EFFICIENT REPRESENTATION AND EFFECTIVE REASONING FOR MULTI-AGENT SYSTEMS

By

Duy Hoang Pham

THE UNIVERSITY OF QUEENSLAND

A THESIS SUBMITTED FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

AT THE UNIVERSITY OF QUEENSLAND

IN APRIL 2010

SCHOOL OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

Typeset in LaTeX 2ε.

Declaration and statements

Declare by author

This thesis is composed of my original work, and contains no material previously published or written by another person except where due reference has been made in the text. I have clearly stated the contribution by others to jointly-authored works that I have included in my thesis.

I have clearly stated the contribution of others to my thesis as a whole, including statistical assistance, survey design, data analysis, significant technical procedures, professional editorial advice, and any other original research work used or reported in my thesis. The content of my thesis is the result of work I have carried out since the commencement of my research higher degree candidature and does not include a substantial part of work that has been submitted to qualify for the award of any other degree or diploma in any university or other tertiary institution. I have clearly stated which parts of my thesis, if any, have been submitted to qualify for another award.

I acknowledge that an electronic copy of my thesis must be lodged with the University Library and, subject to the General Award Rules of The University of Queensland, immediately made available for research and study in accordance with the Copyright Act 1968.

I acknowledge that copyright of all material contained in my thesis resides with the copyright holder(s) of that material.

Statement of contributions to jointly authored works contained in the thesis

The thesis contains the following joint works:

1. Governatori and Pham (2005a,b). I was only responsible for implementing the reasoning mechanism and its interfaces with RDF and XML.
2. Governatori et al. (2008). I was only responsible for implementing the reasoning mechanism and designing the markup language for the modal defeasible logic.
3. Pham et al. (2008a,b,c). I was responsible for defining the problems and developing the solutions in the works. The co-authors have contributed by discussing the problems, checking technical errors and improving the readability of the works.

Statement of contributions by others to the thesis as a whole

No contributions by others.

Statement of parts of the thesis submitted to qualify for the award of another degree

None.

Published works by the author incorporated into the thesis

- (Pham, 2008; Pham et al., 2008a) - Incorporated in Chapter 5.
- (Governatori and Pham, 2005a,b; Governatori et al., 2008) - Partially incorporated as the section of the implementation of reasoning mechanisms and the designs of the markup languages for defeasible logics in Chapter 5.
- (Pham et al., 2008b,c) - Incorporated in Chapter 6.

Additional published works by the author relevant to the thesis but not forming part of it

None.

Duy Hoang Pham

Acknowledgements

PhD research is distressing, but an enjoyable experience after years of struggling to define the research problem and to publish the work. Many people have helped me to get through this difficult process. It is an opportunity for me to thank them all.

First and foremost, I owe the maturation of my thesis to my advisory team including Dr Guido Governatori, A/Prof Robert Colomb and Dr Nghia Duc Pham (Queensland Research Lab, NICTA). I am greatly indebted to my advisors for their time, effort and energy for the development of my knowledge and my research skills as well as for my work. I highly appreciate the logical lessons and discussions that Guido gave me for my very first steps in the field of knowledge representation and reasoning. His talent and inspiration have guided me through difficult moments, both at work and in my daily life.

I would like to express my gratitude to A/Prof Abdul Sattar and the members of the SAFE-Agents group at Queensland Research Lab, NICTA. The time at the group is unforgettable and valuable, not simply for the research experience, but more importantly, for acquiring friends and developing friendships.

I wish to thank the members of my Data Knowledge Engineering group at the School of IT and Electrical Engineering, especially my logic group (Vineet Nair, Insu Song, Simon Raboczi and Subhasis Thakur). The regular meetings and seminars provided me with plenty of ideas, valuable suggestions and encouragement. I greatly owe Dat-Cao Ma, Simon Raboczi, and Vineet Nair for their patience and wisdom in correcting my lengthy papers.

