan is a sequence of actions \(\alpha = [a_0, ..., a_{length(\alpha)}]\). The number of actions in the plan is \(length(\alpha)\). A plan is generated for a planning problem, which is a tuple \(\langle \Delta, I, \phi\rangle\), where \(\Delta\) is the domain specification, \(I\) is the initial state generated using the belief base, \(\Psi\), and \(\phi\) is the goal state. A set of agent plans is denoted using \(\Gamma\). When each action in the plan has a cost, the cost of a plan is the sum of the cost of each action.

Definition 2.1.7 (Intention). An intention, \(\theta\) is the subset of goals from the goal base, \(\Phi\), that the agent is currently pursuing. Each intention is a tuple \(\theta = \langle \phi, \alpha \rangle\), that is, a goal
and an optimal plan associated with it. The optimal plan here refers to a plan that has the
minimal cost in cases where a goal may have many possible plans. The tuple representation
means that an intention is a goal and a means to achieving that goal.

Artificial Agent Reasoning

There are two core reasoning processes of cognitive agents, deliberation and means-ends
reasoning. Together, these two processes are referred to as practical reasoning. Deliberation
takes as input the beliefs of the agent and the agent’s goals, and outputs the intentions.
Means-ends reasoning then takes these intentions as input and produces candidate plan(s)
that achieve the intentions as the output. Planning can either be done at design time
for the agent by the agent designer, completely by the agent at run-time or partially at
design-time and the agent can complete the plan at run-time, which is the idea behind
SharedPlans (Grosz and Kraus, 1999).

In the first approach, the designer generates a set of plan rules that the agent triggers
at run-time. Such is the case of the classical belief-desire-intention (BDI) agent (Bratman,
1987, Meneguzzi and De Silva, 2015). The agent relies on predefined recipes, that is, a set
of plan rules and their triggering conditions. At run-time, the agent matches the triggering
conditions and selects the plan rule that matches the triggering condition and executes it.
Alternatively, languages such as GOAL (Hindriks, 2014) and Jason (Bordini et al., 2007)
support declarative goals, which are achieved by a plan body. The plan body nonetheless
needs to be defined that can be done at run time. This approach is suited to situations in
which there are few well-defined behaviours and the environment conditions do not require
the agent to consider alternatives. However, this does require the designer to consider all
normal and exceptional scenarios at design time and provide the plan rules in a specific
order to realise the required agent behaviour.

At the other extreme is planning from first principles. Usually referred to as model-based
approach, this approach employs automated planning (Ghallab et al., 2004) concepts and
tools to generate the agent behaviours. A model of the agent and the current state of the
world as seen by the agent (the agent’s belief and goal bases) is supplied to a planner (usually
an off-the-shelf planner) to generate the plan bodies. The advantage of this approach is
that the agent designer does not necessarily have to detail every scenario that the agent
will face. However, the challenge involved with this approach is to design a model that is
sufficient to generate the required behaviours. It should be noted though, that this process
is easier, more manageable, and extensible than finding solutions to almost every scenario
the agent has to go through. Finally, a hybrid approach is also possible where a designer
can provide some or partial plans and an agent can either complete these plans at run-time or perform first-principles planning.

The approach taken in this thesis is a combination of these approaches. In the first empirical study presented in Chapter 5, the behaviours of the agents are pre-scripted and are sufficient to complete the assigned tasks. In the the study presented in Chapter 6, the behaviours of the agents are generated via first-principles planning. The hybrid approach has not been attempted.

**Communication Planning** It is desired that agents have social capacity, that is, are cooperative and collaborative agents that can work with others. This is an important skill as agents may find themselves in situations where they need assistance of other agents, are required to provide assistance to others, or work with others for efficiency reasons. One obvious way that agents can interact with others is via communication. Communication can be multi-modal (Turk, 2014) and these modalities can operate simultaneously, and agents need to reason about their communication actions like any other action.

Agent communication is typically based on speech-act theory (Austin, 1975, Searle, 1969). This theory treats communication as an action, which is intended to change the state of the world. When interacting with other agents, this action is intended to change the mental state of others. The view that communication is an action implies that the agents can perform communication planning. Austin (1975) considered three aspects of an utterance. An utterance has an illocutionary force, that is, the type of utterance, and perlocutionary force, which represents what the utterance is attempting to achieve. Each speech-act can be decomposed into performative and content. There are different types of performatives (Searle, 1969), such as representative (e.g. informing), directives (e.g. requesting), commisives (e.g. committing), expressives (e.g. expressing), and declarations.

While the discussion of various aspects of communications continue (Green, 2015), in this thesis, we assume that the agents are able to communicate using an agent communication language. The Knowledge Query and Manipulation Language (KQML) (Labrou and Finin, 1994, Mayfield et al., 1995) and The Foundation for Intelligent Physical Agents (FIPA) ACL are two such agent communication languages. KQML, which provides the basis of FIPA ACL, is a model based on three components: 1) propositions (content); 2) performatives (capturing agent’s intentions in sending a message); and 3) ontologies (meanings). FIPA ACL enhanced KQML elaborating on the performative set and semantics. The meanings of FIPA specified communicative acts have preconditions and effects, specifying when the message can be sent and the message’s intended purpose. This is similar to STRIPS-like
encode of planning actions, which also have preconditions and effects. Such an approach facilitates reasoning with mental states of agents (senders and receivers), that is, reasoning about the possible mental states of the agents before and after transmission.

2.2 Existing Models of Human Agent Interaction (HAI)

The design of systems in which humans and agents work together is not a new one [Chen and Barnes, 2014]. This section discusses the principles that guide the development of these systems. Initially, models for human agent interaction (HAI) were based around the notion of static allocation of functions to automation [Fitts, 1951]. The term automation can be used to cover a broad range of systems, for example robotic systems or software. For the purpose of the discussions that follow, the term agent is used to refer to these different forms of automation, and these two terms are used as synonyms for the following few paragraphs.

In the context of HAI, function allocation is the process of determining the allocation of tasks to the agents. The Fitts list [Fitts, 1951], also known as MABA-MABA (Men are better at, Machines are better at) list, is a criteria for performing function allocation. The Fitts list allocates functions based on the capabilities of the humans and automation. The functions best performed by automation should be automated, while others should be delegated to humans. In principle, allocating functions based on the strengths and weaknesses of the agents is acceptable. However, while the Fitts list is based on sound scientific theory and makes valuable contributions [de Winter and Dodou, 2011], the notion of static and crisp allocation of functions, that is all-or-none automation, does not capture scenarios in which the automation could provide different levels of support to the human for a given function. This means, characterising autonomy as full or none is not a useful approach. For this reason, a direct implementation of the Fitts List is not practical.

Later, the notion of levels of automation [Beer et al., 2014; Funk and Miller, 2008; Miller, 2012; Parasuraman et al., 2000; Sheridan and Verplank, 1978] guided the development of the HAI models. In a study looking at supervisory control of undersea vehicles involved in underwater search and object recovery tasks, Sheridan and Verplank (1978) realised that automation is not all-or-none but can instead be described on a continuum. This is contrary to the idea behind Fitts List. Accordingly, Sheridan and Verplank (1978) proposed a ten-scale model for levels of automation (LOA) describing how machines/automation and humans share the control of a system. At the lowest level, the human operator performs the tasks manually without any support from the automation and at the highest level, the automation performs the tasks autonomously. Going from the lowest to the highest level,
the intermediate levels describe how the automation slowly replaces the human operator. Following this initial model, a number of other models based on LOA were later proposed (Beer et al., 2014, Funk and Miller, 2008, Miller, 2012, Parasuraman et al., 2000).

An important limitation of the above models is that teamwork is not explicitly or adequately addressed. This is in part because the notion of LOA is based on substitution, that is, automation replacing the human partner and not human and automation collaborating. This means that the concept of LOA is insufficient and incomplete for designing complex human-machine teams who may be engaged in joint activity (Beer et al., 2014, Funk and Miller, 2008, Miller, 2012, Parasuraman et al., 2000).

To design systems capable of exhibiting human-agent collaboration in joint activities, Johnson et al. (2014) proposed the Coactive Design Method, which allows designers to identify, analyse, implement and evaluate interdependence relationships between agents. The term “coactive” is used to emphasise that humans and agents will work closely and require continuous interaction. According to Johnson et al. (2014), a model of autonomy that accounts for human-machine or machine-machine interdependence will be able to address the capabilities, strengths, and weaknesses of all parties. This captures the principle of Fitts list but allows for teamwork between the agent and the human counterpart. It will allow the parties to coexist in a cooperative task, exhibiting teamwork without requiring explicit assignment of levels of automation for different stages. Instead of thinking in terms of a scale of LOA, the analysis of interdependence provides insight into how the automation and the human will collaborate. Interdependence has been highlighted as the central organising principle of Coactive Design Method, and has been defined as “the set of complementary relationships that two or more parties rely on to manage...dependencies in joint activity” (Johnson et al., 2014). The Coactive Design Methods works by identifying how the parties are interdependent, and using this information to implement the required functionality in the automation and the operator’s interface to help both manage the interdependence relationship. Interdependence relationships must be complementary, and help identify what common ground (Clark, 1996, Miller et al., 2017) should comprise.

An important dimension that remains unexplored fully by the above works is that of the shared understanding or common ground between team members. The literature on human teams, which has focused on similar constructs, such as SMM, can certainly help in determining the important elements of the common ground. Issue in relation to common ground has been pervasive in the space of human-agent interaction. Parasuraman and Riley (1997) pointed out that operators over-reliance on automation leads to ineffective monitoring of automation, and can cause loss of situation awareness (SA). SA is “perception
of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the future” (Endsley, 1995b). In simple terms, this refers to how well the user knows what is going around him or her in relation to the user’s tasks. Studies by Johnson et al. (2012), Onnasch et al. (2014), Stubbs et al. (2007) and Li et al. (2014) reveal that the benefits of automation having more control over tasks than human counterpart are improved routine performance and reduced workload while the costs are loss of situation awareness and inability to effectively take back control of the system from the automation. A number of techniques have been proposed to mitigate some of these costs. For example, the control should be shared such that the human is kept in the loop (Parasuraman et al., 1996) because this improves the operator’s routine performance and there is minimal loss of situation awareness. It can also be helpful to represent the system dynamics in a salient manner such that the system states and any abnormalities are identified easily (Jamieson, 2007, Shattuck et al., 2008, Vicente and Rasmussen, 1990). These results indicate that an autonomous agent may become part of the problem itself. The designers of human-agent systems must therefore pay close attention to creating and maintaining common ground among other things, and the success of SMM in enabling effective teamwork in human teams can provide guidance on what are the important elements of the common ground for effective teamwork.

2.3 Team Cognition and SMM for Human Teams

The literature on SMM for human teams is vast. This section reviews the key concepts and results from recent meta-analytic studies (DeChurch and Mesmer-Magnus, 2010a, b, Mesmer-Magnus et al., 2017, Mohammed et al., 2010) and selected articles focusing solely on aspects relevant to this thesis. In the discussions, the term shared mental model is used as a synonym for related constructs, such as team mental model. There are at least two important aspects of SMM that informs the design of SMM for artificial agent or human-agent teams. These are: 1) the content; and 2) measurement of SMM. Also, of relevance are the key outcomes of SMM as these outcomes not only motivate the reason for research into SMM but also provide some benchmark effects that guide the decision on the success or failure of SMM for artificial agent teams.

Team effectiveness is largely a function of interaction processes and emergent states (Kozlowski and Ilgen, 2006, Marks et al., 2001). Team processes include team member interactions and emergent states is effectively team cognition (DeChurch and Mesmer-Magnus, 2010a, Kozlowski and Ilgen, 2006, Marks et al., 2001). Team behavioural processes, as
Marks et al. (2001) explain it, are:

“members’ interdependent acts that convert inputs to outcomes through cognitive, verbal, and behavioural activities directed toward organising task work to achieve collective goal”.

Examples of team process include planning, goal setting, coordinating, and team-backup (supportive) behaviour. Team performance is an objective or subjective judgement of how well a team meets valued objectives (DeChurch and Mesmer-Magnus, 2010a).

Team cognition refers to the representation and distribution of team relevant knowledge within teams. Existence of such team level knowledge enables members to anticipate and execute actions (Kozlowski and Ilgen, 2006). Team cognition impacts at least the team processes and team performance (Cannon-Bowers and Salas, 2004). There are at least two forms of team cognition: compositional and compilational. In compositional form, individual-level states are similar in form and function to team-level ones, and in compilational form, team-level states are generally different in form to the individual-level counterpart) (Kozlowski and Klein, 2000). The shared mental models (SMM) literature is largely representative of compositional form. In compositional emergence, sharedness is central and it refers to the degree to which members’ mental models are similar. However, identical mental models are not a requirement. Various other terms have been used to capture sharedness, including similarity, agreement, convergence, compatibility, commonality, consensus, consistency, and overlap. SMM refer to a common understanding by the team members regarding task, team, and temporal aspects of their work (Klimoski and Mohammed, 1994, Mohammed et al., 2010, 2015). SMM fulfill multiple functions, such as allowing team members to interpret information in a similar manner (description), share expectations concerning future events (prediction), and develop similar causal accounts for a situation (explanation) (Cannon-Bowers et al., 1993).

A wide variety of studies indicate that when team members have similar mental models, they are likely to achieve high performance levels because they are able to accomplish the tasks efficiently without the need for explicit coordination and communication (DeChurch and Mesmer-Magnus, 2010a). Similar mental models also enable members to interpret changes in the task environment in a compatible way (Cannon-Bowers et al., 1993). This enables them to anticipate the needs and actions of other members while dealing with these changes (DeChurch and Mesmer-Magnus, 2010a). Further, when teams have similar mental models, they are on the same page regarding the functioning of the team, the team’s strategy, and when deadlines have to be met (Mohammed et al., 2010, Mohammed and Nadkarni).
This is likely to facilitate the synchronisation of team activities among the team members, enable efficient communication, and ensure that the team activities are aligned with collectively agreed upon goals. In contrast, when team members do not have similar mental models, team activities may lead to misunderstandings, process loss, and frustration among the team members.

**Figure 2.3.1:** SMM content proposed by Cannon-Bowers et al. (1993).

### 2.3.1 SMM Content

Cannon-Bowers et al. (1993) proposed four nonindependent SMM content domains (Figure 2.3.1): an equipment model (knowledge about tools and technology), a task model (understanding of work procedures, strategies, and contingency plans), a team interaction model (awareness of member responsibilities, role interdependencies, and communication patterns), and a team model (understanding of teammates’ preferences, skills, and habits). However, in practice, researchers have collapsed these into two categories, which are taskwork model and teamwork model. The taskwork model captures work goals and performance requirements, while the teamwork model includes the interpersonal interaction requirements and skills of other team members.

When members share a task mental model they have a similar understanding about how the task should be accomplished in terms of procedures and practises, as well as about the
resources needed to accomplish the tasks. Team members that have a similar task mental model are more likely to communicate information in a way that is understood by the recipients. When members have a shared team mental model, they have a similar understanding about the team interaction, their responsibilities, the relation between their roles, and the knowledge, skills, and abilities of each team member. Shared understanding among the team members on how to interact with each other is likely to facilitate a variety of team processes, including communication and coordination (Mathieu et al., 2000). Mohammed et al. (2015) defined a temporal SMM as agreement among group members concerning deadlines for task completion, the pacing or speed of activities, and the sequencing of tasks, which helps the members prioritise their time.

SMM represent different types of knowledge structures. SMM knowledge may be declarative (knowledge of what), procedural (knowledge of how), and strategic (knowledge of the context and application) (Rouse et al., 1992). Whereas knowledge structures refer to descriptive states of nature believed to be true, SMM can also have belief structures, which refer to desired states of nature that are preferred or expected. There are two important properties of SMM: sharedness and accuracy. Accurate SMM mirror the “true state of the world”. Highly convergent and accurate mental models are expected to yield the greatest team performance benefits (Mohammed et al., 2010).

2.3.2 SMM Measurement

When measuring SMM, both content and structure are important (DeChurch and Mesmer-Magnus, 2010b). Content refers to the knowledge that comprises cognition, whereas structure represents how concepts are organised in the minds of members. Elicitation techniques derive mental model content, and representation techniques reveal the structure or the pattern of relationships between elements (Mohammed et al., 2010). Theoretical work on the measurement of mental models generally outlines three important characteristics that permit measurement of the degree of convergence or similarity among team members’ organized knowledge representations: (a) elicitation method, (b) structure representation, and (c) representation of emergence. The elicitation method refers to the technique used to determine the content or components of the model. Commonly used elicitation techniques for human teams are similarity ratings, concept maps, rating scales, and card sorting tasks (Mohammed et al., 2010). Structure representation refers to the organised knowledge structures, that is, the degree of correspondence between how the knowledge contained in the model is represented in the mind and how the knowledge representation is modelled by the researcher. Similarity ratings and concept maps are common methods. Finally, repre-
sentation of emergence is concerned with how the individual-level contents and structure are collectively considered at the team-level.

2.3.3 Outcomes of SMM

Decades of research established a strong positive relationship between sharedness and team performance across a variety of task and team types and regardless of measurement type (DeChurch and Mesmer-Magnus, 2010a,b, Mohammed et al., 2010, 2013). Both taskwork and teamwork mental model sharedness have been found to predict performance while similarity of taskwork mental models had stronger direct effects on team performance as compared with teamwork models but converse has been reported as well. SMM allow members to predict information and resource requirements. In addition to team performance, SMM convergence has also been positively associated with various team processes, including back-up (supportive) behaviour quantity and quality, coordination, and communication. For example, Mathieu et al. (2000) showed that team processes mediate the relationship between SMM sharedness and team performance. Thus, interpersonal interactions are more effective when team members share mental models, enabling them to perform better.

2.3.4 Team Interdependence

Interdependence is a defining characteristic of teams, referring to the “extent to which team members cooperate and work interactively to complete task” (Stewart and Barrick, 2000). Research indicates that the cognition-process relationship is stronger for compositional emergence under conditions of high team interdependence while the cognition–performance relationship is stronger for compositional emergence under conditions of moderate, rather than high, interdependence (DeChurch and Mesmer-Magnus, 2010a,b, Mohammed et al., 2010, 2015).

When considering compositional emergence (e.g., shared mental models), cognition was more predictive of team behavioural process for highly interdependent than for moderately interdependent teams, which is consistent with the expectation that as the interdependence of the task increases, overlap in members’ understanding of important aspects of the task and team will enable smoother synchronisation of joint actions, and permit members to better anticipate one another’s needs (Marks et al., 2001). However, in support of Kozlowski and Ilgen’s (Kozlowski and Ilgen, 2006) prediction, compositional cognition was more predictive of team performance for moderately interdependent teams than for highly interdependent teams.
2.4 Formal Teamwork Models

The discussion now shifts from the SMM research conducted using human teams to research taking a formal perspective on teamwork. Research on formal models of teamwork has been conducted for decades and is still on-going (Bradshaw et al., 2011, Bratman, 1992, Castelfranchi, 1995, Dunin-Keplicz and Verbrugge, 2011, Gilbert, 1989, 2008, Grant et al., 2005, Grosz and Kraus, 1999, Jennings, 1993, Johnson et al., 2014, Kinny et al., 1994, Levesque et al., 1990, Rao et al., 1992, Searle, 1990, Singh, 1998, Sycara and Sukthankar, 2006, Tuomela and Miller, 1988, Wooldridge and Jennings, 1999). This section reviews the formal models of teamwork with the aim of providing not only a more formal account of teamwork but also to link these constructs to the literature on SMM. Such links will help formulate a more formal understanding of the core concepts of SMM that has been discussed informally in literature looking at SMM for human teams.

Joint activity can be investigated from two perspectives: internal or external (Searle, 1990, Singh, 1998, Wooldridge and Jennings, 1999). In the external view, the actions performed by the agents are studied in order to determine when and how well the agents are cooperating while in the internal view, the agents internal state is used as the basis for evaluation. When viewing agents from a purely external perspective, it is impossible to determine whether they have coordinated their actions. Therefore, coordination is best studied by examining the mental state of the individual agents. The question then is, what aspects to consider? Numerous formal models have been proposed to answer this question. Some of the most influential ones are reviewed in the proceeding sections.

In explaining how joint action is possible, researchers have emphasised the role of intentions, beliefs and other propositional attitudes (see for example (Bratman, 1992, Tuomela and Miller, 1988)), as well as commitments (see for example (Castelfranchi, 1995, Gilbert, 2008, Jennings, 1993)). Underlying most of the theories of teamwork is the notion of collective intentionality. Collective/shared/joint intentionality refers to collaborative interactions in which participants share psychological states with one another.

There has been some debate as to whether these collective attitudes can be reduced to summaries of individual attitudes or not (Gilbert, 1989, Rao et al., 1992, Searle, 1990, Tuomela and Miller, 1988); for example, whether the belief of a group is merely the combination of the belief of its members, or whether individuals can have beliefs that conflict with the beliefs of groups in which they are a member. There is a general agreement that joint actions involve joint intentions and that these do not reduce to a mere summation of individual intentions, even supplemented by mutual beliefs or mutual knowledge (Gilbert.
1989, Searle, 1990, Tuomela and Miller, 1988). These views have led researchers to propose a number of formal models of teamwork, which are discussed next.

A number of formal models of teamwork have been proposed (Dunin-Keplicz and Verbrugge, 2011, Grant et al., 2005, Grosz and Kraus, 1999, Levesque et al., 1990, Rao et al., 1992, Wooldridge and Jennings, 1999). The following sections review the five most successful and influential models, and following that aim to link the elements of each of these models to the fundamental components of the shared mental model proposed for human teams. Each of the five models is used to highlight the important elements that form part of mental state of artificial agents. These elements are, therefore, important to consider when considering SMM for artificial agents.

2.4.1 Joint Intentions

The most comprehensive attempt to formalise individual intentions is due to Levesque et al. (1990). Their model has two levels of detail: a fundamental one which provides the primitives for a theory of action (e.g. definitions of beliefs, goals and action) and a second layer which builds upon these concepts to develop a theory of rational action. At the second layer they capture the notion of commitment by defining a persistent goal. Agent A has a persistent goal to achieve objective G, relative to its motivation M, if and only if the following conditions prevail: (i) A believes that G is currently false; (ii) A wants G to be eventually true; (iii) this state of affairs will continue until A comes to believe either that G is true or that it will never be true or that M is false. An intention is then defined as a commitment to act in a certain mental state: agent A intends to do action G if it has the persistent goal to have done that action and, moreover, to have done it believing throughout that it was doing it.

They use constructs of dynamic logic to describe sequences of actions and modal operators to express time associated propositions. In their logic, $\alpha_1;\alpha_2$ is sequential action composition and $\alpha_1||\alpha_2$ is the concurrent occurrence of $\alpha_1$ and $\alpha_2$. $p?$ is a test action: if $p$ is true, the action succeeds, but if $p$ is false it fails. In addition to dynamic logic constructs, they use temporal modal logic operators and modal operators for mental states given in Table 2.4.1.
UNTIL\((p, q)\) specifies that until \(p\) is true \(q\) will remain true.

\(\square p\) indicates that \(p\) is true from now on.

\(\lozenge p\) indicates that \(p\) is true at some point in the future.

DONE\((x, y, \alpha)\) indicates that \(\alpha\) has just happened and \(x\) and \(y\) are its agents.

DOING\((x, y, \alpha)\) indicates that \(x\) and \(y\) are doing \(\alpha\).

GOAL\((x, p)\) says that \(x\) has \(p\) as a goal.

BEL\((x, p)\) says that \(x\) has \(p\) as a belief.

MB\((x, y, p)\) says that \(p\) is mutually believed by \(x\) and \(y\).

<table>
<thead>
<tr>
<th>Table 2.4.1: Temporal modal logic operators and modal operators for mental states.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joint Persistent Goal</td>
</tr>
</tbody>
</table>

JPG, joint persistent goal with respect to \(q\) would be expressed as follows:

\[
JPG(A_1, A_2, DONE(A_1, A_2, \alpha), q) =
MB(A_1, A_2, \neg DONE(A_1, A_2, \alpha)) \land MG(A_1, A_2, DONE(A_1, A_2, \alpha)) \land
UNTIL([\neg MB(A_1, A_2, DONE(A_1, A_2, \alpha)) \lor MB(A_1, A_2, \square \neg DONE(A_1, A_2, \alpha))
\land MB(A_1, A_2, \neg q)],
WMG(A_1, A_2, DONE(A_1, A_2, \alpha)))
\]

The two clauses indicate that \(A_1\) and \(A_2\) mutually believe that \(\alpha\) has not been done yet and that they both have the mutual goal for \(\alpha\) to be done eventually. The clause that starts with UNTIL specifies when the agents will stop having the persistent goal and what goals and beliefs \(A_1\) and \(A_2\) will have until then. In particular, the agents will stop having the JPG when either they mutually believe that has been done, or they will mutually believe that will never be done or they will mutually believe that the reason for doing, namely \(q\), is not true any more. Until then, they will have a weak mutual goal (WMG) (refer to Levesque et al. (1990) for formal definition), which means that they mutually believe that both of them have a weak goal with respect to the other agent. \(A_1\) having a weak goal with respect to \(A_2\) that \(\alpha\) will be done means that one of the following is true: (i) \(A_1\) does not believe that the \(\alpha\) has been done and has a goal that it will be eventually done. (ii) \(A_1\) believes that \(\alpha\) has been done and has a goal that this belief will become a mutual belief with \(A_2\). (iii) \(A_1\) believes that \(\alpha\) will never be done and has a goal that this belief will become jointly believed with \(A_2\).
Joint Intentions

$A_1$ and $A_2$ having joint intentions to do $\alpha$ means that they have a joint persistent goal to do $\alpha$ while mutually believing (throughout the execution of ) that they are doing it.

$$JI(A_1, A_2, \alpha, q) =$$

$$JPG(A_1, A_2, \)$$

$$DONE(A_1, A_2, \)$$

$$UNTIL(DONE(A_1, A_2, \alpha));$$

$$MB(A_1, A_2, DOING(A_1, A_2, \alpha))\); \alpha, q)$$

2.4.2 Team Plans

Kinny et al. (1994) and Rao et al. (1992) focus on groups performing complex plans to achieve a goal. They discuss how to find a suitable team for a task; how to synchronise the establishment of joint goals and the adoption of intentions; how to assign roles; and how to maintain proper temporal relations while executing different parts of the plan by different members of the team. They reduce all the multiagent attitudes to single agent attitudes and provide mechanisms for expressing which part of a joint plan is the responsibility of each member of the team. A plan includes team variables that represent the roles of the plan. There may be a library of plans that are available with non-assigned roles. Once a team is formed, the roles are assigned to specific individual agents or specific teams. The authors describe two protocols the agents can use to form the relevant beliefs and intentions (commit-and-cancel and agree-and-execute). If the agents agree on a plan and the role assignments, they form a joint intention.

2.4.3 SharedPlans

In SharedPlans formalism (Grosz and Kraus, 1999), agents are said to have plans when they have a particular set of intentions and beliefs. Both shared plans and individual plans are considered. Shared plans are constructed by groups of collaborating agents and include subsidiary shared plans formed by subgroups as well as subsidiary individual plans formed by individual participants in the group activity. The formalisation distinguishes between complete plans — those in which all the requisite beliefs and intentions have been established — and partial plans. The plan definitions apply intention operators as follows:

- $\text{Int.To}(G, \alpha, T_i, T_\alpha, con(\alpha), IC_\alpha)$ represents agent $G$’s intention at time $T_i$ to do action
α at time $T_\alpha$ under the constraints $\text{con}(\alpha)$ in the context $IC_\alpha$.

- Int-Th($G, prop, T_i, T_{prop}, IC_{prop}$) represents an agent’s ($G$) intention at time $T_i$ that a certain proposition $prop$ hold in the intentional context $IC_{prop}$ at time $T_{prop}$.

An Int.To commits an agent to means-ends reasoning while Int.Th forms the basis for meshing sub-plans, helping one’s collaborators, and coordinating status updates, all of which play an important role in collaborative plans; any of these activities may lead to the adoption of an Int.To and thus indirectly to means-ends reasoning. The Int.To and Int.Th attitudes are important also for communication planning. When communication is viewed as an action, an agent can form an Int.Th for a number of reasons, such as to have the intention that members of the team know a piece of information. The agent can then form Int.To communicate that piece of information.

2.4.4 Cooperative Problem Solving

Wooldridge and Jennings (1999) propose a model of cooperative problem solving that describes the process from its beginning, with some agent recognising the potential for cooperation with respect to one of its goals, through to team action. The model consists of four stages:

1. recognition: in which an agent identifies the potential for cooperation;
2. team formation: in which the agent solicits assistance;
3. plan formation: in which the newly formed collective attempts to construct an agreed joint plan; and finally,
4. execution: in which members of the collective play out the roles they have negotiated.

Mutual Mental States

One of most important derived modal operators relate to that of mutual mental states. Mutual belief is defined via an ‘everyone believes’ (E-Bel) operator, which states that every agent in group $g$ has the belief that $\varphi$. This belief can be nested, as indicated below.

- Everyone believes: $(\text{E-Bel } g \varphi 0) \overset{\text{def}}{=} \varphi$
- Everyone believes (nested): $(\text{E-Bel } g \varphi u + 1) \overset{\text{def}}{=} \forall i.(i \in g) \Rightarrow (\text{Bel } i \ (\text{E-Bel } g \varphi u))$
- Mutual Belief: $(\text{M-Bel } g \varphi) \overset{\text{def}}{=} (\text{E-Bel } g \varphi u)$ for all $u \in N$
COMMITMENTS, CONVENTIONS, AND INTentions

A commitment is a pledge or a promise; a convention is a means of monitoring a commitment. The most important property of commitments is that commitments persist: having adopted a commitment, we do not expect an agent to drop it until, for some reason, it becomes redundant. The conditions under which a commitment can become redundant are specified in the associated convention.