My special thanks to Ms Kate Williamson (School of IT and Electrical Engineering), the NICTA staff including Mr Mark Medosh and Mrs Barbara Duncan for tackling the complexity

of the paperwork of all kinds during my research/setup, a working environment, applying for a fund, planning conference journey, just to name a few.

I would like to thank Mr Tony Roberts for his excellent editing service. His valuable suggestions have substantially improved the readability and consistency of my thesis.

It would be a big mistake if I did not acknowledge the substantial support from my sponsors. Without the financial support from the Ministry of Education and Training in Vietnam (under the Project 322), I could not come to Australia for my research. I also thank The University of Queensland for the financial assistance and the Queensland Research Lab, NICTA for the top-up scholarship during my research. Finance for conferences and workshops is always a problem for research students. Again, I am grateful to A/Prof Abdul Sattar for his generosity. The support from the NICTA allowed me to publish and present most of my work. Thanks also are due to the organising committees of Advanced Modal Logics 2006, Australia AI 2007 and Knowledge Representation and Reasoning 2008 for providing me with a student scholarship for attending the conferences.

Naturally, I greatly thank my parents and my family, who are a constant source of help in all aspects of my life, and my grandparents who have shown a huge encouragement and inspiration in my study but could not wait to witness my completion. Last but not the least, I owe Nga, my wife, and Nhi, my daughter, for their love and for reminding me how relative things are.

List of publications

- Guido Governatori and Duy Pham Hoang, *DR-CONTRACT: An Architecture for e-Contracts in Defeasible Logic*. In Claudio Bartolini, Guido Governatori, and Zoran Milosevic (eds). Proceedings on the 2nd EDOC Workshop on Contract Architecture and Languages (CoALa 2005). Enschede, NL, IEEE Press, 2005.

- Guido Governatori and Duy Pham Hoang, *A Semantic Web Based Architecture for e-Contracts in Defeasible Logic*. In A. Adi, S. Stoutenberg and S. Tabet (eds). Rules and Rule Markup Languages for the Semantic Web. RuleML 2005, pp. 145-159 (2005) LNCS 3791, Springer, Berlin, 2005.

- Duy Hoang Pham, Guido Governatori, and Simon Raboczi, *Agents adapt to majority behaviours*. IEEE 2008 International Conference on Research, Innovation and Vision for Future in Computing and Communication Technologies, pp. 7-12 (2008). Ho Chi Minh, Vietnam, 2008.

- Duy Hoang Pham, *Efficient Representation and Effective Reasoning for Multi-Agent Systems*. Doctoral Consortium in International Conference on Principles of Knowledge Representation and Reasoning, Sydney, Australia, 16-19 September 2008.

- Duy Hoang Pham, Subhasis Thakur, Guido Governatori, *Defeasible Logic to Model n-Person Argumentation Game*. Proceedings of the Twelfth International Workshop on Non-Monotonic Reasoning, pp. 215-222 (2008). Sydney, Australia, 13–15 September 2008.

- Duy Hoang Pham, Subhasis Thakur, Guido Governatori, *Settling on the Group's Goals: An n-Person Argumentation Game Approach*. 11th Pacific Rim International Conference on Multi-Agents (PRIMA 2008), Hanoi, 15-16 December 2008. Hanoi, Vietnam, 15-16 December 2008.

- Duy Hoang Pham, Guido Governatori, Simon Raboczi, Andrew Newman, and Subhasis Thakur. *On extending RuleML for modal defeasible logic*. In Nick Bassiliades, Guido Governatori, and Adrian Paschke, editors, RuleML 2008: The International RuleML Symposium on Rule Interchange and Applications, Lecture Notes in Computer Science, Berlin, 2008. Springer.

Abstract

A multi-agent system consists of a collection of agents that interact with each other to fulfil their tasks. Individual agents can have different motivations for engaging in interactions. Also, agents can possibly recognise the goals of the other participants in the interaction. To successfully interact, an agent should exhibit the ability to balance reactivity, pro-activeness (autonomy) and sociability. That is, individual agents should deliberate not only on what they themselves know about the working environment and their desires, but also on what they know about the beliefs and desires of the other agents in their group. Multi-agent systems have proven to be a useful tool for modelling and solving problems that exhibit complex and distributed structures. Examples include real-time traffic control and monitoring, work-flow management and information retrieval in computer networks.