Joint commitments have a number of parameters. First, a joint commitment is held by a group \( g \) of agents. Second, joint commitments are held with respect to some goal \( \varphi \), which is the state of affairs that the group is committed to bringing about. Third, joint commitments are held relative to a motivation \( \psi \), which characterises the justification for the commitment. They also have a pre-condition \( \chi \), which describes what must initially be true of the world in order for the commitment to be held. Formally,

\[
J\text{-Commit}(g \varphi \psi \chi c)
\]

Joint commitments are used to define joint intentions, which are held by a group \( g \) with respect to an action \( \alpha \) and motivation \( \psi \). Here \( \diamond \Phi \) means that \( \Phi \) is eventually satisfied; \( \chi_{soc} \) and \( c_{soc} \) involve social pre-conditions and conventions. We can read joint intention to mean that the group \( g \) has a joint commitment that eventually \( g \) will believe that \( \alpha \) will happen next, and then \( \alpha \) happens next.

\[
(J\text{-Intend} g \alpha \psi) \overset{\text{def}}{=} (M\text{-Bel}(g \text{\ Agts} \alpha g)) \land \\
(J\text{-Commit} g \alpha \Diamond (M\text{-Bel}(g \text{\ Does} \alpha))); \psi \chi_{soc} c_{soc})
\]

The detailed formalisms of each of the four stages is not discussed in this chapter.

2.4.5 COLLECTIVE INTENTIONS (TEAMLOG)

TeamLog [Dunin-Keplicz and Verbrugge 2011, Dunin-Keplicz et al. 2010] is a theory of collective motivational attitudes that enables teamwork in BDI systems. When building a logical model of teamwork, agents’ awareness about the situation is essential. This notion is understood as the state of an agent’s beliefs about itself, about other agents and about the environment. Together they constitute three levels of agents’ awareness: intra-personal (about the agent itself), inter-personal (about other agents as individuals) and group awareness. For teamwork to succeed, its participants need to establish a common view on the environment. This can be built by observation of both the environment and other agents.
operating in it, by communication, and by reasoning. TeamLog defines various concepts related to beliefs, knowledge, intentions, and commitments. Here the focus is primarily on commitments.

Collective Intentions and Collective Commitment

Collective intention, as a specific joint mental attitude, is the central topic addressed in teamwork. In TeamLog, teams are created on the basis of collective intentions: a team is constituted as soon as a collective intention among the members is present and stays together as long as the collective intention persists.

After a group is constituted via collective intention, a collective commitment between the team members needs to be established. While a collective intention is an inspiration for team activity, the plan-based collective commitment expresses the case-specific details provided by planning and action allocation. It is reflected in bilateral commitments towards individual actions. A bilateral commitment from agent $i$ towards agent $j$ to perform action $a$ is represented as $\text{COMM}(i, j, a)$. A group $G$ has a collective commitment to achieve goal $\varphi$ based on social plan $P$ $C\text{-COMM}_{G,P}(a, \varphi)$ iff all of the following hold: The group mutually intends $\varphi$ ($M\text{-INT}_G(G; \varphi)$); a successful execution of social plan $P$ leads to $\varphi$ ($\text{cons}(\varphi, P)$); and finally, for every action $a$ from plan $P$, there is one agent in the group who is bilaterally committed to another agent in the group to fulfil that action ($\text{COMM}(i, j, a)$).

$$\begin{align*}
C\text{-COMM}_{G,P}(a, \varphi) &\leftrightarrow M\text{-INT}_G(G, \varphi) \land \{\text{awareness}_G(M\text{-INT}_G(G, \varphi))\} \land \\
& \land \text{cons}(\varphi, P) \land \{\text{awareness}_G(\text{cons}(\varphi, P))\} \land \\
& \land \bigwedge_{a \in P} \bigvee_{i, j \in G} \text{COMM}(i, j, a) \land \{\text{awareness}_G(\bigwedge_{a \in P} \bigvee_{i, j \in G} \text{COMM}(i, j, a))\}
\end{align*}$$

Strong, Weak, Team, Distributed Collective Commitments

In the case of strong collective commitment, the team knows the overall goal, collectively believes that the social plan is correct and that things are under control. However, team members do not need to know exactly who is responsible for each task. In weak collective commitment, the team knows the overall goal, but does not know the details of the plan: there is no collective awareness of the plan’s correctness, even though actions have been appropriately allocated. In team commitment, the team is unaware of their existing collective commitments. Therefore, agents cannot infer if any action they take will result in completion of plan $P$. Finally, in distributed commitment, agents
may not even know the overall goal and there is no collective intention. No real team is created and agents work as rather loosely coupled group of agents in a distributed manner.

To summarise the five models: the focus of Joint Intentions is the persistence of joint intentions. The focus of Team Plans is on the team formation, complexity of actions, roles of agents (role assignment), and establishment of joint goals. SharedPlans emphasises individual intentions and actions needed for teamwork, plan and sub-plan execution and coordination in explicit time structure. Cooperative Problem Solving emphasises joint commitment to a goal and cooperation and mental states. Collective Intentions emphasises the formalisation of collective intentions in a multi-modal logical framework, mental attitudes in cooperation, formal modelling of group consistency, and failure recovery.

2.4.6 Other Approaches to Teamwork

Distributed Constraint Optimisation Problems (DCOPs)

DCOPs (Leite et al., 2014) have emerged as one of the main coordination techniques in multiagent systems due to their ability to optimise the global objective function of the problem described as the aggregation of distributed constraint cost functions. The main motivation to employ this formalism as a decentralised coordination model in multiagent systems is that DCOP algorithms are distributed, scalable, and robust. By definition, DCOP consists of a set of constraints and variables distributed among agents, where each variable has a finite and discrete domain. Each value in the domain represents a possible state of the agent. Constraints are denoted by a cost or reward relation between a pair of variable assignments. Therefore, DCOP aims to find a sequence of actions that minimises the global cost of the problem modelled as a set of constraints.

Decentralised control of partially observable Markov decision processes (DEC-POMDPs)

Markov decision processes (MDPs) are often used to model sequential decision problems involving uncertainty under the assumption of centralised control. The decentralised partially observable Markov decision process (DEC-POMDP) (Amato et al., 2013) is a rich framework to formulate sequential decision making and control problems for a distributed group of agents collaborating to achieve a common goal under uncertainty. As it is often the case that communication has some cost, latency or unreliability, centralisation may not be possible or may result in a poor solution. In contrast, solutions to Dec-POMDPs
yield decentralised control policies that the agents execute to collaboratively optimise the common objective. However, while many more specialised multiagent models have been widely studied, the more general problem of scaling up Dec-POMDP solution methods with an increasing number of agents is still an open research question. There has been a large amount of work in recent years on utilising problem structure to increase scalability in optimal and approximate solution methods as well as more scalable sub-classes that relax problem assumptions, which show progress.

2.4.7 Coordination Architectures

A number of coordination architectures based on one or more of the formal teamwork models or based on the principles of these formal models have been proposed. Generalised partial global planning (GPGP) and TAEMS (Durfee and Lesser, 1991; Lesser et al., 2004) provides a framework for coordinating multiple AI systems that are cooperating in a distributed sensor network. Members have an initial plan, which they then exchange with others and collectively tune the partial global plans. In GPGP, the joint plan is modified through a whiteboard mechanism by all involved robots. When they all agree on the result, each robot sends it to its own scheduler which starts its execution. Similarly, in RESTINA (Giampapa and Sycara, 2002), which is based on SharedPlans, agents begin with their version of Partial Shared Plans. Agents map their capabilities to the task requirements taking into consideration social parameters, such as authority, and generate a candidate list of roles. The agents then have to reach consensus on who will play which role in the team plan. Once consensus has been reached, the agents commit to executing the team plan.

Tambe (Tambe, 1997b) present a model of teamwork called STEAM (Shell for TEAMwork), based on enhancements to the Soar (state, operator and result) (Rosenbloom et al., 1993) architecture. More discussion on STEAM will appear in Chapter 6. Machinetta (Schurr et al., 2005), a proxy-based hybrid approach to enabling teamwork among diverse entities, is derived from the earlier STEAM and TEAMCORE coordination architectures. Machinetta defines a plan-based interaction mechanism and an architecture for the interaction in which the proxies are equipped with a model of the commitments and responsibilities necessary for teamwork. In other works similar to Machinetta, for example Tambe et al. (2005), researchers emphasise the role of hybrid representations for scalability, expressiveness and robustness in agent teams. Tambe et al. (2005) illustrated the synergistic interactions between distributed POMDPs, DCOPs, BDI systems and game theoretic representations.
2.5 Conclusion

A number of formal models of teamwork concepts overlap with key components of shared mental models. For example, to setup joint intentions, the team members need to know the current team tasks, potential team or member plans and variations, and so on. These are core components of the taskwork model. In addition, joint intentions and commitments require the team to be aware of the required roles, role assignments, capabilities of the team members, and so on. These are all core components of the teamwork model. In addition, setting up joint intentions and commitments and creating an awareness of these commitments require information flow via existing communication channels and protocols. These are generally assumed to be commonly known by all members and also an integral part of teamwork model.

As described in Dunin-Keplicz and Verbrugge (2011), members can establish a common view of the environment by observation of both the environment and other agents operating in it, by communication, and/or by reasoning. The formal models discussed in this chapter have focused on the outcome of the teamwork processes and have assumed shared understanding of various aspects, which have been made explicit in the SMM related research. Literature on formal models of teamwork point out many aspects or terminology common to shared mental model research, such as agents, teams, beliefs, goals, plans, common belief, common knowledge, joint or collective intention, collective commitment, group awareness, social convention, mutual mental states, shared plans, and joint action.

There is clearly an overlap between the formal models of teamwork and the key concepts of shared mental models. In fact, most of the concepts discussed under the umbrella of shared mental models are also discussed in one or more of the formal models of teamwork. However, the SMM literature provides a comprehensive account of what the common view or common ground comprises. In fact, SMM literature points out what components of SMM the human teams have found to be useful. Using these two streams of research, Chapter 4 provides a computational model of SMM.

Before moving to Chapter 4, Chapter 3 reviews another important concept relevant to teams: interdependence. As has been already established by other researchers and will be pronounced in Chapter 3, the interdependence between agents in combination with shared goals binds the team together. In addition, a recent method of designing systems for human-robot teamwork highlighted interdependence as the its central organising principle (Johnson et al., 2014). Therefore, a more detailed review of interdependence is undertaken in Chapter 3, and examples are used to illustrate the relevance of such an analysis to SMM.
Interdependence in Multiagent Systems

Teamwork is a form of collaboration in which members work towards achieving shared goals and there is also usually *interdependence* of actions (often involving different roles) and resources [Bradshaw et al., 2014]. Interdependence is a multidisciplinary construct. A taxonomy of interdependence has stemmed out of organisation psychology literature [Courtright et al., 2013; Johnson and Johnson, 1989; Thompson, 1967; Wageman, 1995, 2001]. In the context of multiagent systems, interdependence has been discussed using various theories, for example, social dependency theory [Castelfranchi, 1995, 1998; Castelfranchi et al., 1992; Lau et al., 2015; Sichman, 1998], and game theory [Grossi and Turrini, 2012].

Not only has interdependence been discussed in the context of understanding teamwork, a recent human-agent system design method has been inspired by it. To tackle the intricacies of designing agents capable of exhibiting collaboration with humans, Johnson et al. (2014) propose the *Coactive Design Method*. This method allows designers to identify, analyse, implement and evaluate *interdependence relationships* between agents involved in joint activities. *Interdependence* has been highlighted as the central organising principle of Coactive Design Method, and they defined interdependence as:

> “the set of complementary relationships that two or more parties rely on to manage...dependencies in joint activity”

This definition of interdependence builds on insights from earlier studies of human systems [Thibaut and Kelley, 1959; Thompson, 1967; Van De Ven et al., 1976] that interdependence is more complex than simply mutual dependence: it is about relationships and


31
includes consideration of the purpose of those relationships, which is to manage dependencies in joint activity. Interdependence relationships must be complementary, and help identify what common ground (Clark, 1996; Miller et al., 2017) should comprise.

As has been discussed in Section 2.3.4 of Chapter 2, team interdependence has been linked with SMM research. In most SMM research conducted using human teams, team member interdependence has been seen as a function of team effort or interdependence due to the tasks that team members perform (DeChurch and Mesmer-Magnus, 2010a,b). However, there are various forms of interdependence relationships, and discussions on interdependence stems from a number of disciplines. Therefore, there is a need to understand interdependence and its different forms in more detail. To support the need for such an investigation, and building on the claim by Johnson et al. (2014) that analysis of interdependence helps identify important elements of common ground, Chapter 5 presents a study undertaken to identify the type of SMM content that helps improve team performance as the level of agent interdependence increases. In this chapter, however, the discussion focuses solely on the different perspectives taken on interdependence and its different forms.

In addition, while the Coactive Design Method explicitly considers interdependence as an organising principle, it does not provide a formal framework in which to reason about interdependence types. The research only considers task interdependence, and does not consider other types, such as goal and agent interdependence. Therefore, two additional fruitful contributions can be made in this area: (1) the extension of the concept of interdependence beyond tasks; and (2) a more rigorous analysis of interdependence. According to sociologists and organisational theorists (Courtright et al., 2015; Johnson and Johnson, 1989; Puranam et al., 2012; Saavedra et al., 1993; Thibaut and Kelley, 1959; Thompson, 1967; Van De Ven et al., 1976; Wageman, 1995, 2004) numerous types of interdependence exist, including interdependence between agents, tasks, and goals.

Inspired by the above research, this chapter semi-formally defines the concepts of task, agent, and goal interdependence. The aim is to provide clear, formal definitions that could be used as a lens through which to view problems and consider design alternatives. Specifically, the objectives of this chapter are:

- Review the term interdependence from the perspectives of organisation psychology and distributed artificial intelligence community working on multiagent systems.
- Informally, define the different forms of interdependence.
- Provide semi-formal definitions of task, agent, and goal interdependence.
Demonstrate (using an example) the benefit of identifying the different forms of interdependence and relevance of such fine-grained analysis of interdependence to SMM.

This chapter is organised as follows: Section 3.1 discusses the existing perspectives on interdependence and the different forms of interdependence; Section 3.2 discusses a domain used in this and other chapters in this thesis for empirical studies and as an example to illustrate concepts; Sections 3.3, 3.4, and 3.5 present the definitions of task, agent, and goal interdependence respectively; Section 3.6 illustrates the benefit of the fine-grained analysis of interdependence and its relevance to SMM; Section 3.7 links the definitions presented in this chapter with existing work; and Section 3.8 concludes the chapter.

3.1 RELATED WORKS

The concepts of dependence and interdependence have been studied for decades. The main idea of this section is review some of the common accounts of these concepts and present a view that is most relevant to the kind of work done in this thesis. It is hoped that a side effect of this endeavour may later lead to a unified view of the construct that will be useful in designing artificial agents.

3.1.1 INTERDEPENDENCE - AN ORGANISATION PSYCHOLOGY PERSPECTIVE

Most of the work discussed in this section originates from organisation psychology. According to the groups and teams literature, there are various types of interdependence forms, and these have been grouped into two forms, structural and behavioural (Courtright et al., 2015, Wageman, 2001), as shown in Figure 3.1.1.

Behavioural interdependence refers to the interdependence of the members due to the interaction between them when completing the team tasks (Wageman, 2001). That is, these are teamwork processes or activities (DeChurch and Mesmer-Magnus, 2010b, Marks et al., 2001). Team processes are categorised as either transition, action, or interpersonal processes. Transition and action are taskwork focused and refer to activities and interactions aimed at planning and coordinating. Interpersonal processes, on the other hand, refer to member activities that help the team manage their interpersonal relationships.

Structural interdependence refers to the interdependence between agents because of task, goal, and reward systems of the team (Wageman, 2001). According to Courtright et al. (2015), structural interdependence is a distal predictor of team performance through proximal effects of teamwork behaviours. Therefore, structural interdependence can be seen as the reason for behavioural interdependence (Courtright et al., 2015). Structural
interdependence has been further classified as either task interdependence (Saavedra et al., 1993, Thompson, 1967, Wageman, 2001), which captures the work processes and resource distribution, or outcome interdependence (Saavedra et al., 1993, Wageman, 2001). Outcome interdependence itself is of two types: (1) goal interdependence, which is a measurement of collective output; and (2) reward/feedback interdependence, which refers to outcomes that accrue to the group as a whole. The key property of outcome interdependence is that the reward is generally taken to be a team-level measure, and not individual-level.

Saavedra et al. (1993), based on work by Thompson (1967), proposed four types of task interdependence: pooled, sequential, reciprocal, and team. The order of these tasks represents increasing levels of dependence between agents. The more the task work requires the members to be dependent on each other, the more the chances of interaction, which in the long term can result in better information sharing, coordination and joint decision making (Courtright et al., 2015). In addition, Saavedra et al. (1993) discuss goal interdependence. A detailed discussion of these two types together with agent interdependence will be presented later in this chapter as these informed the study presented in Chapter 5.

Yet, another view from this discipline is that of epistemic interdependence (Puranam, 34)
Epistemic interdependence occurs when one agent’s optimal choices depend upon a prediction of another agent’s actions. Puranam et al. (2012) claim that task interdependence is neither necessary nor sufficient for epistemic interdependence to exist between the agents performing the tasks, and that interdependence between agents is necessary but not sufficient for epistemic interdependence to exist between them. It is the existence of broad incentives or rewards that result in agent interdependence. Let us consider their example. With specialisation in a pin factory, two agents A and B, make 300 pin heads and 300 pin tails respectively, which can be soldered together to produce 300 pins by the end of the day. The tasks are interdependent, because 300 pin heads alone or 300 pin tails alone are worthless, whereas 300 pins are worth something to the factory owner. When each worker is paid as long she produces her 300 heads or 300 tails respectively, they face a narrow incentive structure. Broad incentives correspond to the reward of individual actions or their results in a manner that makes them at least partly dependent on the other agent’s actions. If one of the workers in the example above was paid only on delivering 300 complete pins made by assembling the heads she makes with the tails the other worker makes, she is in effect facing broad incentives. If both are paid their wages only on delivery of 300 complete pins, both face broad incentive. Puranam et al. (2012) through their discussion have stressed the importance of predictive knowledge, which involves taking a cognitive perspective; that is, considering others’ having mental states and reasoning about these mental states during decision making. While their view was in context of organisation design, the cognitive perspective is very strong in distributed artificial intelligence community, as Section 3.1.2 (next) demonstrates.

### 3.1.2 Social Dependence Theory and the Likes

Castelfranchi et al. (1992) discussed the various types of dependence relationships between agents due to resources or goals (or actions), as shown in Figure 3.1.2. These dependence relationships may exist but be unknown by the dependent parties, known, wanted or even created. They consider resource dependence to be different from a social dependence. While resource dependence is based on an external object or event, a social dependence requires some form of dependence on other agents, for example, for an action performed by another agent. They define different patterns of social dependence relationships:

- **OR-Dependence:** An agent has many options due to the fact that either there are many agents that can perform an action that is required or that many different actions may help achieve the same goal but are performed by different agents.
Figure 3.1.2: Interdependence forms based on social dependence theory.

- **AND-Dependence**: An agent is dependent on many agents due to many agents required in achieving an action or goal (multiparty dependence), or an agent is dependent on one agent but for many actions or goals (multigoal dependence).

- **Bilateral Dependence**: This happens when agents are dependent on each other for actions or goals. Agents may be dependent due to a common goal or action (mutual dependence) or due to different goals (reciprocal dependence).

A dependent agent may either rely solely on the benevolence of others to get other agents to help him/her achieve the action or goal, or attempt to influence others, exercise power over others if possible or identify and use mutual or reciprocal dependence.

Sichman [1998] goes a step further by explicitly accounting for awareness of these dependence relationship. They defined social reasoning as the ability of an intelligent agent to reason about others during decision making process. They provide their agents a particular data structure, which they call “external descriptions”. These descriptions capture the agent’s goals, actions, resources, and plans. When shared, these external descriptions form a key part of the common ground, or by a little stretch, a key part of the shared mental model. An agent is then independent and “autonomous” if it does not depend on any action.
or resource of another agent in accomplishing the agent’s goals. If there is dependence, then mutual dependence is based on a single goal, while reciprocal dependence is based on different goals. When both agents are aware of these dependencies (via external descriptions), the agents are said to mutually believe the dependence. Otherwise, the dependence is locally believed, meaning that one of the agents is aware of the dependence.

Castelfranchi (1998) extend their discussion on social dependence (Castelfranchi et al., 1992) by stressing the importance of social commitments in social actions. They define goal-directed behaviour as one in which the goal not only controls and guides the agent actions but also determines action search, selection and its failure or success. The agents considered in this definition are cognitive agents, that is, agents that have an internal representation or mental state consisting of at least goals, plans, and beliefs. An action taken by a cognitive agent is social action (SA) if that action is taken considering another agent as being a cognitive agent with goals. One form of social action requires delegation, which requires some level of dependence on another agent, the delegated agent. There are three situations that can arise. The delegated agent may be completely unaware of this dependence (and exploitation) resulting in unilateral weak delegation, the delegating party may induce this delegation by some means resulting in delegation by induction (e.g. fisherman using baits to exploit the reactive behaviour of fish), or delegation by acceptance, in which parties agree on, potentially negotiate and are aware of the delegation. Delegation by acceptance requires social goal adoption, that is, the delegated party adopts a goal in service of the requirements of the delegating party. According to Castelfranchi (1995, 1998), social goals form the glue to joint action and requires not only individual commitment of an agent to his/her actions or collective commitment of the team to the collective intentions but also social commitment where each agent is committed to another for some goal or action.

3.1.3 Summary

The different types of dependence and interdependence relationships and the papers describing these types have been summarised in Table 3.1.1. There are a number of lessons learnt from the discussion on social dependence theories (Castelfranchi, 1995, 1998, Castelfranchi et al., 1992, Sichman, 1998). Agents taking a cognitive perspective on others is a key requirement. Formation of the mutual or reciprocal dependence, that is, interdependence gives rise to cooperation. Finally, these interdependence relationships form a key requirement to team formation and existence in addition to the setup and maintenance of joint goals or intentions. Without the interdependence relationships, agents will satisfy the requirements of joint intentions (Levesque et al., 1990) but will not actually be con-
considered a team. Interdependence coupled with commitments form the key requirements for teamwork. These interdependence relationships may exist due to capability deficiency or common resources or may be created specifically to form a team or to avoid power or control imbalance.

In addition to agent interdependence due to goals, there are many other forms of interdependence according to organisation psychology literature that have already been introduced. In the next section, semi-formal definitions of task, agent, and goal interdependence are presented.

### 3.2 Running Example

In choosing the domains, the following were considered. The domain had to be a collaborative multiagent one in which members worked together to achieve the team goals. The types of team’s joint tasks had to range from loosely coupled to tightly coupled with joint actions, allowing for experiments with different forms of interdependence relationships. There had to be a foreseeable relationship between the members sharing information to establish SMM

<table>
<thead>
<tr>
<th>Type</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Interdependence)</td>
<td></td>
</tr>
<tr>
<td>Resource dependence</td>
<td>Castelfranchi et al. (1992), Lau et al. (2015)</td>
</tr>
<tr>
<td>Social dependence</td>
<td>Castelfranchi et al. (1992), Lau et al. (2015)</td>
</tr>
<tr>
<td>OR-Dependence</td>
<td>Castelfranchi et al. (1992), Lau et al. (2015)</td>
</tr>
<tr>
<td>AND-Dependence</td>
<td>Castelfranchi et al. (1992), Lau et al. (2015)</td>
</tr>
<tr>
<td>Behavioural Interdependence</td>
<td>Courtright et al. (2015), Wageman (2001)</td>
</tr>
<tr>
<td>Structural Interdependence</td>
<td>Courtright et al. (2015), Wageman (2001)</td>
</tr>
<tr>
<td>Task Interdependence</td>
<td>Saavedra et al. (1993), Thompson (1967)</td>
</tr>
<tr>
<td>Outcome Interdependence</td>
<td>Saavedra et al. (1993), Wageman (2001)</td>
</tr>
<tr>
<td>Goal Interdependence</td>
<td>Deutsch (1949), Saavedra et al. (1993)</td>
</tr>
<tr>
<td>Epistemic Interdependence</td>
<td>Puranam et al. (2012)</td>
</tr>
<tr>
<td>Hard Interdependence</td>
<td>Johnson et al. (2014), Wei (2015)</td>
</tr>
<tr>
<td>Soft Interdependence</td>
<td>Johnson et al. (2014), Wei (2015)</td>
</tr>
</tbody>
</table>

**Table 3.1.1:** Dependence or interdependence types and papers defining or explaining these types.
and improvement of the team performance. This means that agents, either homogeneous or heterogeneous, had to share information to enable the team to achieve the team goals efficiently and/or have situations where if team members fail to achieve their goals, the team fails to achieve the team goals. The information shared could be the agent’s intentions or other information relevant to improving the team performance. The agent’s plans had to exhibit some level of dependency on the intentions of the other agents, that is, an agent had to consider what others were doing to achieve the team goals. A partially observable environment was desired as this would generally require agents to make communication decisions related to SMM and remain coordinated. The team activities had to be bounded in terms of time, that is, there had to be a clear start and end to the team activities to assist in the analysis of sharedness of SMM. The Blocks World for Teams (BW4T) (Johnson et al., 2009) domain fits this requirement.

To aid the discussions of various concepts, the Blocks World for Teams (BW4T) (Johnson et al., 2009) domain will be used throughout this and remaining chapters. The BW4T domain is an extension of the classical Blocks World domain. In BW4T, a team of agents find and deliver coloured blocks in a sequence. The office-like environment has a set of rooms, each containing coloured blocks, and a drop zone. Agents have a map of the area but do not know the location of the required blocks. The agents search the rooms, find the required blocks and drop these in the drop zone. Agents have to go to each room to perceive the blocks that are present in it. Agents cannot see each other but can communicate with others. An example map is shown in Figure 3.2.1. Each room has one door. The team’s joint task, i.e. the sequence of colours, is displayed at the bottom left. A black triangle appears on top of a colour if the colour has been dropped off. The room above the joint task is the drop zone. The agents are represented by either black squares if they are not currently holding a block or the colour the agent is holding, and their names are displayed in red. In the version of the BW4T proposed by Johnson et al. (2009), each agent can lift one colour at a time, all agents can see and lift every colour, only one agent is required to lift each block, the colours should be dropped in the same order as it appears in the team’s joint task, and finally, if multiple agents drop different blocks in the drop zone simultaneously, only the block matching the first undelivered colour will be accepted by the testbed.

There is no particular requirement that teams must have SMM to complete the BW4T joint activity. But what are the advantages if they do? Knowing about which room each agent is going to search will help the team complete the search faster by avoiding multiple agents searching the same room. If the agents share information about the blocks that they
see, this can help coordinate the agents in terms who can deliver which block quicker and complete the task faster. For example, it would very inefficient to have all agents attempting to deliver the first block. This example illustrates that when agents establish joint intentions in relation to their search and block delivery activities, the team can achieve their joint goal faster. The faster completion times can be used as a proxy for team performance. Therefore, there are advantages if a team solving the BW4T task establish and use SMM.

There are other interesting properties of the domain and the features of the testbed that simulates this domain ([Johnson et al., 2009](#)). The agents have partial observability of the environment. Without communication, an agent can only know about the blocks present in the rooms that the agent has searched. Similarly, the location of teammates may be unknown, unless this information can be communicated or inferred from the communicated information.

### 3.3 Task Interdependence

This section presents the formal definition of task interdependence. Agents are assumed to be systems capable of autonomous *actions* on the environment. *Tasks*, which may require one or more actions, are performed to achieve goals. *Goals* represent the state of the environment that the agent wants to achieve. Task interdependence exists when the value generated from performing each task is different when the tasks are performed together versus when the tasks are performed separately ([Puranam et al., 2012](#)).
Definition 3.3.1 (Task). The following grammar defines tasks:

\[ t ::= \text{act} \mid t_1; t_2 \mid t_1 \parallel t_2 \mid t_1 \|\| t_2 \]

in which \( \text{act} \) is an atomic action of an agent, \( t_1; t_2 \) represents sequential execution such that \( t_1 \) executes before \( t_2 \), \( t_1 \parallel t_2 \) represents interleaved execution, and \( t_1 \|\| t_2 \) represents true concurrent execution. The difference between interleaving and true concurrency is that in interleaving, actions cannot execute simultaneously, whereas in true concurrency, actions (and entire tasks) can execute simultaneously. We use the shorthand \( t_1 \odot t_2 \) to mean composition of \( t_1 \) and \( t_2 \) using any of the operators \( ; \), \( \parallel \), or \( \|\| \), and the shorthand \( \ol{t_1 \parallel t_2} \) to mean the complement of (any composition other than) \( t_1 \odot t_2 \); for example, \( \ol{t_1; t_2} \) means either \( t_1 \parallel t_2 \), \( t_1 \|\| t_2 \), or \( t_2; t_1 \). The operators \( \parallel \) and \( \|\| \) are commutative, so the complement \( \ol{t_1 \parallel t_2} \) does not contain \( t_2 \parallel t_1 \), and similarly for \( \|\| \). We consider the semantics of interleaved concurrency to be that of CSP (Hoare, 1985). For true concurrency, we consider the semantics to be that of independent actions (Winskel and Nielsen, 1993).

However, this chapter does not get into the details of defining or presenting semantics for these as these are primarily used as a tool for discussion and understanding teamwork, and not for defining computational ways of reasoning about interdependence.

Definition 3.3.2 (Task Value). The tangible value generated by executing a task \( (t) \) in some context \( (c) \) is measured by \( \mathcal{V}_c(t) \).