There are two broad challenges that the agent community is currently investigating. One is the development of the formalisms for representing the knowledge the agents have about their actions, goals, plans for achieving their goals and other agents. The second challenge is the development of the reasoning mechanisms agents use to achieve autonomy during the course of their interactions.

Our research interests lie in a model for the interactions among the agents, whereby the behaviour of the individual agents can be specified in a declarative manner and these specifications can be made executable. Therefore, we investigate the methods that effectively represent the agents' knowledge about their working environment (which includes other agents), to derive unrealised information from the agents' knowledge by considering that the agents can obtain only a partial image of their working environment. The research also deals with the logical

reasoning about the knowledge of the other agents to achieve a better interaction.

Our approach is to apply the notions of modality and non-monotonic reasoning to formalise and to confront the problem of incomplete and conflicting information when modelling multi-agent systems. The approach maintains the richness in the description of the logical method while providing an efficient and easy-to-implement reasoning mechanism. In addition to the theoretical analysis, we investigate *n*-person argumentation as an application that benefits from the efficiency of our approach.

Keywords

multi-agent systems, defeasible logic, non-monotonic reasoning, artificial intelligence

Australian and New Zealand Standard Research Classifications (ANZSRC)

- 080101- Adaptive Agents and Intelligent Robotics: 50%.
- 080203- Computational Logic and Formal Languages: 50%.

Contents

List of Figures

List of Tables

1 Introduction

In this chapter, we briefly explain the concept of a multi-agent system and the general issues related to the development of multi-agent systems. We then introduce our approach to the problem of knowledge representation and reasoning by considering the condition of incomplete and conflicting information. At the end, we present the outline of the work presented in the thesis.

1.1 Multi-agent systems

1.1.1 A glance at the field

Multi-agent systems could not be reduced to simple collections of individual agents, because the agents in the systems interact with each other by different fashions to fulfil their designated

tasks. The major topic of multi-agent researches is to investigate the interactions between the agents, which are computational entities having the ability to act autonomously in their environment on behalf of their owners. Autonomous actions imply that the agents could work out the sequences of actions required to achieve their designated objectives at a certain level of optimisation. In other words, agents are aware of their activities and do not simply follow pre-assigned procedures towards the objectives.

Technically, interactions among the agents are carried out by the exchange of messages so that agents can gain more knowledge about their environment and other agents to fulfil their goals. Coordination is a very important and interesting type of interaction. Coordination is considered in a shared environment where agents need to coordinate to solve a problem. According to Weiss (1999), there are two kinds of coordination, cooperation and competition. In cooperation, the agents work as a team to achieve their common goals; the agents in a team succeed or fail together. However, in competition, the agents' goals may be in conflict with each other. As a result, the individual agents try to maximise their benefits at the cost of the other agents'.

Multi-agent systems are a useful tool for modelling and solving problems having complex structures, such as real-time traffic control and monitoring (Burmeister et al., 1997; Dresner and Stone, 2004; Durfee, 1996; Fischer, 1996; Ljungberg and Lucas, 1992), work-flow management in enterprise (Huhns and Singh, 1998; Merz et al., 1997; Singh and Huhns, 1999), information retrieval over the Internet (Decker et al., 1997; Sycara et al., 1996; Zhang and Lesser, 2006) and electronic commerce (Schrooten and de Velde, 1997; Sierra, 2004; Tsvetovatyy et al., 1997). The multi-agent approach can offer robust (no human intervention) and flexible solutions, because individual agents can autonomously work towards goals and, more interestingly, can interact with each other to complete the tasks.