Definition 3.3.3 (Task Independence). Tasks are independent if and only if the value generated by executing the composed tasks is the same as the sum of the value of each task in isolation. That is, we gain nothing by composing the two tasks. More formally:

\[ \text{TaskIndep}(t_1, t_2) \iff \mathcal{V}_c(t_1) + \mathcal{V}_c(t_2) = \mathcal{V}_c(t_1 \odot t_2) \quad (3.1) \]

We can generalise this to \( n \geq 2 \) tasks:

\[ \text{TaskIndep}(t_1, \ldots, t_n) \iff \mathcal{V}_c(t_1) + \mathcal{V}_c(t_2) = \mathcal{V}_c(t_1 \odot t_2) \land \mathcal{V}_c(t_1) + \mathcal{V}_c(t_3) = \mathcal{V}_c(t_1 \odot t_3) \land \ldots \land \mathcal{V}_c(t_{n-1}) + \mathcal{V}_c(t_n) = \mathcal{V}_c(t_{n-1} \odot t_n) \quad (3.2) \]

which is equivalent to

\[ \text{TaskIndep}(t_1, t_2) \land \text{TaskIndep}(t_1, t_3) \land \ldots \land \text{TaskIndep}(t_{n-1}, t_n) \]

Note: A value function that takes as arguments more than two tasks has not been
Example 3.3.1 (Obstacle detection in human agent team (Independence)). One task required in the DARPA Robotics Challenge is obstacle detection (Johnson et al., 2014). Consider two tasks, \( \text{SenseObsHum} \) and \( \text{SenseObsRob} \). The human detects obstacles via a user interface while the robot uses sensors, which are assumed to work perfectly for this example. The two tasks are independent because the value of identifying an obstacle together is the same as both the human and robot identifying them independently (I am assuming here that both parties can execute these tasks accurately). Given that the context \( c \) is detecting obstacles, independence is explained by applying Equation [3.1]:

\[
\mathcal{V}_c(\text{SenseObsHum}) + \mathcal{V}_c(\text{SenseObsRob}) = \mathcal{V}_c(\text{SenseObsHum} \odot \text{SenseObsRob})
\]

Definition 3.3.4 (Task Dependence). Composed execution of dependent tasks (using any composition operator) generates a value greater than independent execution of those tasks. If task \( t_2 \) is dependent on \( t_1 \) then formally:

\[
\text{TaskDep}(t_1 \mid t_2) \iff \mathcal{V}_c(t_1 \odot t_2) > \mathcal{V}_c(t_1) + \mathcal{V}_c(t_2)
\] (3.3)

As with task independence, task dependence can be generalised to \( n \geq 2 \) tasks.

Definition 3.3.5 (Soft/Hard Task dependence). A soft (hard) task dependence is when a task is dependent on another task, but the value of executing the dependent task is non-zero (zero). Formally:

\[
\text{SoftTaskDep}(t_1 \mid t_2) \iff \text{TaskDep}(t_1 \mid t_2) \land \mathcal{V}_c(t_2) > 0
\]

\[
\text{HardTaskDep}(t_1 \mid t_2) \iff \text{TaskDep}(t_1 \mid t_2) \land \mathcal{V}_c(t_2) = 0
\] (3.4)

Definition 3.3.6 (Task Interdependence). Two tasks are interdependent if there is a two-way dependency between the tasks. Formally:

\[
\text{TaskInterdep}(t_1, t_2) \iff \text{TaskDep}(t_1 \mid t_2) \land \text{TaskDep}(t_2 \mid t_1)
\] (3.5)

This can be generalised to \( n \geq 2 \) cases as is done for task independence.

Definition 3.3.6 captures the essence of interdependence but not the differences between the different types of interdependencies. Therefore, next we formalise several types of interdependencies, based on organisational theory (Saavedra et al., 1993).
3.3.1 Types of Task Interdependence

According to organisational theorists and sociologists (Puranam et al., 2012, Saavedra et al., 1993, Thompson, 1967), there are four types of task interdependence: sequential, reciprocal, team, and pooled, as shown in Figure 3.3.1. In the following sections, we formalise these definitions. We give definitions for pairs of tasks, but each is generalisable to \( n \geq 2 \), as done for independence.

**Figure 3.3.1:** Four types of tasks form a hierarchy of increasing levels of interdependence.

**Definition 3.3.7 (Sequential Task Interdependence).** The value generated by sequential composition of two interdependent tasks is greater than any other composition. Formally:

\[
TaskInterdep(t_1, t_2) \land V_c(t_1; t_2) > V_c(t_1; t_2) \tag{3.6}
\]

Note that this is not simply task dependence, because the value of performing the first task can be higher if the second task is performed; e.g. picking up a hose is only valuable if the hose is subsequently attached.

**Example 3.3.2 (BW4T).** This is the classical BW4T scenario discussed in Section 3.2 above. To simplify the process, consider that the team is required to deliver only two colours with \( t_1 \) as a label for delivering the first colour, and \( t_2 \) for the second. These tasks are abstract, and this and future examples do not take into consideration the sub-tasks of each task. Consider the following measure of performance. The team gets a reward of 1
point for completing the team goal. In addition, the team gets 1 point for delivering each colour. The team gets 1 additional point for using a composition of the tasks that allows the team to complete the task the fastest. Using the reward structure, we can see that:

$$\mathcal{V}_c(t_1; t_2) > \mathcal{V}_c(t_1 \parallel t_2)$$

In this situation reciprocal execution will result in a score of 0 as an agent cannot partially deliver a block. Concurrent execution is definitely possible but dropping both blocks at the same time will result in the first block being accepted only - note that the first block needs to be delivered before the second one can be dropped. This means that a sequential execution will give the team 4 points as opposed to 1 point with concurrent execution.

**Definition 3.3.8 (Reciprocal Task Interdependence).** The value of reciprocally interdependent tasks is strictly greater than the value generated by any other composition. The notion of temporal lag between tasks is captured by the interleaved execution of the tasks. Formally:

$$\text{TaskInterdep}(t_1, t_2) \land \mathcal{V}_c(t_1 \parallel t_2) > \mathcal{V}_c(t_1 \mid t_2)$$ (3.7)

**Example 3.3.3 (Writing a paper (Reciprocal Interdependence)).** Consider one person writing a paper and another proof reading it and providing feedback, represented by the tasks WritePaper and ReadPaper. While these tasks can be executed sequentially, in many cases, more value is generated if the reader provides feedback on individual sections through the writing. But doing these truly concurrently would require the reader to read over the writer's shoulder, which is presumably not as valuable. Applying Equation 3.7:

$$\mathcal{V}_c(\text{WritePaper} \parallel \text{ReadPaper}) > \mathcal{V}_c(\text{WritePaper} \mid \text{ReadPaper})$$

**Example 3.3.4 (BW4T).** Recall that the team's goal is to deliver a sequence of colours. Consider the sequence of four colours: red, blue, green, yellow. We make a slight modification to the original scenario. Let's assume that we have a team of two heterogeneous agents, $a_1$ and $a_2$. Agent $a_1$ has the capability to detect and deliver red and green blocks while $a_2$ can deliver blue and yellow blocks. Further, we aggregate the two sub-tasks of $a_1$ and label these as $t_1$ (where $t_1$ refers to delivering red and green blocks), and the tasks of $a_2$ as
Since the blocks need to be delivered in the order the colours appear in the sequence, the two tasks need to be executed reciprocally, that is, delivery of red (by \(a_1\)), then blue (by \(a_2\)), followed by green and yellow (by \(a_1\) and \(a_2\) respectively). Using the same reward structure as in Example 3.3.2, it is easy to see that:

\[
V_c(t_1 \parallel t_2) > V_c(t_1 \parallel t_2)
\]

If two blocks, e.g. red and blue are dropped at the same time, only the first block, the red one, will be accepted. The blue block can only be delivered after the red block. This means that sequential or parallel execution of \(t_1\) and \(t_2\) will result in the team delivering the first two blocks, which gives the team a total of 2 points as opposed to 6 points with reciprocal execution. Here parallel execution refers to the delivery of two colours by different agents at the same time.

**Definition 3.3.9 (Team Task Interdependence).** Team task interdependence exists for joint activities — that is, when agents must jointly execute one or more actions. There is no temporal lag between task executions, because tasks are executed simultaneously. For such tasks, the value of a truly concurrent composition is strictly greater than any other composition. Formally:

\[
\text{TaskInterdep}(t_1, t_2) \land V_c(t_1 \parallel\parallel t_2) > V_c(t_1 \parallel\parallel t_2)
\]

**Example 3.3.5 (Cooperative object transportation (Team Interdependence)).** Two robots are to move a table that is too heavy for one robot to move by itself. Consider two tasks \(\text{MoveEnd}_1\) and \(\text{MoveEnd}_2\), where \(\text{MoveEnd}_1\) means that one robot will lift and move the table at one of the two ends. Only true concurrent execution of the tasks will be able to move the table. Applying Equation 3.8:

\[
V_c(\text{MoveEnd}_1 \parallel\parallel \text{MoveEnd}_2) > V_c(\overline{\text{MoveEnd}_1} \parallel\parallel \text{MoveEnd}_2)
\]

**Example 3.3.6 (BW4T).** Let’s consider another change to the scenario, which is the introduction of heavy blocks. Such blocks require two agents to pick up. Specifically, one agent can carry and deliver the block (e.g. by means of a trolley) but cannot pickup (load) the block because it is heavy. To simplify the example, assume that the team is required to deliver only one colour, and receive the same reward as discussed in Example 3.3.2. To
stick with convention of tasks, we label agent $a_1$ picking up the block as $t_1$ and $a_2$ doing the same as $t_2$. It is easy to see that:

$$V_c(t_1 \parallel t_2) > V_c(t_1 \parallel t_2)$$

That is, the team accrues a reward of 3 with concurrent execution. If sequential and reciprocal executions were possible, these would result in slower completion times and team would get 2 points.

**Definition 3.3.10 (Pooled Task Interdependence).** Composition of tasks exhibiting pooled interdependence result in greater value than of executing the tasks independently. This is simply equivalent to the initial definition of interdependence, but with no constraint on how the tasks are composed. Each task contributes its share towards the group outcome, and failure of any task means that the goal will not be achieved. This is a *weak* form of interdependence where the tasks are interdependent but interdependence is not sequential, reciprocal, or team.

### 3.4 AGENT INTERDEPENDENCE

In the previous section, task interdependence is agnostic on who executes the task or who gains the value of the execution. In this section, we consider how *agents* can be independent, dependent, and interdependent. Throughout this section, we use the same grammar to define tasks, and $a_i, t_j$ to mean that agent $a_i$ executes task $t_j$. The semantics remains the same, essentially ignoring the agent prefixes, which are just labels. This means that for composed tasks, if two agents $a_1$ and $a_2$ both execute the same action $act$, this is a synchronised event (they must execute together). We treat such synchronisations as *joint* actions.

According to Puranam et al. (2012), agent interdependence results when “...the reward to A from A’s actions depends on the actions taken by B...”. We formalise the agent’s reward using the notion of *utility*. We introduce *soft* and *hard* agent dependence and interdependence. Hard dependence means that the agent cannot complete its tasks independently (utility is zero), while for soft dependence, it can complete the task, but receives more utility if it works with other agents.

**Definition 3.4.1 (Agent Utility).** An agent’s utility when executing a task $t$ in a context $c$ is defined as $U_c^u(t)$. Each agent has their own utility function, which could be associated
with measures such as how quickly the agent finishes a task.

**Definition 3.4.2 (Agent Independence).** Two agents, $a_1$ and $a_2$, are independent if each agent’s utility is not affected by the actions of the other agent. Formally:

$$\text{AgentIndep}(a_1, a_2) \iff \forall t_1, t_2 : U_c^{a_1}(a_1.t_1 \odot a_2.t_2) = U_c^{a_1}(a_1.t_1) \land U_c^{a_2}(a_1.t_1 \odot a_2.t_2) = U_c^{a_2}(a_2.t_2)$$  \hspace{1cm} (3.9)

**Definition 3.4.3 (Agent Dependence).** Agent $a_2$ is dependent on $a_1$ for task $t_2$ if the utility of $a_2$ increases when agent $a_1$ executes a task $t_1$. Dependence is related to specific tasks; that is, it is possible to be dependent on another agent for some tasks, but not others. Formally:

$$\text{AgentDep}(a_1 | a_2, t_1, t_2) \iff U_c^{a_2}(a_1.t_1 \odot a_2.t_2) > U_c^{a_2}(a_2.t_2)$$  \hspace{1cm} (3.10)

Agent dependence can also be defined for some unnamed task as:

$$\text{AgentDep}(a_1 | a_2) \iff \exists t_1, t_2 : \text{AgentDep}(a_1 | a_2, t_1, t_2)$$  \hspace{1cm} (3.11)

**Definition 3.4.4 (Soft/Hard Agent Dependence).** Agent $a_2$ has a soft (hard) dependence on agent $a_1$ if the utility of $a_2$ at its task is non-zero (zero) and the utility of $a_2$ increases when agent $a_1$ executes one of its tasks. Formally:

$$\text{SoftAgentDep}(a_1 | a_2, t_1, t_2) \iff \text{AgentDep}(a_1 | a_2, t_1, t_2) \land U_c^{a_2}(a_1.t_1 \odot a_2.t_2) < U_c^{a_2}(a_2.t_2)$$

$$\text{HardAgentDep}(a_1 | a_2, t_1, t_2) \iff \text{AgentDep}(a_1 | a_2, t_1, t_2) \land U_c^{a_2}(a_1.t_1 \odot a_2.t_2) = 0$$  \hspace{1cm} (3.12)

**Definition 3.4.5 (Agent Interdependence).** Agents $a_2$ and $a_1$ are interdependent if there is a two-way dependence between the agents. More formally:

$$\text{AgentInterdep}(a_1, a_2, t_1, t_2, t_3, t_4) \iff \text{AgentDep}(a_1 | a_2, t_1, t_2) \land \text{AgentDep}(a_2 | a_1, t_3, t_4)$$  \hspace{1cm} (3.13)

This definition demonstrates that agent interdependence is not merely task interdependence over utilities, because two pairs of dependent tasks between the agents is enough to establish interdependence. This is consistent with Puranum et. al.’s view ([Puranam et al., 2012](#)) that task interdependence is neither necessary nor sufficient to establish agent interdependence.
As with agent dependence, we can omit the tasks \( t_1 \) to \( t_4 \) to avoid explicitly naming the tasks that are (inter)dependent.

**Definition 3.4.6** (Soft/Hard Agent Interdependence). Soft (hard) agent interdependence between two agents exists when both agents have soft (hard) dependence on each other. Formally:

\[
\begin{align*}
\text{SoftAgentInterdep}(a_1, a_2) & \iff \\
\text{SoftAgentDep}(a_1 | a_2) \land \text{SoftAgentDep}(a_2 | a_1) \\
\text{HardAgentInterdep}(a_1, a_2) & \iff \\
\text{HardAgentDep}(a_1 | a_2) \land \text{HardAgentDep}(a_2 | a_1)
\end{align*}
\]

(3.14)

**Example 3.4.1** (Object transportation (Hard Agent Interdependence)). In Example 3.3.5, two robots \( (r_1 \text{ and } r_2) \) moved a heavy table. Assume that the utility is: \( U_c^{a_i}(t) = \text{total\_time} \) and the context \( c \) is object transportation. The utilities of the individual tasks are zero because the robots cannot move the table alone. When the two robots cooperatively move the table, the utility of each increases:

\[
\begin{align*}
U_c^{r_1}(r_1.MoveEnd1) &= U_c^{r_2}(r_2.MoveEnd2) = 0 \\
U_c^{r_1}(r_1.MoveEnd1 || r_2.MoveEnd2) &> U_c^{r_1}(r_1.(MoveEnd1 \odot MoveEnd2)) \land \\
U_c^{r_2}(r_1.MoveEnd1 || r_2.MoveEnd2) &> U_c^{r_2}(r_2.(MoveEnd1 \odot MoveEnd2))
\end{align*}
\]

**Example 3.4.2** (BW4T (Soft Agent Interdependence)). Let's assume that a team of two homogeneous agents are required to deliver the colour sequence: red, blue, and since the agents are homogeneous, they can both deliver both colours. Consider the situation where there are more than one of each colour present in the environment and that both agents know the location of the blocks. It is, therefore, possible for both agents to deliver the blocks independently. However, if both agents drop a red block then only one will be accepted. If we take the same performance measure as in Example 3.3.2 but also give the agents 2 points for avoiding delivering the same colour and -1 in case of task duplication (delivering the same colour), we can see that if \( a_1 \) delivers the red block before:

\[
U_c^{a_1}(a_1.red) > U_c^{a_2}(a_2.red)
\]

Similarly, it is possible for each agent to get 4 points (2pts for delivery of each colour, 1pt for completing team task, 0pts for fastest strategy, 1pt (2 + -1) as utility) for delivering
at least one colour first. However, if both agents coordinate with each other and create
a situation where $a_1$ delivers red and $a_2$ blue, then both agents can get 2pts in utility,
(potentially) additional point for fastest strategy leading to 6pts in total for each. This
simply implies that:

$$
\mathcal{U}_{c}^{a_1}(a_1, red; a_2, blue) > \mathcal{U}_{c}^{a_1}(a_1, (red \odot blue)) \land \\
\mathcal{U}_{c}^{a_2}(a_1, red; a_2, blue) > \mathcal{U}_{c}^{a_2}(a_2, (red \odot blue))
$$

3.5 **Goal Interdependence**

Goal interdependence shapes the nature of interaction between agents [Deutsch, 1949]. It
refers to the extent to which the individual agent’s goals are aligned with the group goals.
If the agent’s goals are positively correlated with the group goals then the agents exhibit
a cooperative behaviour because the agents believe that attainment of their goals promotes
the attainment of others’ goals, and vice versa. On the other hand, if the agent and group
goals are misaligned or negatively correlated, competitive behaviour may be observed.

**Definition 3.5.1** (Goal Achievement). Let $\text{achieve}(g)$ be a test of whether a goal $g$ is
achieved or not.

**Definition 3.5.2** (Goal Independence). Two goals, $g_1$ and $g_2$, are independent if the
achievement of each goal is not influenced by the achievement of the other. More formally:

$$
\text{GoalIndep}(g_1, g_2) \iff (\text{achieve}(g_1 \land g_2) \leftrightarrow \text{achieve}(g_1) \land \text{achieve}(g_2))
$$

(3.15)

That is, achieving goals $g_1$ and $g_2$ independently will achieve the greater goal of $g_1$ and $g_2$,
and vice-versa.

**Definition 3.5.3** (Goal Dependence). Goals $g_2$ is dependent on $g_1$ if the achievement of $g_2$
requires the achievement of $g_1$. That is, if goal $g_2$ is achieved, it must have been that goal
$g_1$ has also been achieved. More formally:

$$
\text{GoalDep}(g_1 \mid g_2) \iff (\text{achieve}(g_2) \Rightarrow \text{achieve}(g_1))
$$

(3.16)

**Definition 3.5.4** (Goal Interdependence). Two goals, $g_1$ and $g_2$, are interdependent if the
achievement of each goal requires the achievement of the other. This means that there is a
two-way dependence between the goals. Formally:

\[ \text{GoalInterdep}(g_1, g_2) \iff (\text{GoalDep}(g_1 | g_2) \land \text{GoalDep}(g_2 | g_1)) \] (3.17)

Using the formalism of goal interdependence, we define the concepts of cooperative goal interdependence and competitive goal interdependence.

**Definition 3.5.5** (Cooperative Goal Interdependence). Two goals, \( g_1 \) and \( g_2 \), are cooperatively interdependent if the achievement of each goal implies the achievement of the other. Formally:

\[ \text{CoopGoalIndep}(g_1, g_2) \iff (\text{achieve}(g_2) \iff \text{achieve}(g_1)) \] (3.18)

**Definition 3.5.6** (Soft/Hard Competitive Goal Interdependence). Two goals, \( g_1 \) and \( g_2 \), have a soft (hard) competitive interdependence if (and only if) the achievement of one goal implies the failure of the other. Formally:

\[ \text{SoftCompGoalIndep}(g_1, g_2) \iff (\text{achieve}(g_2) \Rightarrow \neg \text{achieve}(g_1)) \]
\[ \text{HardCompGoalIndep}(g_1, g_2) \iff (\text{achieve}(g_2) \iff \neg \text{achieve}(g_1)) \] (3.19)

For soft competitive goal interdependence, the goals are mutually exclusive: it is never the case that both agents achieve their goals. However, it is possible that both goals are not achieved. The condition \( \text{achieve}(g_1) \Rightarrow \neg \text{achieve}(g_2) \) is a contrapositive of the first line in Equation 3.19, so is not explicitly specified.

**Example 3.5.1** (Managing Sick Trees). Consider an arborist and a woodcutter who are tasked with the identification and removal of sick trees respectively. The goal of the arborist is to identify sick trees and that of the woodcutter is to remove sick trees. The two goals are cooperatively interdependent because the achievement of both are required to get rid of sick trees.

**Example 3.5.2** (Robots sorting objects). Consider two robots sorting objects and placing them in designated locations, with each robot’s goal being to sort the most objects. This is hard competitive interdependence because only one robot can sort more than the other.

**Example 3.5.3** (BW4T). In BW4T, the agents’ goals are clearly interdependent. Consider two agents required to deliver two colours, red and blue and more specifically the goal, \( g_1 \), of \( a_1 \) is getting the red block and goal, \( g_2 \), of \( a_2 \) the blue one. In this situation, while it is the case that both agents achieving their individual goals achieves the team goal, agent \( a_1 \) relies on \( a_2 \) to achieve her goal. If \( a_1 \) achieves his goal and \( a_2 \) does not achieve her goal then
the team task will be incomplete resulting in a failure. Therefore, the agents are clearly
interdependent for different goals. According to the formalism provided by Castelfranchi et al. (1992), this is an example of reciprocal goal interdependence.

Similarly, in case of heavy blocks, which requires two agents to pickup the block, the
agents are interdependent for the same goal. Both need to adopt this goal and achieve
the goal concurrently if the goal of lifting the heavy block is to be achieved. According to
the formalism provided by Castelfranchi et al. (1992), this is an example of mutual goal
interdependence.

3.6 Example: Analysis of Situation using Interdependence Types

The objective of this example is to demonstrate that having an understanding of the different
forms of interdependence, for example, task interdependence, can be helpful when designing
systems which require members to be interdependent. One of the tasks in the DARPA
Robotics Challenge was for the simulated robot to grasp a hose (Johnson et al., 2014). The
robot and the operator each lacked the capability to correctly position the robot’s hand
for the robot to grasp the hose. Consider two tasks \( \text{PosHandOp} \) and \( \text{PosHandRb} \), which
represents the operator and the robot positioning the robot’s hand respectively. The two
tasks are interdependent because there is a need to execute both tasks to be able to grasp
the hose.

Given that the two tasks are interdependent, and that the two tasks are allocated to
and executed by the two agents, it follows that the agents are interdependent on each other,
that is, there is agent interdependence. In addition, if the goals of the agents are considered
such that each task achieves one or more goals, then there is goal interdependence also. For
example, if successful completion of \( \text{PosHandOp} \) achieves the operator’s goal of providing
assistance to the robot and \( \text{PosHandRb} \) achieves the robot’s goal of grasping the hose, then
there is goal interdependence. Let us focus solely on task interdependence for the following
discussions. The tasks could execute sequentially: the robot executes its task before the
operator, and the tasks repeat until the robot’s hand is positioned correctly. Alternatively,
the tasks can be reciprocally executed such that while the robot tries to position its hand,
the operator intervenes as necessary.

For a truly sequential execution of the tasks, the designer must program the robot to only
receive the operator’s command after the robot has positioned the hand. This requirement
provides hints of when the agents should communicate and what type of information is
necessary as far as the SMM is concerned. For the operator to know that the robot has
finished an attempt and that the robot may need assistance, the termination of robot’s task needs to be communicated to the operator and commencement of the operator’s task needs to be communicated to the operator. This establishes SMM between the two agents in relation to the two tasks. This shared understanding will result in both agents knowing when to act.

However, to enable reciprocal or interleaved execution, the designer needs to provision the robot to accept and execute commands from the operator as it is engaged in its task, and provide the necessary interfaces for the operator. The choice of the execution method depends on the objective of the designer. For example, if the objective is to minimise the temporal lag between the tasks, then reciprocal execution is appropriate. That is:

\[ V_c(PosHandOp \parallel PosHandRb) > V_c(PosHandOp; PosHandRb) \]

As such, the robot should be designed to accept the operator commands that override its own positioning tasks as it tries to position its hand. This requires a slightly different type of information as far as SMM is concerned. Both agents now need to know when the intervention is happening, what the actual commands from the operator are to the robot, and whether the robot is accepting and acting on the operator’s interventions. There is also bound be to more communication compared to the previous approach to keep the SMM updated and for the two agents to execute the task effectively.

The formalism provides designers with a more systematic and rigorous way to analyse the different design possibilities, and to weigh up different design decisions. Further, depending on the type of (task) interdependence, the designer may need to implement different coordination mechanisms between the agents, which may require agents exchanging different types of information to establish SMM.

3.7 Discussion

This section describes two closely related papers: Castlefranchi et al.’s (Castelfranchi et al., 1992) formal definition of agent interdependence and Johnson et al.’s (Johnson et al., 2014) informal work on task interdependence. Castlefranchi et al. (1992) formally define non-social and social dependence, and describe the complex patterns of dependence relationships, such as unilateral and bilateral dependence, AND-Dependence, and OR-Dependence. We go further than Castlefranchi et al.’s formal definitions of agent interdependence (Castelfranchi et al., 1992) by considering
soft agent interdependence. In Castelfranchi et al.’s formalism, a dependence exists from agent $a_1$ to agent $a_2$ if and only if $a_1$ can only complete the task with $a_2$. By considering utility as the measure of task completion, we generalise agent interdependence, allowing us to capture Castelfranchi et al.’s notion of (inter)dependence as hard (inter)dependence (utility for $a_1$ is 0), as well as to consider a notion of soft (inter)dependence (utility for $a_1$ is non-zero, but less than the composed case). Castelfranchi et al. consider that bilateral dependence can either be mutual or reciprocal, differing based on whether the agents’ goals are shared (mutual) or separate (reciprocal).

Johnson et al. (2014) looked at numerous existing definitions of interdependence, including the work of sociologists (Thibaut and Kelley, 1959), and we build on their final definition of task interdependence. We believe that their definitions of interdependence are equivalent to our formal definitions, including their notion of soft and hard interdependence, although their definitions are informal, so these equivalences are not straightforward to assess. However, they do not explicitly consider agent or goal interdependence. Wei (2015) presented a formal graphical language to identify potential interdependence relationships and provided formal definitions of soft and hard task interdependence. In future work, it will be useful to analyse the different forms of task interdependence defined in this chapter using the language provided by Wei (2015).

3.8 Conclusion

Existing literature (Castelfranchi et al., 1992, Johnson et al., 2014) makes a strong case that interdependence is an important concept for the design of human agent collaboration. Taking a more philosophical view of a team and according to discussions of Castelfranchi (1993, 1998), Castelfranchi et al. (1992) and Levesque et al. (1990), while joint intentions form an important requirement for a group to be considered a team, it is the dependence and more importantly, interdependence that forms the basis of forming and remaining part of the team. Unilateral dependence may require benevolence of the non-dependent agent while bilateral dependence or interdependence may help designers (and intelligent agents) to distribute the power and control such that every member can be truly treated as equal team members. From organisation psychology perspective, interdependence, particularly high interdependence, may result in positive effects for the team, such as better information sharing, coordination and joint decision making (Courtright et al., 2015). Therefore, interdependence is an important dimension to explore when it comes to teamwork.

This chapter demonstrated that it is useful to consider the different types of interdepen-
dence, namely task, agent and goal interdependence, and semi-formal definitions of these have been provided. Example 3.6 briefly illustrates how knowledge of the type of task interdependence can assist designers in choosing appropriate agent coordination mechanisms, identifying the potential communication requirements and how these information requirements form part of the SMM. This indicates that there is a link between interdependence and SMM, and supports the view that there is benefit in performing a fine-grained analysis of interdependence.

One important aspect that resonates in both distributed artificial intelligence literature and organisation design is the notion of predictive knowledge (Puranam et al., 2012) or taking a cognitive perspective (Castelfranchi, 1993, 1998, Castelfranchi et al., 1992). Reasoning with the mental states of the team members help agents coordinate effectively, and so does shared mental models. For example, the external descriptions in Sichman (1998) enable agents to deduce certain aspects of their teammates. Given the important role of interdependence in teams, it is natural to explore effects of the different forms on interdependence on SMM.

While it is not feasible to fully explore the different types of interdependence in a single thesis, some effort was needed to establish some link between SMM and interdependence. To achieve this, Chapter 5 presents some preliminary results of experiments that have been conducted to explore the relationship between SMM content and task interdependence. However, before moving to Chapter 5, a computational model of SMM is presented in Chapter 4. This computational model is realised in research undertaken in Chapters 5 and 6.
The primary objective of this chapter is to define and explain a computational shared mental model. A number of concepts have been defined and explained in previous chapters. Where this is the case, explicit reference to previous chapters and sections will be made. Chapters 5 and 6 present empirical studies based on the model described in this chapter.

The computational model is discussed in two parts. First, the fundamental data structures of the SMM are discussed. These data structures capture the prior knowledge of a team, as well as the team and task related information gathered during task execution. Second, a number of SMM related processes are discussed. SMM management requires a number of individual and team level processes, which enable teams to establish SMM and keep SMM updated. Potential algorithms for some of the important processes are provided. Note that these are simple illustrative algorithms and may be customised based on requirements of a domain. The chapter discusses these processes to simply highlight the purpose of these processes.

This chapter is organised as follows: Section 4.1 starts the chapter by reviewing SMM related research conducted using artificial agent teams or human-agent teams. Afterwards, Section 4.2 provides a high-level view of the SMM and the necessary components of the proposed SMM are discussed including: 1) SMM state; 2) task model; 3) interaction model; 4) member models; and 5) domain model in Sections 4.3, 4.4, 4.5, 4.6, and 4.7 respectively. Section 4.8 discusses the key SMM management processes, and Section 4.9 concludes the chapter.

4.1 Related Works

While there have been a number of studies looking at various aspects of shared mental models for artificial agent teams and human-agent teams (Fan and Yen, 2011, Goodrich and
only two proposed potential SMM (Jonker et al., 2011; Scheutz et al., 2017) with the intent of understanding the data structures and algorithms required for using SMM as a basis for designing artificial agents.

Studies such as Goodrich and Yi (2013), Nikolaidis and Shah (2013) and Koppula et al. (2016) clearly demonstrate the benefit of taking a shared mental model perspective to human-agent interaction. Nikolaidis and Shah (2013) encoded a robot teaming model using Markov Decision Process (MDP) (Ghallab et al., 2004) for a simple place-and-drill task performed by a human and a robot. In this encoding, the role of the robot was captured in the learnt policy while the role of the human was captured by the transition probabilities. They cross-trained the robot and demonstrated that the cross-training resulted in converged mental models, and this resulted in better team performance in terms of amount of human’s wait-time and the number of concurrent movements by the two parties. However, Koppula et al. (2016) showed that this simple model was unable to cope with diverse human behaviours. In particular, the model did not take into account the humans ability to act intuitively and when required, deliberatively (Koppula et al., 2016). In summary, the work of Koppula et al. (2016) demonstrated that an agent that can learn to distinguish between these two types of decision making processes, that is intuitive and deliberative, was able to collaborate more effectively with humans. Similarly, Goodrich and Yi (2013) provided their agents with a teammate mental model, team interaction model, and team task models. These models give the agent the ability to reason about the human actions and adjust their behaviour to support the human when acting as a wingman in a human-robot search task performed in a simulated 2D environment.

Other studies, such as Tambe (1997a), Fan and Yen (2011), Yen et al. (2006), and Stubbs et al. (2007), study other aspects of shared mental models. Yen et al. (2006) employed shared mental models to make proactive communication decisions. They provide their agents with shared mental models consisting of four components: team process (team plans and progress on each plan), team structure (roles, agents, and role assignments), domain knowledge (domain-dependent common knowledge), and team communication model. Based on these models, they craft a set of communication decision rules that allow the agents to make proactive communication. Their work provides a strong evidence of using shared mental models as a basis of designing artificial agents as well as key components that are effective in implementing shared mental models in artificial agent or human-agent teams. Tambe (1997a) presents a model of teamwork called STEAM (Shell for TEAMwork). STEAM
introduces the notion of team operators, which instantiates the team’s joint intentions. To apply team operators, STEAM maintains a team state, which is an agent’s abstract model of the team’s mutual beliefs. They do not explicitly categorise the mutual beliefs into categories but their primary component was joint intentions, which is a sub-component of the teamwork model. In a separate study, Stubbs et al. (2007) highlighted the implications of not properly addressing grounding (Traum and Allen, 1992). They sighted problems related to common ground (Clark, 1996, Miller et al., 2017, Stalnaker, 2002) when artificial agents do not provide information required to establish transparency and trust. Common ground is a concept closely aligned with shared mental model, with SMM literature providing more detailed account of what are the various types of content that form the common ground.

The studies discussed in the previous paragraph demonstrate the importance of shared mental models. While successful and important as a stepping-stone to building a comprehensive computational model of shared mental models, most lack the generality and the details that the construct requires (except for example, Yen et al. (2006)). In particular, these works take a selective view of shared mental model, that is, they consider only some of the SMM components that suits particular domains without justification of not including other components. For example, Nikolaidis and Shah (2013) gave the robot a model that meshed together some parts of taskwork and teamwork models. At the core, the robot was learning a taskwork model. However, a secondary outcome was learning the robot’s role in light of the human’s action, which is a sub-component of the teamwork model. Therefore, while this work was inspirational, it did not address the complete view of shared mental models. Similarly, Goodrich and Yi (2013) provided their robots with almost all key components of the shared mental models. However, this work is an application of shared mental models without any intention of providing a generic model or framework of shared mental models.

The debate here is not on including every component of shared mental models in every domain, but without addressing all of the components, the application of the construct remains incomplete. This incompleteness or variability does not help to reconcile the two threads of research. Therefore, while inspiration is taken from the above works, the works of Jonker et al. (2011) and Scheutz et al. (2017) provide a more detailed analysis and has the potential of leading to a more general computational model of SMM. Hanna and Richards (2014) and Hanna et al. (2015) are relevant in investigating multi-modal interaction, that is, verbal and non-verbal communication. They proposed a framework based on activity theory (Engestrom, 2000) and speech-act-theory (Searle, 1969). Their studies demonstrated that multi-modal interaction enhanced the human’s perception of the developed SMM and
trust in IVAs. This thesis, however, only considers explicit communication.

To extend the concepts of SMM that has been well studied for human teams (Mohammed et al., 2010) to human-agent teams, Jonker et al. (2011) proposed a mental model ontology. They view a team as a system, which consists of interacting members. Instead of using the traditional categorisation of SMM into taskwork and teamwork, they define team activities and physical components as two of their important components. A team performs team activities and has physical components, for example, team members and equipment. A team has at least two types of team activities: (1) task execution; and (2) team interaction. Task execution captures the team tasks and member(s) responsible for those tasks, while team interaction, which they see as induced by team activity requires team members to communicate. Communication is a form of interaction in their model. A team member is an agent with a mind comprising many mental models, and an agent maintains a mental model of the team. The basic components of this mental model have been described previously; these are team activities and physical components. Based on this conceptualisation, they proposed a measure that could be used to assess the similarity or the overlap of agents’ mental models. This measure is explained later in Section 5.1 on page 81 of Chapter 5.

Jonker et al. (2011) presented a mental model ontology, which helps understand key concepts required to engineer artificial agents on the basis of shared mental models. While they highlight the key concepts and their relationships, the models are still too coarse grained to actually design agents. It also requires a bit of effort to reconcile these models with the kind of constructs discussed in the human teams literature. However, this difference may simply be required because when designing artificial agents, designers have to explicitly address many cognitive aspects that are implicit in studies considering human teams. That is, in the context of human teams, these models are assessed through a number of well-known methods or tools (DeChurch and Mesmer-Magnus, 2010b). For artificial agents though, we need to consider both, the conceptual model and a measurement technique.

In a more recent work, Scheutz et al. (2017) proposed a framework for using SMM as a basis of designing human-agent teams. Scheutz et al. (2017) proposed a formal model using predicate logic. They categorise the predicates into five classes. These are:

- Agent capabilities and propensities, such as perceptions and actions;
- Moment-to-moment agent and task states, such as beliefs adopted goals and plans;
- Known and accepted obligations and norms pertaining to the task and performance domains;
Activity and equipment types; and

Functional roles of agents in teams.

For example, to capture agent capabilities, they propose using predicates such as \textit{CAPABLE}(A,X), where \(A\) is the agent and \(X\) is the task the agent is capable of executing. An agent’s goal, \(\gamma\), can be represented as \textit{GOAL}(A,\gamma). The agents are provided knowledge, in terms of a set of rules, to update the SMM based on events, such as perceptions. The rules are essentially conjunction of predicates (beliefs) of the agent that usually leads the agent to make conclusions or generate new beliefs based on what the agent already knows. Their framework also supports inclusion of human performance factors, such as workload. Such performance factors can accompany rules that allow the agent to react appropriately.

The SMM data store in the middle of in Figure 4.1.2 maintains the current information about the team and task states. The processes shown on the left of the SMM in Figure 4.1.2 are update processes and the ones on the right are control processes. The update processes allow the agent to perceive the environment and update the SMM while the control processes enable the agents to determine their behaviour. The control processes are:

- task prediction process: to gauge changes in the performance of human teammates;

- task performance and augmentation process: to enable agents to choose their behaviour to adapt in the team or to compensate for human’s degrading task performance; and

- task assignment process: that enables agents to consider allocation or re-allocation of tasks given human performance factors.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|}
\hline
\textbf{Equipment function} & \textbf{Task} & \textbf{Team Interaction} & \textbf{Team (Teammates’)} \\
\hline
Operating procedures & Task procedures & Roles/responsibilities & Knowledge \\
Likely failures & Likely contingencies & Information sources & Skills \\
Equipment/system limitations & Environmental constraints & Interaction patterns & Performance history \\
\hline
\end{tabular}
\caption{Shared mental model components prevalent in studies concerning shared mental models for human teams.}
\end{table}
Figure 4.1.2: Computational framework of SMM proposed by Scheutz et al. (2017)
Figure 4.1.3: Simplified teamwork mental model of Scheutz et al. (2017).

Scheutz et al. (2017) also proposed a simplified teamwork mental model (Figure 4.1.3).

**NOTE:** The work presented in this thesis was initiated much earlier than the model proposed by Scheutz et al. (2017). The novelty of the work presented in this thesis lies in the fact that I provide not only a computational model but also an implementation and validation of the proposed model using the experiments performed in Chapter 6. However, their model has other components, such as components to capture human performance factors, which I have not addressed in this thesis. However, given that the proposed computational model is generic, these components and their associated reasoning mechanisms can be integrated into the proposed model.

**SMM Synchronisation**

The development, maintenance, and synchronisation of SMM within the team may take different forms, for example, external to the agent’s mental state or internal. In the case of external SMM approach, agents can use a central repository, blackboard approach (Corkill, 2003) or even the environment (Weyns and Michel, 2015; Weyns et al., 2007). Alternatively, each agent can maintain a version of SMM internally (the SMM is a construct modelled as a part of mental state) and an appropriate synchronisation method can be used to maintain the consistency of the SMM, that is, ensure that the team has a single view of SMM while each agent actually maintains its version of SMM. One such method of synchronisation is explicit communication between agents.
4.2 A Computational model of SMM

The computational model of the SMM proposed in this thesis is shown in Figure 4.2.1. The view taken in this thesis is that SMM has both the data structures and the accompanying processes. The model shown in Figure 4.2.1 shows the fundamental data structures. SMM has prior information about the environment, members, tasks, and so on, and information about tasks and team established during team activities. Both types of data needs to be captured when discussing the SMM because both types of data will be used by the artificial agents during the decision making process. Each of the components shown in Figure 4.2.1 will be explained next.

The architecture of a basic agent is discussed in Chapter 2 (See Figure 2.1.1 on page 9 and accompanying discussions). Figure 4.2.2 shows a more detailed architecture. This architecture highlights the important processes required by the interpreter. The objective of this figure is to show the influence of each component of the SMM on the key processes of the artificial agent. Where the arrows are not labelled, this implies that every aspect of the component is important. In other instances, the acronyms of the particular information has been used, for example, JJ for joint intentions. The discussion on each of the key processes shown in Figure 4.2.2 will be presented after discussion of the components shown in Figure 4.2.1.

Here inspiration is taken from the work done by Dunin-Keplicz and Verbrugge (2011), who presented a formal model of teamwork concepts using multi-modal logics. While their treatment of the concepts is rigorous, this thesis takes a simplified view of their work that is sufficient to present a theoretical view of the SMM model being proposed in this thesis.

4.3 SMM State Information

The purpose of this component is to accrue task and team related information regarding the on-going activities. This component is made up of:

- Current Team: a group of agents may come together to accomplish a shared goal. This sub-component captures the agents who are part of the team activity. For example, in BW4T, this sub-component will contain information about the agents that are solving the BW4T task. In more complex domains, this sub-component may help identify new members or members who may have left the team.
Figure 4.2.1: Shared mental model data structures.

Figure 4.2.2: Artificial agent architecture with SMM components and key processes.
• Current Roles: Agents may be required to adopt different roles. The team needs to know who is responsible for which role and when these roles are filled and by which agent. For example, in BW4T, an agent may either take on the role of a search agent or a delivery agent.

• Current Task State: This captures any task related information, such as which resource is available or unavailable, which member has performed the tasks, and which tasks are on-going. For example, in BW4T, this sub-component would contain information about which blocks are present in which room, which rooms have been searched, and which blocks have been picked up.

• Joint Goals: Where possible, this sub-component captures the goals that are currently relevant. While the goal specified in the Task Model may identify all possible goals, those achieved may no longer be required during reasoning processes of the agent. For example, in the BW4T scenario, once a block is delivered, the related delivery goal is assumed to have been achieved and there is no need for the team to reason about that goal.

• Joint Intentions and Commitments: These follow from Section 2.4 and will not be discussed in detail again. For example, in BW4T, this sub-component stores information about which agent is delivering which colour from the sequence or which agent is searching a particular room. Establishing joint intentions will enable the agents to avoid duplicating effort, for example, delivering the same colour or searching the same room.

• Shared Plans: This sub-component stores the active plan of each agent, and it is assumed that meshing the sub-plans will provide a team shared plan. For example, in BW4T, this will store information about the path taken by an agent to go to a room when the goal of the agent is to search a particular room.

Agents may share beliefs, that is, have common beliefs. There are two aspects to shared beliefs. First, agents may all believe a fact without knowing whether or not others believe the fact. Second, the agents believe a fact and know others also believe the fact. The two scenarios result in two definitions of shared beliefs provided below. These definitions have been inspired by the formal models proposed by Dunin-Keplicz and Verbrugge (2011).
4.3.1 Group Beliefs

**Definition 4.3.1** (General Belief). General belief, denoted as, \( E-BEL_G(\varphi) \) means that every agent in group \( G \) has the belief that \( \varphi \). That is: \( E-BEL_G(i, \varphi) \Leftrightarrow \bigwedge_{i \in G} BEL(i, \varphi) \).

**Definition 4.3.2** (Common Belief). Common belief, denoted as, \( C-BEL_G(\varphi) \) is much stronger than general belief and requires agents to have awareness of the beliefs of other members. That is: \( C-BEL_G(\varphi) \Leftrightarrow E-BEL_G(\varphi \land C-BEL_G(\varphi)) \). This definition can be recursive, that is, everyone in \( G \) believes \( \varphi \), and that everyone in \( G \) believes that everyone in \( G \) believes \( \varphi \), and so on. **Note:** This definition is not attainable for agents not co-present; that is, not attainable using communication ([Fagin et al., 1995]). An approximation is to terminate the recursion at some depth and in this thesis, this recursion is terminated at depth 1.

4.3.2 Group Intentions

Following previous discussions, agents may or may not be aware that they have shared intentions, and this gives three definitions of shared intentions.

**Definition 4.3.3** (General Intention). General intention, denoted as, \( E-INT_G(\varphi) \) means that every agent in group \( G \) has the individual intention to achieve \( \varphi \). That is: \( E-INT_G(i, \varphi) \Leftrightarrow \bigwedge_{i \in G} INT(i, \varphi) \).

**Definition 4.3.4** (Mutual Intention). Mutual intention, denoted as, \( M-INT_G(\varphi) \) means that everyone in \( G \) intends \( \varphi \), and that everyone in \( G \) intends that everyone in \( G \) intends \( \varphi \), and so on. That is: \( M-INT_G(\varphi) \Leftrightarrow E-INT_G(\varphi \land M-INT_G(\varphi)) \).

**Definition 4.3.5** (Collective Intention). Common intention, denoted as, \( C-INT_G(\varphi) \) is much stronger than mutual intention and requires agents not only have mutual intention but everyone in \( G \) believes that everyone in \( G \) believes that they have the mutual intention. That is: \( C-INT_G(\varphi) \Leftrightarrow M-INT_G(\varphi \land M-BEL_G(M-INT_G(\varphi))) \).

The above notion of collective intention does not require the agents to make any effort in informing others if any member drops the intention. Note that the collective intention holds only when there is mutual intention and awareness within the group about this. In this regard, the joint intention theory of [Levesque et al., 1990] is much more elaborate about this. Their definition does require the agents to mutually believe termination of an intention or joint persistent goal, which can be achieved in a number of ways, including explicit communication. Section 2.4.1 provides details about their theory.
4.3.3 Collective Commitments

The definitions of collective commitments have been discussed in Section 2.4.5. In this thesis, the agents do not explicitly manipulate commitments to achieve the notion of strong commitments, and this thesis instead opts for agents using weak commitments, which I feel is sufficient to demonstrate the communication planning related approaches discussed in Chapters 5 and 6.

4.3.4 Summary

Clearly establishing common belief, collective intentions and collective commitments can be expensive and even impossible when the communication medium is unreliable. However, the existence of general beliefs, mutual intentions, and weak commitment in the right circumstances can enable the team to function properly. Without awareness there are chances of conflicts but conflicts can be avoided if sufficient number of members are aware of others beliefs and intentions. This means that mutual awareness is not always a strong requirement as long as members who have awareness are proactive in avoiding conflicts and thus ensuring successful teamwork. For this reason, weaker notions of commitment have been used in the studies presented in Chapter 5 and Chapter 6.

4.4 Task Model

Agents are systems capable of autonomous actions on the environment. Tasks, which may require one or more actions, are performed to achieve goals. Goals represent the state of the environment that the agent wants to achieve. A possible definition of a task has been given in Section 3.3, in particular in Definition 3.3.1 on page 40, which is used as the starting point for the following discussions.

A potentially generic method of handling complex and compound tasks is by adopting the Hierarchical Task Network (HTN) planning approach (Ghallab et al., 2004, Kuter et al., 2009, Meneguzzi and De Silva, 2015). Compound tasks can be represented using a task network and an appropriate decomposition method, whether domain dependent or independent, can be used to refine this network until the tasks are primitive. The following are definitions of basic HTN concepts.

Definition 4.4.1 (Task Network). A task network, $TN = (T, C)$, consists of a set of tasks, $T$, and the ordering constraints, $C$. The ordering constraints specify the particular order in which each task $t \in T$ can be executed.
Definition 4.4.2 (HTN Planning Domain). A $HTN_D = \langle A, M \rangle$, where $A$ is a set of operators, and $M$ a set of methods describing how compound tasks can be decomposed into sub-tasks.

The HTN planning problem, $HTN_P = \langle d, I, HTN_D \rangle$, where $d$ is the initial task network, and $I$ is the initial state. As explained by Meneguzzi and De Silva (2015), an efficient algorithm can be used to expand the task network.

Example 4.4.1 (BW4T Task Network). For the BW4T domain, the task network has two abstract tasks search and deliver. The sub-tasks of search task are searching individual rooms, while the sub-tasks of deliver task are delivering the individual sequence of colours. While not shown here, the deliver sub-tasks must be completed in the order of delivering the first block, then the second block, and so on. Each sub-task is essentially primitive in this example, and is associated with a goal, for example, $\text{search\_goal('roomc3')}$ for the search task Search RoomC3.

4.5 Interaction Model

The interaction model requires a number of components, such as communication channels and roles. The multiagent community provides a number of frameworks that identifies and defines each component. Examples of such frameworks are OperA (Dignum, 2004) and OJazzIC (Keogh et al., 2010). The organisational frameworks provide the designers a way to provide regulating structures to facilitate agent interaction. The frameworks provide a number of models, such as the organisational model, which describes the organisational structure by means of roles and interactions, the social model, which specifies how individual agents agree to enact roles, and the interaction model which describes the possible interaction between agents.
Note: A lighter notion of roles was actually implemented for the purpose of experiments, which involved assigning a particular role to an agent based on some capability. This approach was deemed sufficient to illustrate the relevant concepts of SMM as opposed to full organisational model as per existing frameworks mentioned above.

4.5.1 Role

A role describes the expectations of the agents’ actions and interactions. A role may require one or more capabilities. A capability indicates a skill or capacity of an agent to carry out certain tasks.

**Definition 4.5.1 (Role).** A role is a tuple $\mathcal{R} = \langle \text{name}, \text{capabilities} \rangle$, where $\text{name}$ is the role identifier and $\text{capabilities}$ is a set of capabilities required by the role.

4.5.2 Inter-agent Communication Model

This thesis relies on agents using explicit messages, that is, agents performing communicative acts or speech acts. A number of formal dialogue management systems have been proposed. Traum (2017) provides a very recent account of these approaches. The approach taken in this thesis resembles plan-based approach (Cohen and Perrault, 1979), which has its roots in Speech Act Theory (Austin, 1975, Searle, 1969). Each communication act is achieved by sending a message. A message has a particular structure and is similar to STRIPS-like encoding of planning actions.

**Definition 4.5.2 (Message Structure).** A message, $msg$, has a structure represented as a tuple $\langle \text{identifier}(msg), \text{text}(msg), \text{pre}(msg), \{\text{effects}\} \rangle$. The identifier, $\text{identifier}(msg)$, identifies each message uniquely. The actual message text, $\text{text}(msg)$ is the natural language string that gets transmitted to the receiver by the sender. The precondition, $\text{pre}(msg)$, specifies predicates that must be true for the message to be applicable. A message can have different effects based on the context in which the message is sent. Each effect, $e \in \text{effects}$ is a tuple, $\langle \text{context}(msg), \text{add}(msg), \text{del}(msg) \rangle$ where add and delete effects, $\text{add}(msg)$ and $\text{del}(msg)$, specifies the beliefs or goals that are added and removed to/from the belief/goal base following successful transmission of message $msg$, and $\text{context}(msg)$ is the context that determines the add and delete effects. The context enables the sender to test the possible mental states of the receiver in different contexts.

**Definition 4.5.3 (Message Set).** A message set, $\mathcal{M}$, is a set of message structures representing possible messages that the team can use for a given domain. An agent may use
each message structure multiple times and may employ appropriate mailbox management to track sent and received message structures.

**Definition 4.5.4** (Message Transmission). A communication action, \( a \), is a tuple \((text(a), sender(a), receiver(a))\). As explained in Definition 4.5.2, the \text{text}(msg) is the natural language string that gets transmitted to the receiver, \text{receiver}(a), by the sender, \text{sender}(a). A special constant \text{all} can be used for the receiver to trigger a message broadcast.

4.5.3 **Environment-based Communication Model**

The environment may also provide feedback to the agents. This can be achieved using centralised repositories that all members can access to get team-related information, such as, the future team goals.

**Example 4.5.1.** An example of an environment related feedback in BW4T domain is the information regarding the progress of the joint goal (colour sequence), which is to deliver the sequence of colours. The BW4T testbed used for the studies sends all agents a percept when a block is dropped in the drop zone. The agents can process this percept to identify the update to the team’s joint goal. For example, if any member delivers the first block, the testbed lets all agents know about this delivery.

4.6 **Member Models**

Each agent needs to have some indication of the skills and abilities of other members. In this thesis, the agents are provided the knowledge of their team members using two models. The first model is provided using Prolog-like knowledge rules. These rules enable an agent to determine things such as, who is a member of the team and what are their high-level capabilities. The second model is provided using the STRIPS-encoded action models. These models enable an agent to generate potential plans of itself as well as other members.

**Example 4.6.1** (Team Member Knowledge).

In the BW4T domain, an agent may need to know whether another agent is part of the team or not. To enable agents to make this distinction, each agent can be given the following rule, which states that if the name or identifier of the agent appears in the list of team
members (MemberList), then that agent is a team member.

\[ isTeamMember(Agent) \leftarrow teamMembers(MemberList) \land \\
\quad atIndex(Index, MemberList, Agent) \land \\
\quad Index \geq 0 \]

Similarly, each BW4T agent can be given information about the capabilities of the other agents, such as which colour blocks each member can pickup. Enabling different agents to pickup different blocks allows simple heterogeneous agents. The statement, \( canLift(Agent, Colours) \), is prior knowledge about which colours a BW4T agent can pickup.

\[ canLiftColour(Agent, Colour) \leftarrow canLift(Agent, Colours) \land \\
\quad member(Colour, Colours) \]

**Example 4.6.2** (Team Member Action Model).

The BW4T agents have some basic actions, such as \( goTo \) for going to room, \( goToBlock \) for going to a particular block, and \( pickUp \) for picking up a block. The following example provides a STRIPS encoding of the \( goToBlock \) action, which says that when the agent is in a room and does not have the goal to hold any block and the hand is empty, then the agent can execute this action. Note that the \( goToBlock \) action will be used in the context of a plan that requires the agent to pickup and drop a particular block. When executed successfully, the agent will be at the required block and ready to pick the block up. The actions can also have costs, which the agent accrues as a result of executing that action.

```
(:action gotoblock
  :parameters (?colorid - color ?placeid - room ?someplaceid - room)
  :precondition
    (and (in ?placeid) (block ?colorid ?placeid) (handempty) 
      (not (atblockbot)) 
      (not (goal_holding ?colorid ?someplaceid)))
  :effect
    (and (atblock ?colorid ?placeid) (atblockbot) 
      (increase (total-cost) 1))
)
```
4.6.1 Other Member Details

The SMM literature suggests that team members maintain other information about their teammates, such as team member preferences and their performance history. While these attributes are not used in this thesis, it is easy to see how these can be represented using predicates and how these integrate with high-level or action models of the agents.

4.7 Domain Model

In addition to the above data, the artificial agents may need to share information about their environment or domain. We call these collectively as domain knowledge, which can be represented using predicates.

**Example 4.7.1 (Domain Knowledge).**

In the BW4T domain, an agent needs information to identify various types of things, such as the current team goal, which is the first undelivered colour in the colour sequence. The following rule helps the agent determine the current colour that the team is required to deliver. In the rule, `sequence` represents the team’s joint goal (colour sequence), and the `sequenceIndex` predicate identifies the index within the colour sequence identifying the team’s current goal.

\[
\text{nextColourInSequence}(\text{Colour}) \leftarrow \text{sequence}(\text{Sequence}) \land \\
\text{sequenceIndex}(\text{Index}) \land \\
\text{atIndex}(\text{Index}, \text{Sequence}, \text{Color})
\]

4.8 SMM Key Processes

A number of key SMM related processes for successful functioning of a team have been identified. This thesis categorises these processes into three classes based on their function. The three categories and the purpose of each type are:

1. Initialisation: These processes provide the agents with initial data. The information required consists of basic SMM data, such as information about team members, roles, tasks and goals. In the context of designing and implementing artificial agents for this thesis, this involves providing the initial task and team models to the agent. Communication is one potential approach to consider in the context of human-agent teams for gathering this information.
2. Update and Maintenance: The primary objective of these processes is to keep the SMM information current and relevant. There are four key sub-types of processes that fall under this banner, which are:

- Perception processes: When agents are co-located, perception of other agents can provide information that can be used to update the SMM.

- Communication Processes: The agents update their beliefs and the SMM based on inter-agent communication. The assumption here is that the agents are collaborative and truthful, and so the agents can use the communicated information to update the SMM without reasoning about aspects, such as trust.

- Inference Mechanisms: Based on existing beliefs or new information observed or communicated, agents can update the SMM to remove old information and to add new information that can be inferred based on what is already known. As far as this thesis is concerned, agents are provided procedures that enable them to derive new beliefs based on existing ones.

- Synchronisation Processes: These processes can be used to maintain the consistency of the SMM. This can either be achieved automatically using underlying communication channels or by agents using explicit communication. This research, however, does not explore this aspect of SMM maintenance, and we leave this aspect for future work.

3. Utilisation: These processes leverage the SMM in the decision-making process of the agents. The two key processes that fall under this banner are:
• Action Planning: This process helps the individual agents and the team to de-
vise a shared plan. Here the agents can employ strategies discussed by various
researchers under the hood of SharedPlans (Grosz and Kraus, 1999, Kinny et al.

• Communication Planning: While this can also be considered the same as the
previous item, communication planning is treated as a separate process as far as
this project is concerned. This is not to say that communication planning cannot
be performed as part of the planning process, which is what is done in one of the
models (team-model (TM) based planning) presented in Chapter 6.

The key processes of the agent architecture shown in Figure 4.2.2 (page 63) will be
discussed next. There is no suggestion that the SMM model proposed in this thesis will
work only with the particular agent architecture but the proposed model provides some
guidance on important components of the agent that makes use of SMM.

4.8.1 Percept Handler

This process is one of most important ones and serves multiple objectives. One of the basic
functions of this process is to translate the observations and communicated messages into a
form that can be added to the belief base. Observations can be of the environment, of other
agents, and of the self after performing an action. The structure of the messages exchanged
between the agent has already been explained in Section 4.5.2.

The percept handler process can be categorised as an Initialisation processes as well
as an Update and Maintenance process. When the agent is provided prior team and task
information, the percept handler assists in initialising the SMM. During task execution, the
same process assists in keeping the SMM updated.

Example 4.8.1. The following percept rules defined in GOAL programming language (Hin-
driks, 2000) allows a BW4T agent to insert into the belief base the name of room the agent
is currently occupying, if any. The first rule inserts the name of the room when the agent
enters a room, and the second rule removes this belief as soon as the agent leaves the room.
There are other percept rules defined for other aspects. Here bel() accesses the belief base,
percept tests whether the belief base has a particular percept, in(RoomID) is the predicate
used to represent that an agent in in the room named RoomID.

```goal
if bel( percept(in(RoomID)) ) then insert( in(RoomID) ).
if bel( in(RoomID), percept(not(in(RoomID))) ) then delete(in(RoomID)).
```
4.8.2 SMM Update Handler

The SMM Update Handler is another very important process, and falls under the category of Update and Maintenance processes. It serves multiple functions. First, this process uses communicated information to update the SMM State component of the SMM. For example, when the agents converse and determine who will do what, this process updates the SMM State with information about joint intentions.

Example 4.8.2 (Joint Intentions).
Assume that an agent was asking the team members that it intends to achieve the goal, \( g \), and the team members have acknowledged that the agent can proceed, then a joint intention has been setup by the team that member \( X \) has the intention \( g \).

\[
potential\_goal(X, g) \land have\_ack(team, g) \rightarrow smm\_state.add\_intention(X, g)
\]

The other important function performed by this process is inference. Based on existing and communicated information, an agent can update the SMM by generating new beliefs.

Example 4.8.3 (Belief Inference).
Assume that two agents are solving the BW4T task. Agent, Bob visits RoomC3 and tells Mary that he visited RoomC3. Bob observes a set of blocks present in the room. Then later, Mary tells Bob that she is in RoomC3. At this stage, both Bob and Mary can infer that both now know about the blocks present in the room (assuming that no one has removed any block). Since this is now common knowledge, both can update the SMM to reflect this knowledge, for example, as follows:

\[
\text{foreach } b \text{ in RoomC3Blocks:} \\
\quad \text{insert}(\text{block}(b, \text{Mary}) \\
\quad \text{insert}(\text{block}(b, \text{Bob})
\]

4.8.3 Utilisation Processes - Planning and Communication

The following processes are examples of core decision-making processes and are categorised as Utilisation processes.
Task Selector

The agents need to have a method of choosing their tasks. An example is to use the task decomposition technique used in HTN planning (Meneguzzi and De Silva, 2015). However, for the purpose of this thesis, the agents are provided with a domain-dependent task selection method.

Example 4.8.4. For the classical BW4T domain, as previously discussed, there are two high-level tasks: search and deliver. Depending on the number of rooms in the simulation environment and the number of blocks required, these two tasks are decomposed into sub-tasks. For example, the search task has sub-tasks to search each room, and each sub-task has exactly one goal. If the agents do not know the location of the required block, they can search one of the rooms. To decide which room the agent is going to search, the agents use a simple policy of choosing the closest room(s).

Goal Reasoner

As one of the core processes of the agent’s decision-making processes, the purpose of this process is to select the best goal among the many potential goals, that is, perform deliberation. Algorithm 1 shows a simple approach taken in this thesis, although there are a number of more rigorous frameworks that could be used to improve the goal reasoning process (Cox, 2016, Johnson et al., 2016, Roberts et al., 2014, Vattam et al., 2013).

Using the task model, the `getBestGoals()` method identifies the potential goals using the task model, generates the optimal plan for each goal, and then chooses the goal with the plan having the least of these costs (line 1). Between lines 2-5, the agent repeats this process for each of the team members to get an idea of what they may be doing. Lines 8 - 13 simply checks if there are possible overlaps in terms of the goals, for example, if the agents are trying to achieve the same goal. If the agent finds that it will not interfere with another team member, it can proceed to plan for this goal. Note that these plans and goals are an approximation, which depends on accuracy and completeness of the knowledge about team members. If the agent has accurate model of the team members and the task is fully observable, then the plans and goals would be accurate. However, with less accurate models and incomplete information, these approximations would be less accurate. To tackle such instances, non-deterministic planning techniques (Ghallab et al., 2004) can be used.

NOTE: When the above algorithm does not return any goal, the agent will engage other agents to determine who does what. At this stage, the protocols used by the agents to resolve their goals, that is, determine who will be responsible for which goal, will not be
discussed. Chapters 5 and 6 have been dedicated to exploring how agents use explicit communication to determine who will be responsible for which goal.

Input:  
\( T \) - task model,  
\( I \) - mental state of self,  
\( D \) - planning domain model,  
other Agents - mental states of other agents

Output:  
goal - potential goal

1. \( goal_{self} = getBestGoals(T, I, D) \)
2. \( othersgoals = {} \)
3. foreach otherAgent ∈ otherAgents do
   4. \( goal_{otherAgent} = getBestGoals(T, otherAgent, D) \)
   5. \( othersgoals = othersgoals \cup goals_{otherAgent} \)
4. end
7. // Identify potential conflict - agents adopting same goals
8. if \( goal_{self} \cap othersgoals \neq {} \) then
   9. \( possibleGoals = goals - othersgoals \)
10. // Change goal
11. \( goal_{self} = getBestGoal(possibleGoals, I, D) \)
12. return \( goal_{self} \)

Algorithm 1: Goal selection process.

MODEL GENERATOR, PLANNER AND ACTION EXECUTOR

The purpose of Model Generator is to translate the mental state of the agent and the intention of the agent into a STRIPS representation. This allows any off-the-shelf planner to generate the plan(s) of the agent. This process is simple. Since the mental states of the agents are represented using predicates, generating the STRIPS planning instances and the domain specification is straightforward.

The Planner process is representative of the means-ends reasoning process. Any (off-the-shelf) planner that supports the input language used can be used to generate the plan(s) of the agent. The generated plan is added to the plan database maintained by the agent. Finally, the Action Executor simply takes each action and keeps executing the plan unless the action fails or there are no more actions. In both cases, the agent re-starts the deliberation process.
Input: messages - set of SMM related messages, otherAgents - mental states of other agents, goal_self - goal this agent is trying to adopt

Output: selectedMessage

1. $EU_{max} = 0.0$
2. selectedMessage = null
3. foreach msg ∈ messages do
   4. foreach ag ∈ otherAgents do
      5. plans = simulateMsgEffs(ag, msg)
      6. $EU_{msg} = score(plans)$
      7. if $EU_{msg} > EU_{max}$ then
         8. $EU_{max} = EU_{msg}$
         9. selectedMessage = msg
   10. end
4. end
5. return selectedMessage

Algorithm 2: Algorithm for selecting the world knowledge related message to communicate.

Message Selector and Message Sender

When required, each agent needs to determine the message among many others to send. Algorithm 2 shows an example of utility based selection method. The particular algorithm helps the agents to choose the beliefs that may be helpful to the other members of the team, for example, a piece of information may help the team complete the team task quicker. The basic process is as follows. Using the mental state of each agent, new optimal plans are generated for each goal. The message effects are used to generate the possible mental state of the agent assuming successful delivery of a message. Ideally, the agents want to communicate a message that results in quicker plans for the team. There is a chance that multiple candidate messages may be possible at any given time. The score function generates a value between $[0, 1]$ indicating how useful a message will be by considering the cost of the new optimal plans. Plans with a small cost accrue more utility. The message with the highest expected utility, $EU$, will eventually be selected for transmission.

4.9 Conclusion

This chapter defined and explained a computational shared mental model. This model forms the basis of the agents designed and implemented in this thesis, and the basis of empirical studies presented in Chapters 5 and 6.
In order to implement SMM for artificial agents, key data structures and processes have been proposed. The proposed model has been described in two parts. First are the five fundamental data structures. There are: 1) SMM State; 2) Domain Model; 3) Interaction Model; 4) Task Model; and 5) Member Model. These structures capture prior information and information gathered during task execution. In addition to the components discussed in human teams literature, the model proposed in this chapter has been informed by formal models of teamwork and some recent studies in the space of SMM for artificial agent teams. Data structures, such as the SMM State, have been inspired by research in formal models of teamwork.

The second part proposes a categorisation of a number of key processes that are required for the management of SMM. Three categories of processes have been identified: (1) Initialisation - to initialise SMM with prior knowledge; (2) Update and Maintenance - to keep the SMM updated; and (3) Utilisation - to use the SMM for decision-making. Examples of these processes and pseudo codes for some of the key ones have been provided as well.

The next two chapters will present empirical studies based on this model.
Agents perform tasks that range from independent tasks that do not require interactions with others to highly interdependent tasks requiring close and continuous interactions (Bradshaw et al., 2009). When faced with interdependent tasks, effective coordination and collaboration of team members become crucial. One of the key foundations of effective coordination and collaboration is having shared mental models (SMM). Agent interdependence can result due to a number of reasons. One of the reasons is due to task interdependence proposed by Saavedra et al. (1993), and has been discussed in Chapter 3.

While sharedness has been linked with better team performance, central to the notion of SMM is when and what to share. Recent work, such as Harbers et al. (2012), Manner and Gini (2013) and Wei et al. (2014) investigated this question in multiagent systems research. However, with the exception of Li et al. (2013), studies mentioned in the related work section of this chapter only consider sequentially-interdependent tasks, rather than more tightly linked team and reciprocal tasks. A recent report (Smith-Jentsch, 2015) highlights the need for studies considering other types of interdependence, notably intensive task interdependence – a type characterised as a joint action in this chapter.

The subject of this chapter is the communication content. One of the key questions answered by this chapter is:

What information, that is, SMM component, should the agents share when team members engage in interdependent tasks?

To answer the above question, a number of simulation-based experiments were conducted with artificial agent teams in which the teams performed search and rescue like scenarios. The experiments used a simplified view of SMM, mostly capturing the moment-to-moment team and task related information, that is, \textit{SMM State} (Section 4.3 on page 62).

\textbf{Note:} In the following discussions, the term SMM is used to refer to \textit{SMM State}.

The SMM comprised two components, world knowledge and intentions. These were inspired by similar research (Harbers et al., 2014, 2012). The world knowledge comprises the beliefs about the team and task, and intentions represent the current goals of the team and individual members. The other SMM components, such as team member models, task model, and interaction model, do not change during the experiments. These can also change in other settings but for the purpose of this chapter, these models are provided to agents \textit{a priori} and assumed to be static and therefore, do not form the key focus of this chapter. The primary focus is on moment-to-moment task and team related SMM information and how these impact the team performance with various levels of agent interdependence. One way teams can establish and maintain the moment-to-moment task and team related SMM information is by using explicit communication. In this chapter, this explicit communication takes the form of text-based messages exchanged between the agents. There are other approaches such as a blackboard pattern (Buschmann et al., 2007) and implicit communication. I believe that the approach taken in this chapter will generalise to other approaches.

The scenarios were generated using a Blocks World for Teams (BW4T) testbed (Johnson et al., 2009) already introduced in Chapter 3. Using the testbed, two sets of experiments were designed and executed. The first set studied the influence of sharing the two components – world knowledge and intentions – on the team performance for each form of task interdependence. The second set introduces joint actions (team task interdependence) within sequential and reciprocal tasks and studies the influence of sharing the two components on the team performance. Introduction of joint actions allows for a shift from sequential or reciprocal to team task interdependence where members execute individual actions concurrently.

The outline of the chapter is as follows: Section 5.1 discusses the approach used to measure the sharedness of SMM maintained by the artificial agent teams. Section 5.2 discusses some existing studies related to work done in this chapter. Section 5.3 describes
the tasks representing various levels of interdependence and the testbed, and provides the
details of the artificial agents that were implemented. Section 5.4 details the experimental
setup while Section 5.5 discusses the results. Section 5.6 puts these results in the context
of existing works, Section 5.7 concludes the chapter.

5.1 Measuring Shared Mental Models

While Chapter 3 has already discussed all of the key aspects of SMM, one aspect that
was not discussed was assessment or measurement of the SMM. Therefore, next a potential
approach to SMM measurement is presented. While several methods exist for measuring
SMMs for human teams [DeChurch and Mesmer-Magnus, 2010b], one for teams comprising
artificial agents is proposed by Jonker et al. (2011). Harbers et al. (2012) later extended
Jonker et al.’s similarity measure so that it could be applied to teams of agents and they
performed experiments to show that their similarity measure can be used to predict team
performance. The extended version of the measure proposed by Harbers et al. (2012) is
discussed next. In the following discussions, similarity refers to the overlap of the mental
model contents of the agents and the SMM is considered to be made of two components –
world knowledge and intentions.

Figure 5.1.1 shows an example of SMM for the BW4T domain. Assume Bot 1 and Bot
2 are two agents engaged in a joint task. Each has his/her mental model. While engaged
in their task, the agents may communicate their beliefs and goals, making their own beliefs
and goals known to others. For example, notice that each agent has her own as well as
others’ beliefs and goals, which are shown in italics. In the example, the SMM State
is composed of the components - world knowledge (beliefs) and intentions (goals).

Jonker et al. (2011) and Harbers et al. (2012) proposed a compositional measure of
sharedness. They take a black-box approach to assessing the SMM, and their method can
provide subjective as well as objective assessments. Subjective assessments would involve
each agent making assessments of what they believe the sharedness is and objective assess-
ment is usually carried out by the experimenter who would have access to the mental states
of the artificial agents. The subjective assessment is a very usual method used in exper-
iments with human teams [DeChurch and Mesmer-Magnus, 2010a], where questionnaires
and interviews are administered to the human participants. There are other methods or
techniques as discussed by DeChurch and Mesmer-Magnus (2010a).

When measuring SMM, both the content and the structure are important [DeChurch
and Mesmer-Magnus, 2010a]. The measurement method should outline: (a) elicitation
The beliefs and goals of other agents are shown in italics. An agent has certain beliefs and goals that it is not required to communicate, e.g. in(agent, room), and these may not be part of the SMM. Note that SMM is used here to only refer to the SMM State, which forms the focus of this chapter. Other components not emphasised are provided a priori and do not change as far as the experiments in this chapter goes.

Method, (b) structure representation, and (c) representation of emergence. The elicitation method refers to the technique used to determine the content or components of the model. Structure representation refers to the organised knowledge structures, that is, the degree of correspondence between how the knowledge contained in the model is represented in the mind and how the knowledge representation is modelled by the researcher. Finally, representation of emergence is concerned with how the individual-level contents and structure are collectively considered at the team-level.

With cognitive agents programmed in an appropriate agent programming language, such as GOAL [Hindriks, 2009], the elicitation method is straightforward. Such languages or the development environments provide a means to retrieve the beliefs of the agents. Figure 5.1.1 shows how the beliefs are separated into world knowledge and intentions. Therefore, Fig-
Figure 5.1.1 provides the structure of the SMM and the conceptual view of the mental model of the agents, capturing both the structure and the representation of emergence. Emergence is also represented as a floating-point value that captures the level of sharedness of the mental states. In some circumstances, it may make sense to also establish relationships among beliefs in one of the components or between components. However, this was not done for the experiments conducted in this chapter.

The definitions of Harbers et al. (2012) are reproduced with minor changes. Changes are made to the symbols and the definitions are re-worded to align the definitions with the content of the chapter. Harbers et al. (2012) view SMM as having components, which can include sub-components. For example, Figure 5.1.1 shows an SMM with two components. Examples of sub-components can be found in Section 5.3.4. The (sub)components can be queried by posing questions that all team members should be able to answer. The answers are used to compute the model agreements, which is a measure of the similarity of the answers provided by each agent for each question. Definition 5.1.1 enables assessment of each component, as shown in Example 5.1.1.

**Definition 5.1.1 (Model Agreement).** Let $A$ be the set of agents, $M_A$ be the set of all mental models, $m_a \in M_A$ be the mental model of agent $a \in A$. Also, let $Q$ be the set of all questions, and $\text{ans}(m_a, q)$ be the answer of model $m_a \in M_A$ with respect to question $q \in Q$. The agreement between models $M_A$ for questions $Q$ is:

$$ Ag(M_A, Q) = \frac{1}{|Q|} \sum_{q \in Q} \frac{|\bigcap_{m_a \in M_A} \text{ans}(m_a, q)|}{|\bigcup_{m_a \in M_A} \text{ans}(m_a, q)|} \quad (5.1) $$

If $|\bigcup_{m \in M} \text{ans}(m, q)| = 0$ then the agreement for question $q$ is 0.

**Example 5.1.1 (Computing Model Agreement).** Consider the SMM shown in Figure 5.1.1, which has two components. Now consider two question sets, $Q_{WK}$ and $Q_{INT}$ to contain the following two questions, one for each component:

1. WK Question: Who all have visited RoomB1? (Answer: Bot_1)
2. INT Question: Is anyone holding a Red? (Answer: Yes)

The model agreement for the world knowledge question is:

$$ Ag(M_A, Q_{WK}) = \frac{1}{|1|} \sum_{q \in Q_{WK}} \frac{|0|}{|2|} = 0 $$
Note that Bot_1 would respond that he has visited RoomB1 but Bot_2 would give a different answer, for example, “No one”. This gives us the value of 2 for the denominator indicating that there 2 unique answers to the question, and zero (0) for the numerator, indicating no common answers were given.

The model agreement for the intention related question is:

$$Ag(M_A, Q_{INT}) = \frac{1}{|I|} \sum_{q \in Q_{WK}} \frac{|I|}{|I|} = 1$$

The value of 1 indicates that both agents gave the same answer to the only intention related question.

Example 5.1.1 shows how individual components of the SMM can be assessed. The next step is to find a way to combine these assessments.

**Definition 5.1.2 (Compositional Sharedness).** Given a set of agents $A$, a set of mental models $M_A$ (a model for each agent), and questions $Q$ (which is combined question set if there is a separate question set per component), we say that the model $SMM$ is shared to the extent $\theta$, denoted by $Sh(M_A, A, Q, \theta)$, with respect $Q$, iff $Ag(M_A \cup \{SMM\}, Q) \geq \theta$.

The compositional measure $CS$ is:

$$CS(SMM, A, Q) = c(\{CS(cmp, A, Q) \mid cmp \in SMM\}) \quad (5.2)$$

Where $cmp$ is a component of $SMM$ and $c$ is composition function, for example: $\sum_{cmp \in SMM} w_{cmp} CS(cmp, A, Q)$. Each component and sub-component can be weighted to model the relevance of each (sub)component. The weight of each (sub)component is $w_{cmp} \in [0, 1]$ and $CS$ can be normalised to $[0, 1]$ by setting $\sum_{cmp \in SMM} w_{cmp} = 1$.

**Example 5.1.2 (Computing Compositional Sharedness).** Building on Example 5.1.1, the compositional shardness can be computed using Definition 5.1.2. Let’s assume that the two components are weighted equally, that is, $w_{wk} = w_{int} = 0.5$. Then, using a linear function as $c$, the sharedness of the $SMM$ is 50%, that is:

$$CS(SMM, A, Q) = w_{wk} \ast Ag(M_A, Q_{WK}) + w_{int} \ast Ag(M_A, Q_{INT})$$

$$= 0.5 \ast 0 + 0.5 \ast 1 = 0.5$$
5.2 SMM AND TASK INTERDEPENDENCE

Research in the context of human teams (DeChurch and Mesmer-Magnus, 2010b, Mohammed et al., 2010) indicates that compositional emergence (e.g., shared mental models), was more predictive of team behavioural process for highly interdependent than for moderately interdependent teams, which is consistent with the expectation that as the interdependence of the task increases, overlap in members’ understanding of important aspects of the task and team will enable smoother synchronisation of joint actions, and permit members to better anticipate one another’s needs (Marks et al., 2001). However, in support of Kozlowski and Ilgen’s (Kozlowski and Ilgen, 2006) prediction, compositional cognition was more predictive of team performance for moderately interdependent teams than for highly interdependent teams.

A number of studies along similar lines have been conducted using artificial agent teams or mixed teams of artificial agents and humans and a number of studies have considered some of the different forms of task interdependence (Harbers et al., 2012, Li et al., 2015, Wei et al., 2014), and some have also measured sharedness (Harbers et al., 2012, Jonker et al., 2011). Generally, higher sharedness of mental models produces better team performance. For example, Harbers et al. (2012) found higher sharedness correlated with better team performance. In their work, SMM was composed of world knowledge and intentions, which is the same view of SMM adopted in this chapter. Similarly, task interdependence has naturally been part of these studies. However, almost all involve sequentially interdependent tasks. The exception is Li et al. (2015), who introduced joint action in sequentially interdependent tasks and Wei et al. (2014), who studied tasks that were not very strongly sequential. They did this by creating subtasks that multiple agents could complete simultaneously. None of these have systematically studied the different forms of interdependence and their relationships with respect to SMM.

Mixed results have been reported for studies involving sequentially interdependent tasks in terms of which type of information or component contributes more to team performance, that is task completion times. Harbers et al. (2012) reported that when agents communicated their intentions with others, the team performance improved more than if they shared world knowledge. However, Wei et al. (2014) reported that beliefs contributed more to team performance than goals. While Wei et al. (2014) did not measure sharedness, they view the agents mental models to comprise of two components, goals (intentions) and beliefs (world knowledge). We perform further experiments involving sequentially interdependent tasks and may help explain the difference between the two studies.
In a separate study, Li et al. (2015) introduced joint action in sequentially interdependent tasks. They studied search and retrieval tasks using the BW4T testbed. In one setup, agents collaborated on a task in which some blocks were heavier, and required two agents to collect. The agents exchanged goals, beliefs, and both. Their experiments revealed that with joint actions, exchanging goals improved the team performance, measured as completion time, more than sharing beliefs only. When agents shared their goals that fulfilled the current team sub-goal with others, the other team members could start on a new task. This allowed the team to finish the team task more quickly.

These works show that in the context of artificial agent teams or mixed teams of artificial and human agents, sequentially interdependent tasks have been investigated, but other forms of task interdependence have not. This work aims to fill that gap.

5.3 Scenario: Blocks World for Teams

The experiments were conducted using the BW4T domain, which has been introduced in Chapter 3. For the simulation environment, a BW4T testbed (Johnson et al., 2009) was used. However, the basic version of the simulation environment was not sufficient to simulate all of the different forms of task interdependence. As such, the testbed was modified to be able to setup tasks with joint actions.

The BW4T testbed was modified to support design of joint tasks that would be a fair representation of the different forms of task interdependence. In the original version, only one agent could be in a room at any one time. To implement joint actions, similar to Li et al. (2015), “heavy blocks” are introduced to the domain. Heavy blocks require two agents to lift but after lifting one agent can carry it (e.g. in a trolley) to the drop zone. This means in the modified version, two agents can pick up the same block simultaneously, and therefore can be in the same room at the same time. Secondly, for team task interdependence, the blocks could be delivered in any order, that is, we removed the sequential delivery requirement. The reason for removing the sequential delivery requirement is that heavy blocks require concurrent execution, that is, represents team tasks, and sequential delivery of the blocks represents sequential tasks. Therefore, by removing the sequential delivery requirement, the task has simply one form of interdependence, which is team task interdependence.

5.3.1 Agent Interdependence and Task Design

To make the agents interdependent, the different forms task interdependence was used. Different types of tasks were designed to reflect different types of task interdependence and
test the effects of communication content on the team performance for each type of the task interdependence. Some experiments also included joint action within other forms of task interdependence, such as sequential and reciprocal, to test the effects of communication content on the team performance for each combination. Variations of two basic joint tasks (Figure 5.3.1) have been used to realise the different forms of task interdependence.

**Figure 5.3.1:** Basic joint tasks used to simulate different types of task interdependence.

Team Task: In team tasks, agents execute their actions concurrently. The joint task had some heavy blocks. The heavy blocks required one agent to help the other lift it, and afterwards one of the agents delivers it to the drop zone. The act of lifting the heavy block together is the joint action. Additionally, the agents could lift any required colour. This made the agents homogeneous. Consider the task shown in Figure 5.3.1a. In this task, agents can lift both colours. The red blocks are heavy blocks. In order to remove the underlining sequential interdependence from this task, the agents could deliver the blocks in any order, for example, the second (red) block can be delivered before the first (yellow) block. Green, pink and red are heavy blocks in Task 2.

Reciprocal Task: In a reciprocal task, each agent takes his/her turn in completing part of the task. In this task, the agents deliver a sequence of alternating colour sets in the order the colours appear in the task. Furthermore, each agent can lift colours from only one of the two distinct colour sets. This makes the agents heterogeneous in the sense that they have different capabilities. Consider the task shown in Figure 5.3.1a. For this task, one agent would be delivering yellow blocks while the other red ones. The blocks must be delivered in the order they appear. This means that agent delivering the red block now depends on the agent delivering the yellow blocks and vice-versa, making them reciprocally interdependent.

Sequential Task: In a sequential task, the first three colours are delivered by one agent while the remaining three by another agent. The blocks must be delivered in the specified order, but the second agent is free to search for its coloured blocks while the first agent is delivering.
5.3.2 Agent Behaviours

Agents were programmed in GOAL (Hindriks, 2009). The BW4T testbed provides interfaces that enable GOAL agents to interact with it. Using these interfaces, the agents can perceive specific details of the environment, such as the blocks present in rooms, and can perform actions, such as picking up a block.

Agent Actions

Each agent has a set of actions, which collectively describe what the agents are capable of doing. Specifically, the agents are capable of the following actions:

- Go to a place/room;
- Go to a block;
- Pick up a heavy or usual block;
- Put down a heavy or usual block;
- Send a message to one or multiple agents

An example of the GOAL statement of the action that agents use to pick up a heavy block is given in Example 5.3.1.

Example 5.3.1 (GOAL Action for PickUp Heavy Block). The following action precondition requires that both the helper and the requester be at the block before lifting the block. After the action, the requester will end up holding the block. The testbed maintains a unique identifier for each object, for example, a room and block. This is what the variables BlockID and RoomID mean.

```
pickUpHeavy { 
  pre { 
    me(Me), % name of self
    not(holding(_)), % not holding any block
    atBlock(BlockID), % near block and ready to pickup
    block(BlockID,_,RoomID), % can see the block
    atHeavyBlock(Helper, BlockID), % helper is here and ready
    atHeavyBlock(Me, BlockID), % I am ready as well
  }
}
```
not(Me = Helper) % I am not helping myself!
}
post {
    holding(BlockID),
    not(atHeavyBlock(Helper, BlockID)),
    not(atHeavyBlock(Me, BlockID))
}

AGENT PERCEPTIONS

Agents have the ability to perceive the environment. The common things that the agents perceive are:

- Team goal: which is the sequence of colours;
- Delivered block: all agents get a feedback from the environment when a block is delivered to the drop zone;
- Current team goal: the index of the colour that is require immediately;
- Agent location: The current location of the agent, for example whether the agent is in a room, or outside a room;
- Block: block present in the room currently occupied by the agent. The agent can remember all blocks in a room after perceiving it;
- At block: whether the agent is standing next to a block and ready to pick it up;
- Holding block: whether the agent is holding a block;
- State: whether the agent is travelling or standing;
- Map: all information regarding the environment, for example, location and names of rooms;
- Messages: the messages received by the agent; and
- Occupied Room: whether a room is occupied or not.
Some of these perceptions happen every time there is a change, for example, the blocks are perceived when the agents enter a room. There are some perceptions that occur only once, for example, the map is made available to the agents at the beginning of the simulation. The map does not change, that is, the number of rooms and their locations do not change and therefore, the agents do not need to update such information regularly.

**Decision Making Cycle**

The behaviours of the agents are determined by the main control loop programmed in GOAL. The abstract decision cycle of an agent is shown in Figure 5.3.2. The basic steps each agent takes are:

1. decide the colour to search for;
2. choose a room;
3. go to and search room;
4. if required block is found and is not heavy, pick it up;
5. if required block is found and is heavy, ask for help and wait. When help arrives, pick up the block;
6. deliver the block to the drop zone;
7. if help is requested, go to the particular room and help lift the heavy block.

The pseudocode of the control loop is given in Algorithm 3 and Algorithm 4. The algorithms have been split for easier reading. Algorithm 3 captures the key events before holding a block, and Algorithm 4 captures the key events after the agent has picked up a block and how the agent chooses a room to search if the agent does not know the location of required blocks. In GOAL, the reasoner will check the action-rules sequentially and execute the first one that is applicable. Therefore, these rules have to be organised in a certain order to generate the desired agent behaviours.

Initially, agents start searching for the first undelivered colour. However, agents use a lookahead protocol to determine which colour to deliver. In our experiments, we used a two-step lookahead. If an agent knows the location of the first undelivered colour and has
the intention of collecting it, remaining agents search for the second undelivered colour. If the one or more of the remaining agents know the location of the next required colour, they go to that room. However, only one will be able to collect the block. When the first colour is picked up, one agent collects the second colour while others start searching for the third colour. The aim of this is to ensure that sufficient time is dedicated to search. When required to lift a heavy block, an agent only asks for help when it is physically present at the heavy block. Other (helper) agents could potentially infer that help will be required soon and go to the location of the heavy block before the agent actually asks for help because the agent may tell others that it has the goal of going to the (heavy) block. However, the current agents do not perform this level of reasoning and only go to help when asked. Furthermore, if one agent asks for help, all agents that are waiting to drop a block at the drop zone or those that are currently searching for their block will go to help. If the agent knows that the colour that it is searching for, has the intention of holding or is holding is no longer required, then it will discard the colour and go on to deciding what it will do next.

Rooms are chosen randomly and the agents avoid visiting a room more than once unless the room contains multiple required blocks.

While the basic behaviours of agents are almost the same across the different forms of task interdependence, there are differences in the way agents reason about which colour to
Update belief based on perceptions and messages

if help requested to lift heavy block then
    Drop conflicting goals
    Adopt goal to pick up heavy block
else if Know location of required block and no one has intention of getting it then
    Adopt goal to pick up this block
    Inform others of the intention
else if Someone is picking up the next colour and following block is known then
    Adopt goal to pick up next block
    Inform others of the intention
else if Have goal of picking up a block then
    if Someone else picked up the block then
        Revise goals
    else if Have goal to go to room where block is located then
        Go to room
    else if Not in room where block is located then
        Adopt goal to go to room
    else if Have goal to go block and not at the block then
        Go to block
    else if Not at the block then
        Adopt goal to go to block
else if At block then
    if At heavy block and no help requested then
        Send message to request help
    else if At heavy block and help arrived then
        Pick up heavy block
    else if At usual block then
        Pick up block

Algorithm 3: The main control cycle of the GOAL agent - Part 1.

search for:

1) Sequential and reciprocal tasks: Agents choose the first undelivered colour. If another agent has the goal of holding this colour, the agent chooses the next undelivered colour.

2) Team Task: Blocks can be delivered in any order. Therefore, agents do not reason about when the block has to be delivered. Instead, agents have to determine whether the block is heavy and if so, ask for help.

While certain aspects of agent behaviours are different because of task interdependence, there are differences because of what the agents share with each other. Therefore, while the basic decision cycle shown in Figure 5.3.2 and Algorithms 3 and 4 is used by all agents,
1 if Holding block then
2   if Block no longer required then
3     Drop block
4     Revise goals
5 else if Not have goal to go to drop zone then
6     Adopt goal to go to drop zone
7     Go to drop zone
8 else if In drop zone then
9     Put down block
10 else if Have goal to go a room then
11     Go to room
12 else if Do not know location of any required blocks then
13     if Not searched a room and no one else has intention of going to the room then
14       Adopt goal to go to room
15       Inform others of intention
16 else if No other goals possible then
17     Terminate
18 else if No other goals possible then
19     Terminate
20
Algorithm 4: The main control cycle of the GOAL agent - Part 2.

there are some variations in their implementations. The implementation has been guided by what the agents actually do with the information they receive and has been described later in Section 5.3.3. Therefore, if only one component is exchanged, the agent performs reasoning described for that component only.

5.3.3 Task Model

The task model is implicit in the design of the GOAL agents. In GOAL, one way to encode the task model is by using a goal hierarchy. The plan-rules given in Algorithms 3 and 4 have many statements that require a GOAL agent to adopt different goals. These goals are derived from a goal hierarchy. The parent or abstract goals require adoption of child or atomic goals, that is, abstract goals are refined to atomic goals, and these atomic goals have plan-rules that the agents execute. The goal hierarchy for the BW4T agents is shown in Figure 5.3.3. Each goal or sub-goal has associated plan-rules which the agents execute to achieve the goals.
5.3.4 **Interaction Model, Communication and SMM**

Agents are given a set of world knowledge and intention related messages, which is what the agents communicate to develop and maintain the SMM. Agents follow a simple policy when deciding to communicate, which is that they communicate as soon as they have the required information.

As previously explained in Section 4.5 of Chapter 4, each message has a set of effects expressed as beliefs that the messages are intending to convey to the receiver. Since the interaction model is shared, both the sender and receiver know the effects of these messages. Agents exchange six sub-components, three intention related sub-components and three world knowledge related. These sub-components were selected based on prior research work (Harbers et al., 2012, Li et al., 2015) and preliminary experiments that revealed that each sub-component had the potential to improve team performance. The sub-components are communicated as messages, which are discussed next. The keyword imp stands for imperative and indicates what the agent intends to do.

The messages indicative of intentions are:

1) \( \text{imp}(\text{in}(\text{Sender, Room})) \): Sender intends to visit Room.
2) \( \text{imp}(\text{holding}(\text{Sender, Colour, Block})) \): Sender intends to collect Block of Colour.
3) \( \text{imp}(\text{nextblock}(\text{Sender, Colour, Block})) \): Sender has just dropped a block and intends to get the next block.
The messages indicative of world knowledge are:

1) `blockLoc(Sender, Block, Colour, Room)`: Sender has perceived Block of required Colour in Room.
2) `pickedUp(Sender, Colour, Block)`: Sender has picked up Block of Colour.
3) `visited(Sender, Room)`: Sender has visited Room. This message is sent irrespective of whether room contains required blocks.

5.3.5 Using Shared Mental Models

Agents employ the following policies to SMM to choose their activities such that it prevents potential conflicts with the activities of others. The following outlines how the agents use the components of the SMM. We chose a straightforward use of each intention and world knowledge, which was sufficient to test the effect of the component on the team’s performance and avoids side-effects that would have been introduced because of using more complex mechanisms. The intentions are used as follows:

1) An agent will not adopt a goal to go to a particular room if another agent has the goal of going to that room. For reciprocal task, this logic applies when both agents are delivering blocks from the same colour set, that is in a 4-agent team and not in a 2-agent team. Note that the 4-agent team is made of two teams of 2 agents each. Members of the same sub-team will not adopt a goal to go to a room if another member has the goal of going to that room.

2) An agent will not adopt a goal to hold a block that has been delivered.

3) An agent will not adopt a goal to hold a block/colour that another agent has the goal of holding — unless the block is heavy (both agents need to lift it together). For reciprocal tasks, this logic is applicable in a 4-agent team.

World knowledge is used as follows:

1) An agent will not search for a colour if this been found by another agent.
2) An agent will search for the next colour if the currently required colour has been picked up.
3) An agent will not search a room that another agent has already searched.

Agents employ the above policies to SMM to reduce interference and duplication of effort. However, the agents execute these policies as part of their own decision processes and may make decisions simultaneously. This may result in instances where the agents
may adopt similar goals, for example to look for the same colour. Like Wei et al. (2014), we simply implement a “first-come first-served” policy instead of implementing detailed negotiation mechanisms to assist agents resolve these issues.

5.4 EXPERIMENT DESIGN

Two different team compositions were designed for the purpose of experiments, that is: 1) 2-agent team and 2) 4-agent team. The 2-agent team was representative of simple teams. The 4-agent team was a 2x2-agent team, that is, 2 sub-teams of 2 agents each. This composition was required for certain tasks, such as reciprocal tasks in which we needed to have at least one agent for each of the two colour sets. The 4-agent teams also enabled testing for effects of communication content within sub-teams and between sub-teams, that is, the role of world knowledge and intentions within sub-teams and the effects of the two components when exchanged between sub-teams.

5.4.1 MEASURES

A series of simulation experiments were conducted and the following measures were collected for each run. In case of sub-teams, the number of messages and sharedness of the agents with each sub-team was also measured.

1) Completion Time: This is the total time it takes the team to complete the task, that is, deliver the required colour sequence. This measure was used as a proxy for team performance.

2) Number of messages: This measure captures the total number of messages exchanged by the agents. Also counted were the number of messages per component, that is, world knowledge and intentions. These measures are indicative of the communication cost.

3) Sharedness: This measure captures the sharedness of the agents’ mental models, that is, the overlap of the agent’s belief and goal bases. This is a compositional measure (see Section 5.1) and was calculated as the team made progress towards the team goal. Specifically, whenever a block was delivered to the drop zone, all agents log (dump) their belief and goal bases. These logs are then analysed to find the overlapping content, which is used to compute the sharedness values. The two components had a weight of 0.5 and each of the three sub-components had a weight of 0.33. In experiments where only one component was measured, the weight of the component
was set to 1, and only questions related to that component were used. In case of sub-teams, the number of messages and sharedness of the agents with each sub-team was also measured.

5.4.2 Independent Variable

The independent variable is the component of the SMM. This variable has three values (see Section 5.3.4): 1) World Knowledge (WK); 2) Intentions (INT); and 3) World Knowledge and Intentions (ALL).

5.4.3 Setup

Two different maps were used for the experiments, that is, one for each task outlined in Section 5.3.1. Variations of each task resulted in three different task interdependence types. Task 1 (Figure 5.3.1a) is labelled Map 1 and Task 2 as Map 2. The setups are as shown in Table 5.4.1. There were two sets. Set 1 had four setups (S1-S4) (both maps combined with two team compositions) for each of the three types of task interdependence giving us 12 combinations.

Set 2 had two setups, S5 and S6, representing reciprocal and sequential tasks with joint actions respectively. Here the sub-teams were reciprocally or sequentially interdependent and were required to lift heavy blocks. The experiments tested the effect of the SMM components on completion times by employing three communication strategies:

1. ALL-ALL: where agents exchanged the two components with every other agent.
2. WK-Within: where agents shared world knowledge within each sub-team but shared intentions with all agents.
3. INT-Within: where agents shared intentions within each sub-team but the world knowledge with all agents.

<table>
<thead>
<tr>
<th>Setup</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
<th>S6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team size</td>
<td>2 agents</td>
<td>4 agents</td>
<td>4 agents</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Map</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 5.4.1: Experimental setups (S1 – S6) for each type of task interdependence.
For these two setups, only a 4-agent team was used because a 2-agent team would not have enabled the testing of the effects of the two components. For example, at least 2 agents were needed in each sub-team to be able to test the effect of sharing a component within the sub-team. Combining S5 and S6 with the two types of task interdependence (sequential and reciprocal) gave us further 4 combinations, and a total of 16 combinations.

Combining each of the 12 combinations from Set 1 with the three components of the SMM (ALL, INT, WK) and the 4 combinations from Set 2 with the three communication strategies (ALL-ALL, WK-Within, INT-Within) resulted in 48 combinations in total. Each combination was run 30 times resulting in 1440 runs. Each map had 25 blocks pre-allocated to rooms and further 10 blocks were randomly generated giving a total of 35 blocks for each run. Each map had 9 rooms, 1 drop zone and 6 blocks in the joint task. Statistical significance tests were conducted using R (R Core Team, 2013). To test for normality of data, the Shapiro-Wilk test of normality was used, and the effects on the dependent variables were performed with a Welch’s t-test.

5.5 Results

This study was aimed at identifying the components that contributed most to team performance across different forms of task interdependence. Recall that going from sequential to team tasks represents increasing levels of dependence between agents as well as coordination requirements. For simplicity, we collapse the results of the two tasks (shown in Figure 5.3.1) and report the averages and standard deviations.

5.5.1 SMM Components and Team Performance

Figure 5.5.1a shows the average task completion times for the 2-agent and 4-agent teams performing different tasks. These results are for experiments resulting from setups S1-S4. Recall that a 4-agent team comprises 2 sub-teams of 2 agents each. For team tasks, the intentions contributed more to team performance than world knowledge. With a Welch’s t test, a significant effect of components was established (t(118) = 3.0086, p < 0.05, Cohen’s d=0.54) with completion time for INT (Mean: 81.56s SD: 12.17) significantly less than WK (Mean: 88.3s, SD: 12.34). In the team task, some blocks were heavy and the agents were homogeneous, meaning that the agents could pick any colour and deliver the blocks in any order. In such scenarios, knowing the intentions of team members allows agents to avoid duplicating their activities, therefore reducing interference. These results are in line with Li.
et al. (2015), who reported that with joint actions, exchanging goals results in improved completion times.

However, for sequential and reciprocal tasks, different trends have been observed between 2-agent and 4-agent teams. For sequential tasks and 2-agent team, the world knowledge contributed significantly more than intentions in terms of task completion times. With a Welch’s t test, a significant effect of components was established for the 2-agent team ($t(104) = -2.9223$, $p < 0.05$, Cohen’s $d=0.53$) with completion time for WK (Mean: 87.93s, SD: 10.08) significantly less than INT (Mean: 94.7s, SD: 14.83). In this task setting, the first agent delivered first three blocks while the remaining three by the other agent. Because agents had separate sub-tasks, exchanging world knowledge helped the other agent find it’s required blocks faster. Similar finding was made for reciprocal tasks. With a Welch’s t test, a significant effect of components was established for 2-agent team performing reciprocal tasks ($t(118) = 3.3332$, $p < 0.05$, Cohen’s $d=0.6$) with completion time for WK (Mean: 62.83s, SD: 6.88) less than INT (Mean: 67.13s, SD: 7.24).

In 4-agent teams performing sequential tasks, no significant difference in terms of completion times were noted between the two components. With a Welch’s t test, not significant effect of components was established ($t(112) = -0.22451$, $p > 0.1$, Cohen’s $d=0.04$) with completion time for WK (Mean: 61.37s, SD: 8.51) very similar to INT (Mean: 61.76s, SD: 10.86). However, moving from 2-agent to 4-agent team, the importance of intentions increases for reciprocal and team tasks. With a Welch’s t test, a significant effect of components was established for 4-agent team engaged in reciprocal tasks ($t(118) = 4.0613$, $p$
< 0.05, Cohen’s d=0.74) with completion time for INT (Mean: 42.15s SD: 7.31) less than WK (Mean: 47.46s, SD: 7.02). With a Welch’s t test, a significant effect of components was established for 4-agent team engaged in team tasks (t(117) = -2.447, p < 0.05, Cohen’s d=0.44) with completion time for INT (Mean: 56.7s SD: 9.35) less than WK (Mean: 61.1s, SD: 10.31). In these team settings, the agents within each sub-team could choose conflicting goals, for example, choosing the same block to deliver. By exchanging intentions, agents within sub-teams avoided duplicating their activities, therefore improving the completion times.

Teams of heterogeneous agents were used in the reciprocal tasks. Figure 5.5.1 indicates that the performance improved with similar communication cost when we used heterogeneous teams. This could be because the agents could complete the search tasks relevant to their colours more often while the other agents were engaged in delivering their colours. This means that the agents completing the tasks reciprocally would have generally found their colours quicker.

To make these trends clearer, we computed a measure labelled as the component influence (CI) for each task. CI expresses whether one component is more or less important than the other and is computed based on the difference between the completion times achieved when communicating both components and any one of the two components. To normalise the difference between completion times across different experiments, the tanh function is used. This function is quick to determine whether a particular component is important compared to the other or not. The CI for component c is:

\[
CI_c = \tanh(\text{CompletionTime}_{all} - \text{CompletionTime}_c)
\]

The resulting values were normalised to between 0 and 1 using \(CI_{\text{normalised}} = (CI - \min(CI))/(\max(CI) - \min(CI))\). Figure 5.5.2 for 2-agent teams show that with increasing dependence between agents, that is, going from sequential to team interdependence, the importance of intentions increases while the importance of world knowledge decreases. For 4-agent team, the intentions were almost always more important than world knowledge.

5.5.2 Communication Performance

Figure 5.5.1b shows the communication cost (average number of messages exchanged). The number of intentions exchanged was significantly lower than world knowledge overall. With a Welch’s t test, a significant effect of components was established (t(2255) = -13.912, p
Figure 5.5.2: The two graphs (2 agents and 4 agents) show that intentions become more important as the level of interdependence increases and as the number of agents in each sub-team increases. Note: lines used solely to allow the trend to be seen easily. The lines do not indicate a continuous values on x-axis.

< 0.05, Cohen’s d=0.52) with number of WK related messages (Mean: 17 SD: 14) greater than INT related messages (Mean: 11, SD: 10). This indicates that agents generally had more information to communicate about the world knowledge than their intentions. This is intuitive because there are many updates regarding the blocks present in the room.

An interesting observation is that more communication resulted in worst performance in some cases, particularly for larger teams. This is due to the two-step lookahead policy. When agents exchange information about possible blocks, in larger teams this often results in agents trying to collect the same block/colour and this increases the completion time. When agents only exchange intentions, all agents are required to find the blocks themselves, and so search randomly, thus reducing the number of unnecessary runs for the same block/colour. While it is clear that a mechanism could be designed to improve this by using a different lookahead policy, the designed policy is reasonable representation of the type of design that one would construct to solve such a task. Importantly though, this result shows that simply throwing more information towards agents can result in worse performance if significant thought is not given to how that information is used.

5.5.3 Analysis of Sharedness

Let’s start with a confirmation of what is already known in human-teams and is also intuitive; as the sharedness increases the team performance improves. Figure 5.5.3 shows the results of an experiment targeted at demonstrating this relationship between SMM and team performance. In this setup, a team of 2 agents complete the BW4T task sequentially. There were three setups: 1) Full Communication, where agents exchanged all six
components mentioned earlier; 2) Partial Communication, where agents exchanged one information related to each component; and 3) No Communication, where agents did not communicate at all. As can be seen from Figure 5.5.3, as the agents communicate more, that is, increase their sharedness of team and task, their performance improves, that is, task completion times reduce. Obviously, the communication cost (not shown here) would increase with increasing sharedness as agents would communicate more.

Sharedness was also computed in relation to each component at the time a block was delivered to the dropzone. Generally, higher sharedness correlated with improved completion times. For simplicity we show the data for team task and note that the results for sequential and reciprocal tasks are similar. For example, Figure 5.5.4 shows the sharedness at the time each correct block is dropped off for team tasks. The plotted delivery times are the time differences between block deliveries. For team tasks, exchanging intentions achieved the best completion times and the sharedness was highest for this component. Notice that in Figure 5.5.4, sharedness of intentions is highest across all six blocks and the delivery times when teams exchange intentions are fastest across most of the six blocks.
Figure 5.5.4: Sharedness and delivery times for a 2-agent team engaged in a team task.

**Sharedness and Sub Teams**

In the context of 4-agent teams, sharedness of members was measured for each sub-team for tasks solved by these teams. Overall, sharing intentions resulted in the best completion times and the sharedness of intentions was highest for reciprocal and team tasks. For sequential tasks, a significant increase was noted in the importance of intentions compared to 2-agent team. This supports the finding that higher sharedness results in better completion times. The other consistent finding is that in situations where we may have members of sub-teams potentially duplicating their efforts, sharing intentions with each other helps avoid such conflicting actions and therefore, improves the completion times.

For the more complex combination of tasks of setups S5 and S6, that is, sequential and reciprocal tasks with joint actions, similar trends were observed. Figure 5.5.5 shows the completion times for these tasks and Figure 5.5.6 shows the sharedness. The results indicate that sharing intentions within teams and world knowledge with everyone for sequential tasks achieves the best team performance but the results for reciprocal tasks were not that pronounced. Overall, these results are consistent with earlier findings that intentions and team tasks are positively correlated. Also, world knowledge and reciprocal and sequential tasks are positively correlated.
Figure 5.5.5: Completion times for reciprocal and sequential tasks with joint actions

Figure 5.5.6: Sharedness for reciprocal and sequential tasks with joint actions for the two teams (represented by the two bars)
5.6 Discussion

The intention of the chapter was to identify the components that contributed most to team performance across the different forms of task interdependence. The results show that as the interdependence increases, the importance of intentions to team performance also increases. These results are in line with Harbers et al. (2012) and Li et al. (2015) who found that when team members exchanged intentions, the team performance improves. In Harbers et al. (2012), teams were engaged in sequential tasks and their team composition was similar to our 4-agent team while in Li et al. (2015), the authors introduced joint actions in sequentially interdependent tasks.

While the results are in line with the above works, when team members can perform their sub-tasks independently, e.g. in 2-agent teams, exchanging world knowledge contributes more to team performance for sequential and reciprocal tasks. This makes sense intuitively: if other members provide potentially useful information, such as location of blocks that one is required to deliver, the team performance improves. This is a form of soft interdependence (Johnson et al., 2014) where one team/member ‘helps’ another voluntarily. This is known as backup behaviour in human teams literature. In case of 4-agent teams, intentions contributed more to team performance across all forms of task interdependence. This indicates that team composition plays a role in which component is important to team performance.

The findings that are partially consistent with Wei et al. (2014) who found that for sequential tasks, beliefs contributed more to team performance. While this is consistent with the results of the 2-agent team, a marked increase in the importance of intentions was observed when 4-agent team was concerned. These differences may hinge on other factors, such as how effectively the agents use the information that they receive. This is an area of future work.

Finally, the findings are consistent with others (e.g. Harbers et al. (2012)) in terms the role SMM plays in improving team performance. Across all tasks and both team compositions, higher sharedness of SMM resulted in improved team performance.

5.7 Conclusion

The four types of task interdependence form a hierarchy, from pooled to team, representing increasing levels of dependence between team members as well as increasing needs for coordination. Experimental results established that with increasing levels of interdependence, the importance of intentions to the team increases as well. This happens because as the
level of interdependence increases, the agents are required to work closely and in some cases concurrently. In such cases, knowing others intentions allows one to adapt their behaviour to the actions of others. When putting these findings in the context of SMM, these findings point out which sub-components of the SMM State proposed in Chapter 4 are impacted by the increasing levels of agent interdependence. Joint intentions become more important as interdependence increases while task state becomes more important for less interdependent tasks.

Team composition also played a role in determining which component contributed more to team performance. In team compositions where agents could perform their tasks independently, for example, in sequential and reciprocal tasks, world knowledge contributed more to team performance. When multiple team members may be engaged in a single sub-task, the potential of interference increases and so does the importance of knowing the intentions of others.

A factor to investigate further is the reasoning capability of the agents; that is, how the agents reason with information that they receive from others. This chapter also has not explicitly analysed the behavioural changes in the agents when agents switch from one task interdependence type to another, making this another opportunity for future investigation.

This chapter established what the agents should communicate. The next chapter also explores the communication aspect but proposes a more generic method of realising the interactions rather than pre-scripting the communication policies. In addition, while this chapter was primarily concerned with what, the next chapter is concerned with determining when and what to communicate, and also focuses on how the agents can determine the when and what using a generalisable approach.
Effective teamwork relies on coordination, which requires members’ agreement about the task and members’ expectations. This chapter is concerned with communication planning such that agents are coordinated when using the communication acts identified in the communication plan. Communication planning involves the agents determining what to communicate, when to communicate, and to whom to communicate to the different pieces of information that form the SMM. The previous chapter established that intentions become more important as interdependence increases, and with low interdependence, communicating world knowledge may also benefit the team performance. These were two sub-components part of the SMM State. This chapter is primarily concerned with how the agents determine the what, when, and to whom. The communication decisions will be restricted to the two sub-components of the SMM State, that is, joint intentions (intentions) and task state (world knowledge).

Two multiagent planning (MAP) approaches were investigated: 1) temporal-projection (TP) based; and 2) team-model (TM) based. Both approaches are distributed MAP, where the planning is performed by all agents. This approach does not require reliance on a central planning authority, which is required by centralised MAP where a single agent performs the planning for the team. In many cases, access to such a centralised planner may not be feasible. Also, complete knowledge of the agents and the world states may not be accessible to the centralised entity, making it difficult for a single entity to plan for all agents. Additionally, this chapter explores interleaved planning and execution. In this approach, planning, coordination, and execution are continually revisited and revised. This approach provides flexible approach for dealing with incomplete information and dynamic environments, and suits the interdependent tasks and domains studied in this thesis.

The source of information for both approaches is the shared mental model. The differ-
Figure 6.1: A graphical comparison of the planning and communication activities of the temporal projection and team model based planning approaches.

ence is in how the information is used and what measures are used to make the communication related decisions. The two approaches are shown graphically in Figure 6.1.

The TP based approach involves temporal projection of the future plans and the likely goals of individual agents. The temporal projection is performed for each agent separately. Using the likely goals, the agents determine if there is potential for conflicts, that is, agents may be duplicating their effort. For example, in the BW4T domain, this would mean multiple agents trying to deliver the first colour. In case of potential conflicts, the agents share their goals with other agents in an attempt to inform others of their goals and setup joint intentions. In absence of conflicting situations, the agents reason about world knowledge that could potentially improve the team performance, and the agents communicate if any world knowledge related information helps improve team performance. As shown in Figure 6.1, when using the TP based approach, communication planning happens after potential plans regarding ontic actions (actions not related to explicit communication) have been generated, that is, the approach plans for ontic and communication actions separately.

The TM based approach has been inspired by the construct of SMM. In this approach,
each agent generates a single model for the team and incorporates the communication actions as part of the team model. Intuitively, a search for a solution to such a problem will lead to the selection of what, when, and to whom to communicate during the planning for the ontic actions. Note that communication action can be treated like ontic actions, and therefore, both types of actions can be planned for together. This is unlike other approaches discussed in Section 6.1 and the TP approach, that plan for the two types of actions separately. The shortest possible team plan is used to determine what each member should do, what SMM related information, that is, intentions and world knowledge, the agents should communicate, when, and to whom. While the team plan will indicate whether to communicate intentions or not, the agents need to explicitly reason about the joint intentions and communicate to establish joint intentions when necessary (the exact criteria for determining this is explained later in Section 6.4.2). This extra effort is required because the agents generate the team plans separately and need to ensure that SMM contains the joint intentions, that is, everyone is in fact aware of the intentions of other agents. Communicating world knowledge is simply done as per the generated plan.

When programming cognitive agents using agent programming languages, such as GOAL (Hindriks, 2009), the designer usually has to provide pre-scripted interaction protocols to the agents and the plan-rules that inform the agent when, what, and to whom to communicate to. Some languages, such as Jason (Bordini et al., 2007) do have facilities to do first-principles planning as a backup behaviour, but communication has not been explored in such programming languages. Therefore, while the chapter investigates how to inject SMM into the planning process of an agent via the TP and TM approaches, the chapter also takes a small step in providing mechanisms for doing communication planning in such agent programming languages as well. This will move the focus from pre-scripted policies and provide agents autonomy in determining when, what, and to whom to communicate, and in particular, how to give the agents the ability to make these choices.

The outline of the chapter is as follows: Section 6.1 discusses existing models of communication relevant to this thesis. Before describing the two multiagent planning approaches in detail in Section 6.3 and Section 6.4, the basics of single agent planning are described in Section 6.2. Section 6.5 describes the architecture of the multiagent system that was used to perform the simulation based studies using the two domains described in Section 6.6, which also outlines the measures and the experimental setups for both domains. Section 6.7 discusses the results and Section 6.8 concludes the chapter.
6.1 Existing Models

One common method of making communication decisions is to simulate the outcome of communicating information and measure the expected utility of communicating. For example, Tambe (1997a) presented a model of teamwork called STEAM (Shell for TEAMwork). The agents communicate to establish and terminate joint intentions. The agents consider a number of factors when making communication decisions. For example, when terminating the joint intentions, the agents consider the probability that others may already be aware of the termination ($\tau$) or not ($1-\tau$). If the agents communicate the termination message and the team already knew, then the agents incur a cost, $C_{mt}$, which is the cost of miscoordination. In addition, the agents may be uncertain with probability $\delta$ that a particular event may result in the joint intention to be unachievable. However, to avoid agents erroneously communicating any termination related signal because of this uncertainty, there is cost, $C_n$, for any such nuisance. There is a cost, $C_c$, associated with each communication. The agents communicate if the expected utility of communication exceeds the expected utility of not communicating, that is, iff $EU(C) > EU(NC)$ that is, iff $\delta \cdot \tau \cdot C_{mt} > (C_c + (1-\delta)C_n)$. A similar mechanism is used when agents establish the joint intentions.

A similar approach was taken by Yen et al. (2006), who proposed the Collaborative Agents for Simulating Teamwork (CAST) teamwork model that enables agents to anticipate potential information needs among teammates and exchange information proactively. In order to determine whether a piece of information, $I$, is useful, an agent, $A$, first establishes whether: (1) $A$ believes that $B$ potentially intends to do some action (in the near future) and $I$ matches with some conjunct of the action’s precondition; or (2) $A$ believes $B$ potentially intends to do some action (in the near future) and $I$ matches with some conjunct of the action’s termination condition; or (3) $A$ believes $I$ is essential for $B$ to pursue a goal [i.e., not having information $I$ (e.g., threat information) endangers the achievement of the goal $B$ is pursuing]. Their model associated a probability with each of the three reasons, $\tau, \sigma, \delta$ respectively, and uses these to compute the probability, $p$, that agent $B$ needs $I$. In addition, their model computes the probability, $q$, which represents the probability that $B$ already knows the information. Using these probabilities, each agent then computes the expected utility of communication, $EU(PI)$, and expected utility of not communicating, $EU(NPI)$. Also, agents need to compute $U(I)$, which is the utility of communicating in the ideal case - whether agent needs $I$ but does not believe it. Agents communicate when $EU(PI) > EU(NPI)$.

Roth et al. (2006) modelled the multiagent planning problem with communication as a
Decentralised Partially Observable Markov Decision Processes (DEC-POMDP). They recast the problem as a centralised planning problem performed by a single POMDP agent by assuming no communication cost and generated a complete policy offline. At runtime, the agents decided whether to use communication. In order to decide when the agents should communicate, their agents calculate $L^t$, the distribution of possible joint beliefs of the team generated using the learnt policy and represented as a tree. Agents can calculate a joint action over the distribution of possible joint beliefs. Since the contents of this tree do not depend on any agent’s local observations, the selected joint action is identical across teammates, which is intuitive since the agents are essentially using the same joint policy.

Agents hypothesise about the joint action that would be selected by the team if it chose to communicate by pruning $L^t$ of all of the observation histories that are inconsistent with its own local observations. The agent compares the expected reward of this new joint action, $a_C$, with the expected reward of the joint action that would be chosen if it does not communicate, $a_{NC}$. If the change in expected reward is above some threshold $\epsilon$ (the cost of communication), the agent broadcasts its observation history to its teammates, who then prune their own $L^t$ to be consistent with the communicated observations. After communication, the joint beliefs of the team are synchronised. To decide what to communicate, the agents use a BuildMessage heuristic, which is a hill-climbing heuristic that greedily selects those observations that, when integrated into the joint belief, result in the highest expected reward for the selected joint action. The primary advantage of this heuristic is that it is able to reduce the amount of information required by the team to select the joint action when compared with all the information required to select the same joint action as per the learnt policy. That is, the BuildMessage heuristic selects minimum set of beliefs that are required for the team to select the joint action that maximises the expected reward, and this means that the communicating agent needs to communicate less information.

While Roth et al. (2006) presented joint beliefs using trees, Kamar et al. (2009) proposed a method based on SharedPlan (Grosz and Kraus, 1999) to present the partial and full shared plans. To present possible plans that the team could have, agents maintain a Probabilistic Recipe Tree (PRT), which represents potential recipes that could be chosen by the team. Thus PRT allows an agent to contemplate how the team may complete their partial shared plans. Using PRT, agents reason about the goals and plans of their teams, as opposed to tracking the detailed beliefs of the individual members by considering the joint actions over time. Based on PRT, their agents perform a utility-based evaluation to determine whether informing others something or asking someone about something results in increased utility. If the utility exceeds cost of communication, agents communicate. Amir
et al. (2014) extended this work by casting the problem as a single agent MDP and learning the communication policy offline. The approach allowed agents to consider communicating at a latter stage in their plans rather than immediately. Amir et al. (2014) show that their approach improved the reward accrued by the team.

A stream of other works considering multiagent planning with communication are based on POMDP or DEC-POMDP models. For example, Wu et al. (2011) proposed the Multi-Agent Online Planning with Communication (MAOP-COMM) algorithm. Their agents maintain a belief pool, $B$, which consists of individual agent histories, $H$, (agent actions and observations), their probabilities and resulting joint belief states. At each time step, each agent approximates a one-step lookahead of joint policy, $\bar{\pi} = (\pi_0, ..., \pi_n)$, using the joint history, $\bar{h}^t = (h_0, ..., h_n)$ for each agent $i \in I$. The agents use their belief pools to generate their own actions and predict the actions of others, that is, the agents use the belief pools to generate joint actions, $\bar{\pi}(\bar{h}) = (\pi_0(h_0), ..., \pi_n(h_n))$. The objective of each agent is to find the joint policy, $\bar{\pi}$, that maximises the following value function:

$$ V(\bar{\pi}) = \sum_{h \in H} p(\bar{h}) \sum_{\pi} \prod_{i \in I} \pi_i(h_i) Q(b_{\bar{h}}, \bar{\pi}(\bar{h})) $$

where $p(\bar{h})$ is the probability of the history, $\bar{h}$; $b_{\bar{h}}$ is the joint belief associated with $\bar{h}$; and $Q(b_{\bar{h}}, \bar{\pi}(\bar{h}))$ is the value of policy at $b_{\bar{h}}$, which the agents receive in the future.

Agents using MAOP-COMM communicate when they detect any inconsistency in the belief pools, which may get the agents to communicate information even if it is not relevant to others. This inconsistency is detected using only the local observations of the agent. Intuitively, if the belief pools are same, the joint actions chosen by each agent will be same. A complete or exact solution requires full enumeration of possible joint actions and belief states, which can be expensive. To tackle the computational complexity, the authors implemented a one-step lookahead to estimate the value of future steps instead. However, the success of the approach hinges on a number of factors. Firstly, after the initial best policy is generated by all agents using the same random seed, the future best policy is anticipated to be mostly unique. In case this does not happen, the designer needs to specify a default tie-breaking rule. In cases with many equally valued best policies, for example in the BW4T and similar domains studied in this thesis, the designer cannot be expected to specify tie-breaking rules for every scenario. Secondly, since the consistency of the belief pool is crucial to coordination, the authors suggested that the local observations may help provide hints as to when there are inconsistent belief pools. However, local observations have to provide some indication of what others are doing. Therefore, relying on syncing
the model only after inconsistency is detected using local observations that do not provide useful information about others may leave the team in situations with severely decreased performance. Therefore, while MAOP-COMM does guarantee coordination if the belief pools are same, the chances of mis-coordination are increased greatly when observations do not provide any useful information about others.

Another approach building on the work of Wu et al. (2011) was taken by Gasparini et al. (2016). The difference was the trigger used by the agents to communicate. Wu et al. (2011) used the belief pool inconsistency computed using the local observations while Gasparini et al. (2016) used the distance between the belief nodes of the agents. In addition to the joint histories, their agents maintain a local belief state of each agent for each history. Each belief node, $n^k = (\vec{h}^k, b^0_k, b^1_k, ..., b^k_n, p^k)$, where $\vec{h}^k$ is the joint history with joint belief state, $b^0_k, b^i_k \ i \in I$ is the belief of each agent, $i$, and $p^k$ is the probability of the node. The nodes are clustered and used as the basis of determining when to communicate.

Unhelkar and Shah (2016) proposed a decentralised communication policy, ConTaCT, that enabled agents to decide whether or not to communicate. Their approach is similar to Wu et al. (2011) and Gasparini et al. (2016). However, in their model, the agents may not know the correct or all aspects of the transition matrix, making it difficult for an agent to determine whether a state is reachable or not. The agents get the incomplete or missing information via observations, and then have to decide whether to let other agents know about the new observations. During the selection of future joint actions, it is assumed that the agents will choose policies that increase their individual rewards. After generating potential future joint actions and states, and the agents use a rule-based mechanism to determine when to communicate. Their agents contemplate three scenarios. In first scenario, $\alpha$, the agent computes the total reward if the observation, $b^0$, is ignored completely. In second scenario, $\beta$, the agent computes the total reward using the observation locally but not telling others. In the final scenario, $\gamma$, the agent computes the total reward if the observation is used by everyone. The agent then chooses the scenario that would result in the highest possible team reward. When communicating, the agents communicate all of their beliefs to others. Their approach in terms of communication planning is similar to Tambe (1997a) and Yen et al. (2006) with one difference. In their model, the agents can make decisions about when to incorporate new observations into the mental models of self and others. However, communicating everything will certainly be impractical in many scenarios.
There are two approaches evident in the works discussed above: state-based (for example, Wu et al. (2011)) and plan-based (for example, Kamar et al. (2009)). The state-based approach considers future joint actions and the joint belief states, and selects joint actions that maximise the team reward. The plan-based approach takes a higher level view and considers the joint plans or joint goals. The plan-based approach does require some estimation of others beliefs during plan generation without enumerating every possible joint action combination and evaluation of these.

When it comes to communication planning, all of these models use a utility-based evaluation strategy when deciding whether the agents should communicate or what to communicate. The key issue with the models proposed by Tambe (1997a) and Yen et al. (2006) is estimating the various probabilities. In their works, the designers provided these or rules for agents to generate these. The model proposed by Roth et al. (2006) is centralised and complete policy is learnt offline. The work in this thesis is concerned with distributed plan generation in an online setting. The plan-based approach proposed by Kamar et al. (2009) is very promising and closely resembles the approach taken in this thesis, at least in terms of considering the potential plans of the agents.

The models proposed by Wu et al. (2011), Gasparini et al. (2016), and Unhelkar and Shah (2016) inspire the modelling approach. However, the primary focus of these works is when to communicate, that is, whether to communicate or not. Unlike the works by Roth et al. (2006) and Kamar et al. (2009), they do not address what to communicate. As already demonstrated in Chapter 5, different types of information are required for different levels of interdependence.

6.2 Single Agent Planning (SAP)

Single agent planning is the process for generating a sequence of atomic actions that an agent can execute to take the agent from some initial state to a goal state. The following formal model is based on classical planning (Ghallab et al., 2004), which assumes that the world is composed of finite set of states, is static, deterministic and fully observable, and actions are executed instantaneously, that is, the actions have no duration. While these assumptions are strong, it helps formulate the problem and understand the models.

**Definition 6.2.1** (Single Agent Planning Problem and Solution). A single agent planning problem, \( P \), is a tuple \( \langle I, G, A \rangle \), where \( I \) is the initial state or beliefs of the agent (Definition 2.1.3), \( G \) is the goal state the agent wants to achieve and \( A \) is the set of actions available
to the agent (Definition 2.1.5). The solution to such a problem is plan (as previously defined in Definition 2.1.6).

6.2.1 Planning Problem Representation

A widely used language for representing the single agent problem is the Planning Domain Definition Language or PDDL (McDermott et al., 1998), which is an extension of STRIPS (Fikes and Nilsson, 1971). Modelling a single agent planning problem using PDDL requires providing specifications for the domain and a problem. The domain describes the types of objects, the predicates that can be used to describe the world states, the actions or operators that can be used to solve the problem. The problem is grounded set of predicates describing the initial state and the goals to be achieved. Example 6.2.1 and Example 6.2.2 respectively, provide partial PDDL BW4T domain and a sample problem specification.

Example 6.2.1 (BW4T Domain). The types of objects for the BW4T domain would be types such as blocks and rooms.

```
(:types colour room slot place)
```

The predicates describe things such as, the map, the blocks, and which block the agent is holding.

```
(:predicates
  (connected ?r1 - room ?r2 - place)
  (connected ?r1 - place ?r2 - room)
  (connected ?r1 - place ?r2 - place)
  (block ?colorid - color ?placeid - room)
  (in ?roomid - room)
  (holding ?colorid - color ?placeid - room)
  ...
)
```

The actions describe what the agent can do. In the following, a room is distinguished from other spaces, such as corridors. In the following action, the agent is going from a room to a corridor.
Example 6.2.2 (BW4T Problem). The particular objects that are part of the planning problem are defined first.

(:objects
   red blue green white yellow pink orange - colour
   rooma1 rooma2 rooma3 - room
   ...
)

Then, the initial state or the current state of the agent is described. For example, the following snippet describes the agent believes is the state of the world.

(:init
   ...
   (block white rooma1)
   (block orange rooma2)
   (block blue rooma3)
   (handempty)
   ...
)

Finally, for this sample problem, the agent is attempting to find a plan to deliver the
first two blocks. The sequence has been divided into slots, numbered starting from s1. Each slot is has a colour representing the colour of the block that is required.

\[
(:\text{goal} \ (\text{and} \ (\text{slotcolour} \ s1 \ \text{blue}) \ (\text{slotcolour} \ s2 \ \text{orange})) )
\]

### 6.2.2 Single Agent Planning Algorithms

A number of domain-independent planning algorithms have been proposed (Ghallab et al., 2004), such as state-based, partial order planning, planning graphs, and hierarchical task network planning.

The state-based search algorithms find a sequence of actions such that when executed, the plan or sequence of actions will transition through from initial to the goal state. Using a tree representation, where each node represents a state, a blind search will require enumeration and evaluation of every possible state to identify potential paths (sequence of actions) from the root node (initial state) to the leaf node (goal state). When actions have a cost, these potential plans can be evaluated based on action costs, and an optimal plan is then a plan that costs the least. The process of enumerating every possible state can be simply too expensive for domains with large number of variables, both in terms of space and time. Therefore, a number of domain-independent heuristics (Ghallab et al., 2004) have been proposed to guide the search, that is, node selection. Among the most effective ones is the delete-relaxation heuristic, which simply requires the delete effects be removed from action specifications. A number of state-based planners have been proposed. The one used in this thesis is the Fast Downward planner (Helmert, 2006). The Fast Downward planner performs best-first multi-heuristic search. It uses $h_{FF}$, a heuristic based on the concept of relaxed planning graph, and $h_{CG}$, a heuristic calculated using the causal graph.

### 6.3 Multi-Agent Planning (MAP)

In MAP, the planning task is distributed among multiple agents. This requires the agents to work together to generate a joint plan that can achieve the joint goals. When working in a team, generally, the agents are anticipated to be collaborative rather than competitive. However, the reasons for why each agent joins the team may be different, but this aspect is not emphasised in the domains used in this chapter. A number of approaches to solving a MAP task exists.
Centralised MAP The MAP problem can be cast as a single-agent planning problem, and a single agent can generate a team plan to be followed by all team members. This is referred to as centralised MAP. The solution requires two steps. First, finding a plan that achieves the goals. Second, finding a way to coordinate the agents, that is, identifying individual agent plans that when executed, will achieve the team goals.

Distributed MAP In contrast, in distributed MAP, the agents plan individually. Distribute MAP has a number of challenges. The agents need to know things such as what tasks or goals they are responsible for, how to plan for these tasks, and how to interact with other agents. Answering these questions has given rise to a number of coordination approaches, which are (Durfee 2001):

- Pre-planning Coordination: This requires the tasks or goals to be carefully distributed to individual agents. The agents would then be required to generate individual plans and when executed these plans will achieve the team goal. The agents may also plan sequentially, thus avoiding any conflicts or task duplication. In the context of BW4T, this can be done by dividing the colour sequence among the agents. For example, one agent delivers the first three colours and the other delivers the last three if the total number of colours required is six and the colours are unique. This type of goal allocation to the agents may, however, result in sub-optimal performance in the classical BW4T task.

- Post-planning Coordination: When using this approach, the agents first plan individually and then work with other agents to devise a team plan. This process is referred to as plan merging. The agents may monitor and re-plan when one or more agents fail to complete their plans. In the context of BW4T, this would involve each agent planning for the complete colour sequence and then choosing a team plan such that when each agent delivers their colour, the task is completed quickly.

- Interleaved planning and coordination: In this approach, the planning, coordination, and execution are continually revisited and revised. This implies that the team may not have a full shared plan at the beginning, and in certain cases only have a full shared plan while well engaged in the team activities. This approach works very well for environments that are dynamic, in situations where agents have incomplete information and cannot solve the entire task from the onset. But this approach and its techniques do provide means for the team to achieve the team goals even with
these uncertainties. This is well suited for domains like BW4T. Here the agents do not know the location of the blocks initially and therefore have to decide the rooms to search. Even deciding which rooms to search, the agents do not have to generate a complete plan to search all rooms. The agents continually revisit and revise their search and delivery joint plans to successfully deliver the required colours as soon as possible.

As stated earlier, in this thesis, the agents use interleaved planning and execution and distributed MAP.

6.3.1 MAP model

The MAP model in this thesis takes on STRIPS-like representation and is described in PDDL. The model takes into consideration the beliefs and goals of self as well as other agents, that is, each agent maintains a mental state of each team member. These nested mental states represent what an agent believes about the other agents, and the nesting can be infinite. For reasons already stated in Definition 4.3.2 (page 65), this recursion was terminated at level 1.

The MAP model can be defined as:

- $I$ is a finite set of agents, indexed from $1, ..., n$.
- $\tilde{B}$ is a finite set of possible system belief states. $B_i$ is the possible belief states of agent $i \in I$.
- $G$ is a finite set of team goals, indexed from $1, ..., n$.
- $A$ is a finite set of actions available to the agents. $A_i$ is used to represent the actions of agent $i$.
- $O$ is a finite set of possible observations that are available to all agents. $O_i$ is the observations of agent $i$.
- $\pi$ is the joint plan of the team. $\pi_i$ is the plan of the individual agent. Moreover, $\pi_i^g$ is the plan of agent $i \in I$ for goal $g \in G$.
- $b^0$ is the initial belief state, where $b^0 = \bigcup_{i \in I} b_i^0$. $b_i^0$ is the initial belief state of each agent.
6.4 Multi-Agent Planning Algorithms

This chapter investigates two different strategies for joint communication and ontic action planning: 1) temporal-projection (TP) based; and 2) team-model (TM) based. The following describes each approach in turn. Figure 6.1 (on page 108) earlier presented a comparison between the two approaches graphically.

The algorithms associated with both approaches show how the sub-processes of the Plan Manager and the Communication Manager of the computational SMM and agent architecture shown in Figure 4.2.2 (page 63) have been implemented and how those processes assist in updating the SMM State. For the purpose of the experiments in this chapter, we assumed that the other components of the SMM, such as the Task Model, remain unchanged. In other domains, these can also change, and the approach taken in this thesis will generalise to those domains. Notice also, that SMM forms one of the key sources of information to both planning approaches (for example, in addition to nested beliefs of other agents).

6.4.1 Temporal Projection Based Planning Algorithm

This section first explains the key steps involved in performing TP before discussing the algorithm in Algorithm 5. To start the process, the agents identify the most likely goals, \( G_{\text{opt}}^i \), of each agent, \( i \in I \), individually using Equation 6.2. This is done by first identifying the potential team goals of every agent. The shared task model provides the agents with the ability to identify potential team goals. It is assumed that the agents are also given knowledge of how to process the task model and select applicable tasks for a given context, and this information is also part of the SMM. Next, the probability of each goal is computed using Equation 6.3, where \( \alpha \) is a normalising constant with value \( \frac{1}{\sum_{g \in G} \text{cost}(g)} \). An optimal plan is generated for each goal. A cost function assigns each goal a cost based on the cost of the optimal plan, and these costs are the basis for choosing the most likely goals of each agent.

\[
G_{\text{opt}}^i = \arg \min_{g \in G} p(g) \quad \forall i \in I \tag{6.2}
\]

\[
p(g) = \alpha \times \frac{1}{\text{cost}(g)} \tag{6.3}
\]

Using Equation 6.2, each agent gets an idea of what other members may be doing. However, to proceed, the agent needs to determine what are the chances of any potential conflicting situations leading to task duplication when this is not intended. In the context of
BW4T, examples of conflicts include collecting the same colour or searching the same room. These are regarded as conflicts because task duplication results in teams taking longer to complete the tasks. To avoid such conflicts, a simple policy is to resolve any uncertainty, that is, to ensure that the probability of conflicting situations is zero. Alternatively, the agents could take *risks* by accepting some uncertainty. For the purposes of this thesis, a simple policy of taking no risks was taken, and then the uncertainty can be estimated by taking an intersection of the set of potential goals, as shown in Equation 6.4. The no-risk policy would be to ensure that the set $\zeta$ is empty. This means that the agents can continue working independently as long as they do not face situations where they may be trying to achieve the same goal unintentionally. Working independently does not imply that the agents are not interdependent. It simply implies that the agents can reason about their teammates and are confident that they can complete their part without intentionally interfering in others’ activities.

$$\zeta = \bigcap_{i \in I} G^i_{opt} \quad (6.4)$$

**Figure 6.4.1:** Example BW4T problem.
Example 6.4.1 (BW4T - conflicting search goals).

Consider a 2-agent team solving the BW4T task shown in Figure 6.4.1. The agents have to search the rooms initially to find out the blocks located in each room. Figure 6.4.2 shows part of the TP process. The planning agent, self, identifies the potential goals of the other member, and then generates an optimal plan for each goal. The best goals of each agent are those that have plans with least cost. In BW4T testbed, agents traverse in the environment via way-points. The plan cost represents the number of way points traversed. Let’s assume that both agents start from the FrontDropZone. Notice that there are two closest rooms, RoomC1 and RoomC3, in terms of the number of way points to traverse. Both rooms require the agents to traverse 5 way points. When considering the probability of each agent choosing between nine different rooms, these two rooms have the highest probability. Therefore, the agent determines that the best goals of self and other involves searching these two rooms. In this example, the set $\zeta = \{\text{RoomC1, RoomC3}\}$.

![Table showing the goals, optimal plan cost, and probability for both agents.]

<table>
<thead>
<tr>
<th>Goal</th>
<th>Optimal Plan Cost</th>
<th>p(g)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RoomC1</td>
<td>5</td>
<td>0.14</td>
</tr>
<tr>
<td>RoomC3</td>
<td>5</td>
<td>0.14</td>
</tr>
<tr>
<td>RoomC2</td>
<td>6</td>
<td>0.11</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Most likely goals: RoomC1, RoomC3

<table>
<thead>
<tr>
<th>Goal</th>
<th>Optimal Plan Cost</th>
<th>p(g)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RoomC1</td>
<td>5</td>
<td>0.14</td>
</tr>
<tr>
<td>RoomC3</td>
<td>5</td>
<td>0.14</td>
</tr>
<tr>
<td>RoomC2</td>
<td>6</td>
<td>0.11</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Most likely goals: RoomC1, RoomC3

(1) Consider potential goals and probability of achieving each goal

(2) Select best goals

Figure 6.4.2: Example of temporal projection used to identify the best search goals of two agents in a sample BW4T task.

There are two scenarios that can result due to $\zeta$: 1) $\zeta$ is empty; and 2) $\zeta$ has at least one element. When an agent determines that there are no chances of conflicting
situations, the agent continues with the planned activities. However, as Johnson et al. (2014) pointed out in their work, agents have both hard and soft interdependence between them. Soft interdependence is also demonstrated in human teams when team members take on supportive or backup behaviours. One way agents can realise this behaviour is by identifying information such that the communication of the information improves the team performance. An algorithm to identify useful information has already been presented in Section 4.8 (page 77).

Input: \(\alpha_{prev}\) - previous action, \(I\) - current state, \(A\) - domain model, \(otherAgents\) - mental states of other agents, \(\pi\) - current plan

Output: \(\alpha\) - next action

1 if \(\text{len}(\pi) > 0 \&\& \text{!failed}(\alpha_{prev})\) then
2 \(\alpha = \text{next\_action}(\pi)\)
3 return \(\alpha\)
4 else
5 \(G_{opt}^i = \arg\min_{g \in G} p(g) \ \forall i \in I\)
6 goal\_self = \text{get\_best\_goal}(G_{opt\_self})
7 \(\zeta = \bigcap_{i \in I} G_{opt}^i\)
8 if \(\zeta \neq \emptyset\) then
9 \(messages = \text{get\_Messages}(goal\_self)\)
10 \(\zeta = \text{check\_Int\_msgs}(messages, otherAgents, goal\_self)\)
11 else if \(\zeta = \emptyset\) then
12 \(messages = \text{get\_Messages}(goal\_self)\)
13 \(checkWK\_Messages(messages)\)
14 plan = \text{get\_Plan}(goal\_self, I, A)\)
15 \(\alpha = \text{next\_action}(\pi)\)
16 return \(\alpha\)

Algorithm 5: Single agent planning algorithm.

TP Planning Algorithm

Algorithm 5 shows the agent’s core decision making algorithm. If an agent has a plan, it tries to achieve it as long as possible (lines 1-3). If the agent detects a failure in performing an action, then it re-plans. Each time the agent plans, the agent has to determine possible goals. This is done using the algorithm explained earlier in Algorithm 1 of Chapter 4. Briefly, an agent determines own and others’ potential goals, and using these computes the set \(\zeta\) (6). If the agent finds that there is a possibility of other agents also choosing the same
goal (7), the agent communicates the selected goal to other members (8-10), which allows the team to update the SMM State. Algorithm 6, which is explained later, shows how the agent makes these communication decisions. If the agent finds that no other agents may be choosing the same goal, that is, there is no potential conflict (11), the agent proceeds to reason about any possible world knowledge worth communicating to other agents (12-13) in order to update the task state of the SMM State. The agent then generates a plan for the selected goal and returns the first action in the plan for execution (14-16).

TP Communication Algorithm

Algorithm 6 is used by the planning agent to determine whether to communicate information regarding the goal. The interaction model introduced in Chapter 4 is used by the agents to determine the set of messages. In order to determine whether communicating the planning agent’s goal will be helpful to other agents, the planning agent goes over each message related to the selected goal (1-3). In BW4T domain, an example of a message conveying a delivery goal will be: “I am going to get a Red block from RoomC3”. If the planning agent identifies that, according to the local beliefs of the other held by the planning agent, the other agent may not already believe his/her goal (4), then the message conveying the goal to the other agent is selected for transmission (5). Note that when the same message needs to be sent to all members of the team, the agent can broadcast the message to all agents instead of sending out individual messages. This can be achieved by the mailbox management process and will not be discussed in this thesis.

Input:
- goal_self - goal this agent is trying to adopt,
- messages - set of messages related to goal_self,
- otherAgents - mental states of other agents.

Output:
- selectedMessages - list of messages to be transmitted.

1. selectedMessages = null
2. foreach msg ∈ messages do
3.     foreach ag ∈ otherAgents do
4.         if goal_self ∉ get_potential_goals(ag) then
5.             selectedMessages.add((ag, msg))
6.         end
7.     end
8. return selectedMessages

Algorithm 6: Algorithm to select messages that conveys the planning agent’s goal to other agents.
Algorithm 7 allows the agents to choose world knowledge related information, that is, Task State information, that the agent believes may help others and the team to complete the team tasks quicker. There could be a number of world knowledge related messages, for example, in BW4T, this may be information regarding location of blocks and rooms that have been searched. For each message, the planning agent generates new optimal plans for each possible goal of the other agents by using the perceived new mental state of other agents resulting from communicating the message. The agent uses a function to compute the utility of each plan. A simple utility function would be based on the cost of the plan, for example, $EU(\pi) = \frac{1}{\text{cost}(\pi)}$, where shorter plans implies lower cost (assuming uniform action costs) and receives higher utilities. The message with the highest expected utility, $EU$, will be selected for transmission.

**Input:** messages - set of world knowledge messages, otherAgents - mental states of other agents

**Output:** selectedMessage

1. $EU_{\text{max}} = 0.0$
2. selectedMessage = null
3. foreach $msg \in$ messages do
   4. foreach $ag \in$ otherAgents do
      5. plans = simulateMsgEffs($ag, msg$)
      6. $EU_{msg} = \text{score}(\text{plans})$
      7. if $EU_{msg} > EU_{\text{max}}$ then
         8. $EU_{\text{max}} = EU_{msg}$
         9. selectedMessage = $msg$
      end
   end
4. end
5. return selectedMessage

**Algorithm 7:** Algorithm for choose world knowledge related message to communicate.

**Summary**

The TP based agent, plans for each agent before determining the best course of action. In doing so, if the agent finds a conflicting situation, the agent communicates intentions to resolve the situation. Note that the complete process of negotiation has not been discussed. The communication is simply seen as a trigger point for the complete negotiation between the agents. In absence of conflicting situations, each agent considers communicating if it
improves the team performance. The TP process thus considers communication planning as an after-planning activity, that is, plans for ontic and communication actions separately. The model proposed next will consider communication within the core planning loop, that is, plan for communication actions together with planning for ontic actions of the agent.

6.4.2 Team Model-Based Planning Model

When using TM based planning, each agent maintains a team mental model based on the shared mental model, and has a different representation to that used for single agent planning domain and problem. One way to achieve this representation is explained below.

MAP representation

The MAP problem is represented using PDDL. Although the agents perform distributed planning, it is possible for each agent to take into consideration the beliefs and goals of other agents, and plan as if they were a single entity. This is closely aligned with the notion of shared mental model. To provide the agents with this capability, the following PDDL representations of the domain and problem are used. These are extensions of the problems and domains provided for the single agent planning problem discussed earlier. In order to build the MAP model, each agent needs a slightly different representation to that of single agent planning representation. To start with, objects of type agent allows for addition of agents to the representation. For example,

(:types colour room slot count place agent)

To capture the beliefs and goals that the planning agent maintains about the other agents, that is, the nested beliefs, one simple approach to use is to add a parameter to the affected predicates. In the following BW4T example, each agent can now identify which agent believes the blocks or which agent has the goal of collecting a block from a particular room. The epistemic planning literature (Bolander and Andersen, 2011, Muise et al., 2015) provides guidance on how to capture nested beliefs in general. However, for the kind of tasks used to demonstrate the TM based planning approach, the method that is used in this chapter is sufficient.

(block ?colourid - colour ?placeid - room ?a - agent)
The actions can leverage this multiagent representation to assist the planner in choosing an action for each agent. Since a classical planner is used in this thesis to synthesise a plan for the MAP problem, there are two approaches that can be used to generate a multiagent plan. The first option is to use a sequential turn-taking approach. This is demonstrated in the following example. The precondition, \((\text{curr} \ ?a) \ (\text{next} \ ?a \ ?a2)\), establishes the current agent that is being planned for and the next agent, and following this, the effect, \((\text{not} \ (\text{curr} \ ?a)) \ (\text{curr} \ ?a2)\), changes the current agent. The second option is to remove this sequential requirement and plan for one agent treating the other as a potential resource that can be used. Therefore, other agents will only be involved in a plan if their involvement improves the plan. For the purpose of this thesis, option 1 has been implemented.

The following is an example of a BW4T action. It gets an agent to decide whether to select a block for delivery. To make this decision, the agent has to be in the room where the block is, the block must be visible, the agent should not be currently holding a block (assuming that an agent can hold only one block), the agent should not have an intention of collecting another block, and another team member should not have the intention of collecting the same colour from potentially another room.

```prolog
(:action gotoblock
  :parameters (?colorid - color ?placeid - room
              ?someplaceid - room ?another - room
              ?a - agent ?a2 - agent)
  :precondition (and
                 (in ?placeid ?a) (block ?colorid ?placeid ?a)
                 (handempty ?a) (not (atblockbot ?a))
                 (not (goal_holding ?colorid ?someplaceid ?a))
                 (not (goal_holding ?colorid ?another ?a2))
                 (curr ?a) (next ?a ?a2))
  :effect (and (atblock ?colorid ?placeid ?a) (atblockbot ?a)
             (not (curr ?a)) (curr ?a2)
             (increase (total-cost) 1)))
```

127
When an agent generates the problem instance, the agent includes the nested beliefs of the other agents. For example as follows, which captures the various beliefs and goals of the two agents solving the BW4T task:

```plaintext
!(:objects
   ...
   bot_1 bot_2 - agent
   ...
 )

!(:init
   ...
   (block white roomc1 bot_2)
   (block orange roomc2 bot_2)
   (handempty bot_2 )
   (at dropzone bot_2)
   (block yellow roomb2 bot_2)
   (goal_holding roomc3 blue bot_1)
   (block blue roomc3 bot_1)
   (handempty bot_1 )
   (at frontdropzone bot_1)
   (goal_in roomb1 bot_2 )
   ...
)
```

**TM Communication Algorithm**

The communication model and algorithm is introduced before the core planning algorithm because in the TM approach, the communication planning is integrated with planning for other actions. Therefore, the planner must be made aware of these communication actions and the pre-condition and effects. These actually come from the shared task model.

The messages in the shared task model can be categorised by the type of information that is communicated. For example, in BW4T all messages that inform other agents of the intention to collect a block can be represented using a single communication action specification, as shown in the example below. The only important effect is the *(goal_holding*
which adds the belief that the planning agent has the intention of collecting the particular colour from the specified room. The implication of this belief is that when other agents will know this intention, other agents can proceed to the next team goal. When planning, if the cost of this communication enables the team to devise a less costly plan, the planner will select this action.

```
(:action comm_int_block
  :parameters (?placeid - room ?c - color ?a - agent ?a2 - agent)
  :effect (and
    (goal_holding ?c ?placeid ?a)
    (not (curr ?a)) (curr ?a2) (increase (total-cost) 3)
  )
)
```

The cost of communication actions is also an important design choice. Higher communication cost discourages the use of communication and if the cost of communication is the same as the cost of other actions, agents may be liberal in using communication. Additionally, communication actions can be made compulsory, for example, those actions that setup or terminate the intentions. Making communication compulsory may be required due to the organisational level compulsory reporting policies. This means that while agents may have the autonomy to decide their communication behaviour, some of these behaviours may be dictated by the organisation that the agents work in. The method adopted here allows such communication policies to be encoded into the domain specification with a little effort.

**TM Planning Algorithm**

The MAP algorithm is shown in Algorithm 8. The first parts (lines 1-5) is the same as Algorithm 3 where the agents monitor their plans and keep executing it until they cannot, and compute others likely goals (5). Then the agent generates a team plan (6), \( \Pi_T \), using the SMM. The procedure, `get_my_components` extracts the goal and plan relevant to the planning agent (7). Notice that the extracted agent plan and the agent’s goal may not be yet know by all the agents as this is what the planning agent believes is the optimal team plan. The planning agent needs to ensure that all members are aware of the agent’s goal. If the *SMM State* does not already include information about the agent’s goal, the agent

129
communicates the goal to others (8-10). The complete process of accepting or rejecting the goal by others has been left out. In fact, the agents are given an opportunity to decline the setup of joint intentions, in which case a re-plan is triggered. However, intuitively, since the SMM is used for planning process, all agents should generate the same plan. The joint intentions, however, are required to ensure that these intentions are confirmed by all agents.

If the SMM State has the agent’s goal (11), then the agent can proceed to execute the plan (12). The planning process includes communication planning. This means that if useful, the planner will select the communication actions which have been encoded in the planning model. To handle the communication actions, the agent uses the shared task model to generate the natural language string associated with the communication action and sends the message to particular agents or broadcasts the message to the team (13-15).

Input: $\alpha_{prev}$ - previous action, $I$ - current state, $A$ - domain model, otherAgents - mental states of other agents, $\pi$ - current plan, SMM - shared mental model

Output: $\alpha$ - next action

1. if $\text{len}(\pi) > 0 \&\& \neg \text{failed}(\alpha_{prev})$ then
2. $\alpha = \text{next\_action}(\pi)$
3. return $\alpha$
4. else
5. $G_{opt} = \arg\min_{g \in G} p(g) \ \forall i \in I$
6. $\Pi_T = \text{plan}(T, SMM, G_{opt})$
7. $\text{goal}_{self}, \pi = \text{get\_my\_components}(\Pi_T, G_{opt})$
8. if $\text{goal}_{self} \notin \text{state\_info}(SMM)$ then
9. $\text{setup\_JointIntentions}(\text{goal}_{self})$
10. $\text{update\_SMM}(\Pi_T)$
11. else if $\text{goal}_{self} \in \text{state\_info}(SMM)$ then
12. $\alpha = \text{next\_action}(\pi)$
13. if $\alpha = \text{comm\_action}$ then
14. $\text{message} = \text{getMessage}(\alpha)$
15. $\text{sendMessage}(\text{message})$
16. end
17. return $\alpha$

Algorithm 8: Multiagent planning algorithm based on SMM.
Figure 6.4.3: MAS Platform and Architecture. The architecture is shown for two agents but is generic and scales to multiple agents.

6.5 MAS Platform and Architecture

The BW4T testbed (Johnson et al., 2009) used for the study in Chapter 5 was modified further to be able to setup the environment to perform the experiments. While the testbed was designed for the BW4T domain, the testbed itself can be used to simulate a number of different scenarios. The overall architecture shown in Figure 6.4.3. The testbed was initially designed to be used for experiments with human teams. Later Hindriks (2009) added interfaces such that GOAL agents could interact with BW4T testbed. This extension enabled experiments with artificial agent teams and mixed teams of humans and artificial agents. The testbed, however, did not support planning from first principles. Therefore, the testbed was modified such that the GOAL agent behaviour could be generated using an automated planner rather than writing the plan-rules as would be done when conventionally programming BDI (Rao et al., 1995) agents.

The testbed has a server component to generate the simulation and keep the simulation variables updated based on the actions of the agents. The server also provides the necessary perceptions to the client. The agents interact with a client, which provides action and perception interfaces. The GOAL agent leverages these client interfaces to perform actions and to receive perceptions.
Each GOAL agent is coupled with a separate agent instance in the planning system, as indicated by the dashed lines between the GOAL agents and the MAS Planning System. Each agent instance runs as a separate process and has access to its own instance of the Fast Downward planner. The GOAL agents dump their mental states in a log file, which forms the input to the planning system. Agents have access to their own log files and do not have access to the log files of the other agents. Using the mental states and various elements of the SMM, each agent generates a domain and problem specification and passes these as arguments when invoking the Fast Downward planner \cite{Helmert_2006}, which is a classical planner. The planner returns a plan, if one is found. The planning system is responsible for executing each action, which involves translating the action to the GOAL agent’s action format, and sending the action back to the GOAL agent (the action is sent to the testbed, which routes the action to the GOAL agent). When the GOAL agent performs the action using the client, the action effects are recorded by the server.

6.6 Evaluation

This section explains the details of the simulation-based experiments that were conducted using artificial agent teams to evaluate the success of the two planning approaches in making the communication related decisions.

Two domains were used to evaluate the planning approaches: 1) BW4T - Classical, which has been explained in Chapters \ref{chap:3} and \ref{chap:5}; and 2) a BW4T - Medical Kit Delivery scenario, which is explained in the next section. Both domains were simulated using the modified version of the BW4T testbed, and the same classical planner (Fast Downward) was used for both domains. The aspect that changed was the domain knowledge and the task models of the SMM. Both are domain dependent in terms of the tasks and goals, and their representations. The communication actions were encoded as PDDL actions in the domain specifications for the TM based planning approach. The examples represented earlier provide details of how this has been done. For the TP approach, the agents were given a set of messages using the message structures explained in Section 4.5 (page 67) of Chapter 4.

The procedure for conducting the experiments was the same for both domains. For each domain, at least 30 runs were completed for each setup (explained next) using the BW4T testbed. For each run, run-time data, such as the timing data reported by the planning system and the log files showing the activities of the GOAL agents, was used to compute the measures, such as task completion times. Statistical significance tests were conducted.
using R (R Core Team, 2013). The normality of the data was tested using the Shapiro-Wilk test, and the effects on the dependent variables were performed with a Welch’s t-test or the Mann-Whitney’s U test when one of the samples being compared failed the Shapiro-Wilk test.

6.6.1 EXPERIMENTAL DOMAINS AND MEASURES

This section discusses the two domains used to evaluate the planning approaches.

BW4T - CLASSICAL

A series of simulation experiments were conducted for the classical BW4T task using a 2-agent team completing a sequential task. The number and location of the blocks and the sequence of colours, that is, the joint goal, were randomly generated for each run. The environment contained nine rooms. The following measures were collected for each run:

1) Completion Time: This is the total time it takes the team to complete the task, that is, deliver the required colour sequence. This measure was used as a proxy for team performance. The task completion time includes time spent planning the actions as well as executing them.

2) Execution Time: This is the total time agents spend on executing actions including communication actions. This excludes time spent on planning.

3) Number of messages: This measure captures the total number of messages exchanged by the agents.

The results for this domain were compared with the experiments conducted in Chapter 5, in particular the experiments where the agents shared all of the world knowledge and intentions. This results in a full-communication setting, which can serve as a reasonable baseline. Recall that the agent behaviours were pre-scripted in the GOAL programming language, and these rules were fine-tuned to achieve quick completion times. In that setting, the agents used a pre-determined communication policy and did not reason about the benefit or cost of communication. However, the communication was hand-crafted to be minimal in the sense that the agents only communicated selected intentions and world knowledge.

BW4T - MEDICAL KIT DELIVERY

The BW4T domain was used as the basis for designing a second domain: the Medical Kit Delivery. In a hospital setting, a team of robots are required to assist nurses by delivering
medical kits and cleaning rooms. There are at least two kits stored in different rooms, in a hospital with nine wards. Each ward has a dedicated space to store the kits after use. When a nurse requires a kit, the nurse informs all robots of the number of the ward/room where the kit is required. When the robots are not delivering kits, they randomly move in the hospital cleaning (vacuuming) the wards.

If the robots share their intention about the ward they were going to clean, they avoid cleaning the same wards. Similarly, since there are multiple kits, if the agents share their intention of collecting the kits, they can avoid delivering a kit to the same ward. A nurse requests one kit at a time, and therefore, the robots need to coordinate in order to avoid multiple deliveries for one request. Therefore, it is easy to see that the team, that is, the robots, will perform better when the team will maintain and use SMM.

The messages that were encoded for the domain enabled the agents to communicate the following intentions:

1. Intention of ward to clean: This is similar to the BW4T scenario where agents exchange intention of which room each agent was going to search. Here the agents exchange their intention of which ward they were going to clean.

2. Intention of kit to collect: Since the agents have multiple kits to collect, this message communicated information about which particular kit an agent was going to collect, and from which room.

3. Intention for another to clean a ward: This message allows one agent to request another agent to clean a particular ward. This message may be used when one agent wants another to clean a particular ward while the requesting agent intends to collect and deliver a kit.

In addition, the following message was encoded to communicate the world knowledge:

1. Location of kit: This is information about the location of medical kits.

Similar to the BW4T domain, the SMM State is the focus of investigation as the other components are designer specified and do not change. Also, as has been stated earlier in this chapter, the intention related messages are a trigger point to the complete negotiation performed by the agents, which involves agents exchanging their intentions, and agreeing or rejecting the suggestions. A rejection of particular intention triggers a re-plan.

A series of simulation experiments were conducted using a 3-agent team (1 nurse and 2 robots) completing delivery and cleaning tasks. Two kits were requested for each run, and
these were generated randomly. A script, which simulated the role of the nurse, was written that generated the kit requests at random intervals ranging from 90 and 120 seconds. This interval was chosen as the agents had enough time to deliver the kits. Each run terminated after the second kit was delivered. The ward or room number where the kit was required was supplied by a nurse. The location where the kit was required changed between runs and the agents chose a ward to clean randomly. The following measures were collected for each run:

1) Completion Time: This is the total time it takes for the team to complete the task, that is, medical kit deliveries and any cleaning activities in between.
2) Execution Time: This is the total time agents spend on executing actions including communication actions. This excludes time spent on planning.
3) Number of messages: This measure captures the total number of messages exchanged by the agents.

### 6.6.2 Independent Variable

The independent variable for both experiments is the planning approach, that is, the TP approach and the TM approach. For the BW4T-Classical setting, the pre-scripted policy of Chapter 5 formed another variable and also a baseline for the domain.

### 6.7 Results

In this section, we discuss the results of the experiments conducted for the two domains. Summary of the results are shown in Table 6.7.1. Error bars shown in the figures in this section represent one standard deviation.

<table>
<thead>
<tr>
<th></th>
<th>Completion Time (s)</th>
<th>Execution Time (s)</th>
<th># Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>BW4T</td>
<td>Baseline</td>
<td>59.5 (2.9)</td>
<td>59.5 (2.9)</td>
</tr>
<tr>
<td></td>
<td>TP</td>
<td>395.3 (10.7)</td>
<td>97.4 (14.3)</td>
</tr>
<tr>
<td></td>
<td>TM</td>
<td>253.7 (16.2)</td>
<td>102.5 (12.7)</td>
</tr>
<tr>
<td>Medical</td>
<td>TP</td>
<td>593.83 (34.3)</td>
<td>293.83 (34.3)</td>
</tr>
<tr>
<td></td>
<td>TM</td>
<td>548.46 (33.2)</td>
<td>249.15 (36.1)</td>
</tr>
</tbody>
</table>

**Table 6.7.1:** Averages of the three measures for the two domains with the standard deviations shown in brackets.
6.7.1 BW4T - Classical

The results illustrate the success of the two planning approaches in reducing communication between agents when compared with the pre-scripted policy of Chapter 5, as shown in Figure 6.7.1. The number of messages exchanged by teams using the pre-scripted model presented in Chapter 5 is significantly higher than the two approaches proposed in this chapter. A Welch’s t-test found a significant effect of the approach ($t(28) = 19.16$, $p < 0.05$, Cohen’s $d > 2$) with the Pre-scripted model ($M = 19$, $SD = 2$ (rounded to nearest integer)) resulting in a significantly higher communication than the TP approach ($M = 8$, $SD = 1$), and ($t(28) = 16.89$, $p < 0.05$, Cohen’s $d > 2$) the TM approach ($M = 10$, $SD = 1$). This, I believe, is a significant result. The comparison here is with a hand-crafted policy that was explicitly designed to reduce communication for this specific task, but the two generalised methods outperformed this.

When comparing the TM and the TP approaches, it was anticipated that the TM approach would result in fewer communication than the other models. However, a Welch’s t-test found a significant effect of the approach ($t(38) = 3.72$, $p < 0.05$, Cohen’s $d = 1.18$), with TM approach resulting in more communication than the TP approach. This can be explained by the requirement of the TM algorithm to always setup the joint intentions. In the TP algorithm, the agent chooses between intentions and world knowledge.
Figure 6.7.2: Task completion times for the BW4T domain.

The task completion times are shown in Figure 6.7.2. Not surprisingly, the completion times for the pre-scripted model was significantly less than the other two approaches. The medians of the completion times for the Pre-scripted model, the TP approach and the TM approach were 60.0s, 392.2s, and 249.6s, respectively. A Mann-Whitney’s U test was used to evaluate the difference in the completion times, and a significant effect of the approach was found (U = 0, Z = -6.5398, p < 0.05, r = 0.86), with Pre-scripted model (M = 59.5s, SD = 2.88s) outperforming the TP approach (M = 395.3s, SD = 10.71s) as well as the TM approach (M = 253.7s, SD=16.23s) (U = 0, Z = -6.54, p < 0.05, r = 0.85).

The pre-scripted model required the agents to just do a lookup of the plan-rule for execution. That is, the designer did the planning for the agent. In the other two approaches, the agent made many decisions with the aid of an automated planner. Therefore, the task completion times includes time spent planning the actions as well as executing them. In addition, the choice of the execution control strategy added to the total completion time. The agents were given 2.25s to execute each action in the plan. This was done as an alternative to an approach that would have continuously monitored the state variables to determine the success or failure of an action. The performance may be improved by considering approaches that enable the agents to perform planning of future actions while
Figure 6.7.3: Task execution times for the BW4T domain. This is time for action execution only.

executing current actions.

Figure 6.7.3 shows the execution times only. The higher times for the two models proposed in this chapter can be a result of not being able to precisely separate the planning and execution times. There is a time difference between action completion and the time when the action completion is detected by the planning system. We ran a Mann-Whitney’s U test to evaluate the difference in the execution times, and found no significant effect of the approach (U = 513, Z = 1.4386, p > 0.1, r = 0.18), with the TM approach (M = 102.5s, SD = 12.69s) and the TP approach (M = 97.4s, SD = 14.25s) achieving similar execution times.

When compared with the no-comm model presented in Figure 5.5.3 on page 100, the no-comm performed better with the hand-crafted policy. In a Mann-Whitney’s U test to evaluate the difference in the execution times indicated a significant effect of the approach with the No-Comm model (M = 83.3s Sp = 4.26s) better than the TM (M = 102.5s, SD = 12.69s) and the TP approached (M = 97.4s, SD = 14.25s).
Figure 6.7.4: Task completion times for the Medical Kit Delivery domain.

Figure 6.7.5: Task execution times for the Medical Kit Delivery domain. This is time for action execution only.
Overall, similar trends were observed when compared with the BW4T-Classical domain. The TM approach performed better than the TP approach in terms of overall task completion times. In terms of communication cost, the number of messages exchanged by the agents using the TM approach was more than what the agents required when using the TP approach. This result is similar to the BW4T-Classical domain as well.

We compare the overall task completion times for the two planning approaches in Figure 6.7.4. The medians of the task completion times for the TP approach and TM model were 592.83s and 549.63s, respectively. We ran a Mann-Whitney’s U test to evaluate the difference in the task completion times, and found a significant effect of the approach (U = 1168, Z = 4.6334, p < 0.05, r = 0.61) with the TP approach (M = 593.83s and SD = 34.29) having higher completion times than the TM approach (M = 548.46s and SD = 33.20).

In case of execution only times (Figure 6.7.5), the medians of the execution times for the TP approach and the TM approach were 292.82s and 253.41s, respectively. We ran a Mann-Whitney’s U test to evaluate the difference in the execution times, and found a significant effect of the approach (U = 1168, Z = 4.6334, p < 0.05, r = 0.61) with the TP approach (M = 293.83s and SD = 34.29) having higher execution times than the TM approach (M = 249.15s and SD = 36.10). This may suggest that in the TM approach, the agents may have executed less number of actions. However, this cannot be solely attributed to the
planning approach. This may have been a consequence of the domain models. Note that the two planning approaches use different models. In the TM approach, the communication actions were encoded as part of the model and the planner was used to search for a solution including communication actions. This was not the case with TP approach, which did not have any communication actions as part of the model itself.

When comparing the two approaches in terms of the number of messages exchanged by the agents (Figure 6.7) during task execution, the TM approach required (moderately) more communication than the TP approach. The medians of the number of messages for the TP approach and the TM approach were 11 and 12, respectively. We ran a Mann-Whitney’s U test to evaluate the difference in number of messages, and found a moderate effect of approach ($U = 315$, $Z = -2.0175$, $p = 0.04$, $r = 0.26$) with the agents using the TP approach ($M = 11$ and $SD = 1$) having communicated (moderately) less number of messages than agents using the TM approach ($M = 12$ and $SD = 2$).

6.8 Conclusion

Chapter 5 used a pre-scripted (hand-crafted) communication policy to get the agents to communicate particular types of information as soon as the agents had the information available to them. This chapter proposed two generalisable communication planning approaches to get the agents to communicate the same types of information. The first approach, labelled as the TP approach, performs temporal projection of individual member goals and plans, and performs communication planning afterwards. The second approach, labelled as the TM approach resembles the SMM closely in terms of treating the group of agents working towards common goals as a single unit, that is, a team. The TM based approach plans for all members, that is, the team, and injects communication actions into the team model during the planning process. This has the ability to determine when, what and to whom aspects of communication during team-level planning.

Overall, the results illustrate the success of the two approaches in getting the agents to communicate the two SMM State sub-components. The results also illustrate a trade-off between the task completion times and communication between the two approaches. While performing team-level planning using the TM approach resulted in better completion times than the TP approach, for reasons explained earlier, the agents using the TM approach required more communication than the TP approach. However, as the algorithms demonstrate, overall less effort was required on the part of designer when using the TM approach and the TM approach is more closely aligned with SMM. When using the TM approach,
the designer had to consider the overall team model and the team plan. The TP approach required consideration of individual member plans and a meta-level planning step to determine a combination of individual member plans that would result in a team plan. Not only does the TM approach make the meta-level planning step of TP implicit in the multi-agent planning, the TM approach also provides an opportunity to consider team-level communications rather than communications that stem from coordinating individual member plans of the TP approach. I believe that the TM approach is a better approach to determining team-level communication decisions. In addition, both approaches are generalisable to other domains when compared with the pre-scripted policy of Chapter 5, which was hand-crafted for the particular domain and will work for that domain only.

The results indicate the success of the two approaches in reducing the amount of communication between agents when compared with the pre-scripted policy. More importantly, there is a shift in responsibility for communication planning from the designer to the agents. Many existing approaches have been discussed and these works have provided great insight into achieving this goal. The difference lies in not making communication planning an option, as in Wu et al. (2011) for example, but to treat communication as an integral part of team coordination and a team activity that is compulsory but expensive. Therefore, effort was made to look for ways of balancing communication costs with performance and execution time.

One limitation of the models used for both approaches is the underlying deterministic planning model, which limits the automated planners options to explore different possibilities during the planning process or generating complete or partial policies. Non-deterministic planning (Ghallab et al., 2004) could be explored in future work. On a similar note, epistemic planning technologies (Muise et al., 2015) that explore ways of encoding nested beliefs using generalisable methods could be explored as future work. In addition, the effect of using the sharedness value to trigger communication, that is, determine when to communicate, could be explored in future works.

The agents exchanged only parts of the SMM State component of the SMM proposed in Chapter 4. Another direction for future work would be to consider other types of SMM related information. For example, in this chapter, the domain model was shared a priori and did not change throughout the experiments. However, it could be possible for an experienced human, who may have additional domain knowledge, to share this knowledge with the artificial agents. The proposed approaches are generalisable to different types of information but future work could assist by identifying the representation that conveys this information effectively.
Conclusion and Future Work

A key requirement for effective teamwork is the development of a shared understanding of the team and tasks between members, and the use of this shared understanding when members engage in interdependent tasks. This thesis used shared mental models (SMM) as a way of establishing and maintaining the shared understanding between artificial agents. While the long-term goal is to design for human-agent teamwork, the scope of this thesis had been limited to identifying the data structures and algorithms that enabled artificial agents to represent and use SMM. This was an important first step in designing artificial agents that would leverage SMM for decision-making. This chapter first restates the research questions and then provides a summary of the contributions of the chapters followed by some limitations. Then the practical implications of the research are discussed. Finally, this chapter states directions for future work.

Inspired by SMM related research in human teams as well as formal models of teamwork, this thesis endeavoured to answer the following questions:

1. What are the core components of SMM that are required when designing and implementing artificial agents and what representational approach for these components enables artificial agents to use SMM for decision-making?

2. How does the level of interdependence between agents impact the development and use of SMM?

3. How can SMM be injected into the communication planning process to enable artificial agent teams to establish SMM?
7.1 Contributions

This section highlights the contribution of each chapter to the thesis and highlights some limitations of the studies.

The study presented in Chapter 4 answers the first research question. The chapter discusses the proposed computational SMM. The aim was to identify the key data structures and algorithms or processes that enable artificial agents to use SMM in their decision-making processes. While the literature on human teams predominantly identified the data structures of the SMM, when designing and implementing artificial agents, key processes that would allow artificial agents to establish, maintain and use the SMM are also needed. Informed by the literature on SMM for human teams and formal models of teamwork, the following five data structures have been identified to capture the required SMM knowledge: 1) SMM State; 2) Task Model; 3) Interaction Model; 4) Member Models; and 5) Domain Model. For the key SMM processes, it was useful to categorise these processes into three classes based on their function. The three categories of processes are: (1) Initialisation - to initialise SMM with prior knowledge; (2) Update and Maintenance - to keep the SMM updated; and (3) Utilisation - to use the SMM for decision-making. This model formed the basis of the agents designed and implemented in this thesis as well as the empirical studies presented in Chapters 5 and 6. The data structures were represented using predicates, which defined the domain related concepts and these were used by the agents to represent their beliefs and the decision making rules.

Answering the second research question involved first understanding the term interdependence in more detail (Chapter 3) and then conducting empirical studies to establish the relationship between SMM and interdependence (Chapter 5). In the study presented in Chapter 3, the aim was to make clear distinctions between the different types of interdependence relationships. This lead to a fine-grained analysis of the different forms of interdependence and the proposal of a number of semi-formal definitions of the different types of interdependence. Examples demonstrated that the fine-grained analysis was helpful in identifying the potential communication requirements and that these information requirements form part of the SMM. Examples also illustrated that such formal frameworks allow designers to consider different design options and how knowledge of the type of task, goal, and agent interdependence can assist designers in choosing different agent coordination mechanisms. Another important result of this work was that task interdependence could be used to induce increasing levels of interdependence between agents. This formed an important part of the analysis and paved a way for the study presented in Chapter 5,
which explored the relationship between SMM and levels of interdependence.

In the study presented in Chapter 5, which was an empirical study, the aim was to identify whether different levels of agent interdependence required agents to communicate particular components of the SMM in order to improve their team performance. That is, the chapter established what agents should communicate with increasing levels of interdependence. Given that exploring all components of the SMM was infeasible in this thesis, only the SMM State component was explored. In particular, only two sub-components of the SMM State, namely joint intentions and task state, were used for the study. The different forms of task interdependence was used to induce the various levels of agent interdependence. The results gathered using simulation-based experiments illustrated that with increasing levels of interdependence between agents, the importance of intentions to the team increases as well. This happens because as the level of interdependence increases, the agents are required to work closely and in some cases concurrently. In such cases, knowing others’ intentions allows one to adapt their behaviour to the actions of others. When putting these findings in the context of SMM, these findings point out which of the tested sub-components of the SMM proposed in Chapter 4 are impacted by the increasing levels of agent interdependence. The joint intentions become more important as interdependence increases while task state becomes more important for less interdependent tasks. Chapter 5 also established that team composition played a role in determining which sub-component of the SMM State contributed more to team performance. In team compositions where agents could perform their tasks independently, task state contributed more to team performance. When multiple team members may be engaged in a single sub-task, the potential of interference increases and so does the importance of knowing the intentions of others. This means that in case of joint actions, establishing joint intentions is imperative.

Chapter 6 provides an answer for the final research question. While Chapter 5 established what the agents should communicate, providing the agents with a hand-crafted communication policy, as was done in Chapter 5, is not generalizable. Therefore, the final study presented in Chapter 6 explored generalizable planning approaches to inject SMM into the decision-making process, and focused particularly on communication processes. This chapter was primarily concerned with how the agents decided what, when and to whom to communicate to. Two planning approaches are proposed, that is, the temporal-projection based (TP) and team-model based (TM). Overall, the results illustrate the success of the two approaches in getting the agents to communicate the two SMM State sub-components. Results also suggested that there was a trade-off between the two approaches in terms of the overall task completion times and communication. While performing team-level plan-
ning using the TM approach resulted in better completion times than the TP approach, the agents using the TM approach required more communication than the TP approach. However, as the algorithms presented in the chapter demonstrated, overall less effort was required on the part of designer when using the TM approach and the TM approach was more closely aligned with SMM. In addition, both approaches are generalizable when compared with the hand-crafted communication policy. More importantly, using the two approaches, there is a shift in responsibility for communication planning from the designer to the agents.

While the above paragraphs highlight the key contributions of this thesis, the following points highlight some limitations of the research. Firstly, the empirical studies presented in Chapters 5 and 6 assumed complete and accurate knowledge of the decision making processes of each member and assumed that the members were trustworthy. This helps the proposed algorithms make accurate predictions of the other agent’s goals and plans. This thesis has not investigated cases when this information is incomplete or inaccurate. Secondly, the domains studied in this thesis resulted in members being tightly coupled and having a flat team structure where each member was equal in terms of their placement within the team. While there was some effort made to investigate sub-teams in Chapter 5, the computational SMM has not been validated in the context of more complex teams with hierarchical structures. Finally, the communication channels were assumed to be perfectly reliable. Unreliable communication channels may hinder the development of SMM.

7.2 Practical Implications

Research presented in this thesis has implications for the design of interfaces of systems requiring human-agent teamwork. The term interface is used to refer to the overall interaction mechanisms, which could be in the form of graphical user interfaces or explicit communication between members. These interaction mechanisms will be used by the humans and the artificial agents to interact with each other.

Clearly, common ground is an important factor in realising effective human-agent teamwork. SMM identifies the different types of information that human teams find useful in promoting effective teamwork. Therefore, when artificial agents are designed to reason with and about the different components of SMM, and communicate the different types of information to establish SMM, there is a greater chance of effective teamwork between the humans and agents.

Similarly, when the user interfaces are designed for the human partners, SMM provides guidance on what types of information the human may find useful when interacting with
an artificial agent, and going back to the point made in the previous paragraph, what the artificial agents could communicate to the human. Identifying the SMM components has relevance to the Coactive Design Method (Johnson et al., 2014), which is based on the concepts of observability, predictability, and directability (OPD). “Observability means making pertinent aspects of one’s status, as well as one’s knowledge of the team, task, and environment observable to others... Predictability means one’s actions should be predictable enough that others can reasonably rely on them when considering their own actions. The complementary relationship is considering others’ actions when developing one’s own... Directability means one’s ability to direct the behaviour of others and be directed by others” (Johnson et al., 2014). The OPD requirements guide the development of the automation (robot in their case) and the user interface that allows the actors to manage the interdependence relationships. The SMM components identified in this thesis can assist in identifying what agents need to make observable through the human-agent interface.

Furthermore, there are implications on what aspects to focus on when training the humans and artificial agents to work together. Recall that SMM has prior information that may be accrued through experience of working together. Therefore, training both parties to work with each other is an important exercise. Designers of these training exercises could potentially focus on activities that help foster particular components of SMM, for example, learning each other’s capabilities or preferences.

7.3 Future work

There are multiple directions for future research.

Human-agent experiments: Future work should validate the proposed model and the communication models in experiments involving artificial agents and humans. Both domains used in Chapters 5 and 6 were simulated using the BW4T testbed, which has interfaces for human users as well. Therefore, performing experiments between humans and artificial agents to validate the model is possible with the existing testbed.

However, in the long-term, research is needed to know how to capture the mental states of the human partners at runtime. There are methods employed by researchers considering situation awareness (Endsley, 1995b). There two methods discussed in literature: 1) Situation awareness global assessment technique (SAGAT) (Endsley, 1995a) and Situation Present Awareness Method (SPAM) (Durso et al., 2004). While in SAGAT technique, questions are asked by pausing the team activity, the SPAM technique poses questions while
the task in progress. While these two options provide potential methods of assessing the mental states of the humans, these could potentially be augmented with analysis of verbal and non-verbal communication that may, hopefully, eliminate the need for frequent interventions. As one can imagine, this is a non-trivial task.

When working with teams of artificial agents and humans, research is needed to understand how much information to share, that is, to determine how much overlap between team members is sufficient. When working with humans, it will not be possible for humans to share all of their beliefs. Similarly, the SMM processes proposed in Chapter 4 need to be improved to handle inconsistent beliefs. In addition, the model discussed in this thesis assumes that the agents will have an accurate model of their team members. When dealing with humans, it may be challenging to specify the models accurately. Therefore, research is needed to learn how human models could be elicited or specified. This may happen via communication, which can be unreliable as well, or the agents may use communication to refine the models. Another approach is may be having the agents learn the capabilities of the human member.

**Multi-modal communication:** While this thesis only considered explicit message passing communication, clearly, humans communicate using a number of verbal and non-verbal actions (Turk, 2014). While the underlying model will be able to support multiple modalities, work still needs to be done to integrate text-based, speech-based and other non-verbal behaviours such as gestures, and integrating these into the system is clearly non-trivial. Recent studies (Hanna and Richards, 2014, Hanna et al., 2015) demonstrate the challenges of using verbal and non-verbal communication to establish SMM between artificial agents and humans, such as the challenges that stem from the fact that the human and the artificial agents are situated in different worlds. One is in real while the other in virtual. These studies can provide great insight when exploring multimodal interaction.

**Epistemic planning:** While in the thesis, each agent maintained a model of the other, encoding of the beliefs of other agents was achieved using a simple approach, that is, by appending the names of the agents to their respective beliefs. This was sufficient for the kind of tasks studied in this thesis but the approach was limited to depth of 1 in terms of nesting beliefs. Epistermic planning literature (Bolander and Andersen, 2011, Muise et al., 2015) provides guidance on how to capture nested beliefs in general. Using these methods agents will be able to nest their beliefs to depths of more than 1 and solve these models using computationally tractable methods.
Human-agent interface design: As alluded to earlier, SMM provides guidance on what types of information the human may find useful when interacting with an artificial agent. It would important to know how to represent the different components of the SMM to the human to foster transparency and trust between the two parties. A promising area to explore is Ecological interface design (EID) (Dinadis and Vicente, 1996; Jamieson, 2007; Reising and Sanderson, 2002; Shattuck et al., 2008; Vicente, 2002; Vicente and Rasmussen, 1990), which is an approach used to design interfaces for complex dynamic systems with the aim of highlighting salient aspects of the system to the human user in intelligent ways. Hanna et al. (2014) and Hanna and Richards (2014) studied the relationship between trust and SMM and these studies can be used to inspire work in this direction.

7.4 Final remarks

Taking inspiration from concepts that work well for human teams can provide great insight when designing artificial agents that will be required to function as effective team members when working in mixed teams of humans and agents. Inspired by this thought, this thesis proposed and empirically validated a computational SMM. I believe that this model will allow artificial agents to be effective teammates and more widely trusted and accepted by humans.
References