1.1.2 General issues

Successfully building multi-agent systems involves a number of challenging issues. In fact, resolving those issues requires support from many disciplines, such as economics, philosophy, logic and social sciences. Bond and Gasser (1988) show typical aspects that should be taken into account when designing a multi-agent system:

- Agents should know how to represent, manipulate and distribute their goals and tasks to other agents to coordinate and synthesise the results.
- Appropriate languages and protocols are the first requirements for agents for effective interactions among the agents.
- Representation and reasoning about the actions, plans and knowledge of other agents are another challenge for interacting within multi-agent systems.
- Agents need to understand how to represent and reason about the state of their interaction processes. This helps the agents to evaluate the progress in their coordination efforts and to improve their coordination.

Over and above those issues is the great importance of having a formal tool to describe the multi-agent systems and the interactions between the agents to ensure that the system complies with the specifications.

1.2 Research aims

There are two broad challenges that the agent community is currently investigating. One is the development of formalisms for representing the knowledge the agents have about their actions, goals and plans for achieving their goals, and other agents. The second challenge is the development of the reasoning mechanisms which agents use to achieve autonomy during the course of their interactions.

Our research aims to build a multi-agent framework where an agent can efficiently reason about other agents in a group. Our framework considers the logical formalism to represent agents' knowledge, which is a partial image of the working environment and can contain conflicting information from other agents. Furthermore, we aim to construct an efficient reasoning mechanism so that it can be easily implemented and verified. Among logical approaches, defeasible logic efficiently tackles the problem of incomplete and conflicting information in terms of the representation and reasoning. Also, the majority rule (the social choice) can be a simple but efficient method to reach a common acceptance within a group in the presence of conflicts. By

combining the extended defeasible logic and the majority rule, our framework can efficiently represent and effectively reason about different types of knowledge within a group of agents. Interestingly, our reasoning mechanism can tackle with the paradox of the social choice. In addition, the framework targets to model a complex interaction between agents. That is the dialogue between n parties (agents), where agents argue to reach a majority acceptance not only for a conclusion but also for its explanation (proof of the conclusion). Our extended reasoning mechanism allows agents to efficiently tackle with the emergent and possibly conflicting knowledge from other agents during the course of dialogue. We have succeeded to construct a multi-agent framework with simple representation and efficient implementation. The framework allows us to reason about other agents and to model dialogue between n agents using existing techniques namely defeasible logic and the social choice with a minimal overhead.

In particular, our research interests are for a model where the interaction among the agents and the behaviour of the individual agents can be specified in a declarative manner and those specifications can be executable. Therefore, we investigate the methods that effectively represent the agents' knowledge about their working environment (which includes other agents), to derive unrealised information from the agents' knowledge by considering that the agents can obtain only a partial image of their working environment.

Our research also investigates the integrations between the notions of modality and non-monotonic reasoning to formalise and confront the problem of incomplete and conflicting information when modelling multi-agent systems. The approach maintains the richness in the description of the logical method while providing an efficient and easy-to-implement reasoning mechanism.

To balance between the expressiveness and the computational tractability, we extend the formalism of defeasible logic by Billington (1993) to capture different types of agents' knowledge. Also, we develop the reasoning strategies to identify beliefs common to a group of agents and to solve the conflicting knowledge obtained from other agents. As a result, our agents can reason about the others and, hence, achieve a better interaction.

The computational efficiency is another concern for our model. We can show that the complexity of the extended reasoning mechanism is proportional to the size of the agents' knowledge – that is, the multiplication of the number of rules and of the literals constituting an agent'

knowledge. Besides the theoretical analysis, we investigate n-person argumentation as an application that benefits from the efficiency of our technique.

Resulting from our research, we initiate the Defeasible Rule Markup (DRM) to facilitate the exchange over the Internet of the descriptions of the agents' behaviour. In addition, we develop a Java-based package supporting defeasible reasoning over the DRM descriptions. The package also aims at the interactions between the modal notions in the reasoning process. The differences in the interactions result in different types of agents. The simple approach is to separate modal notions into different layers. Then, the interaction is predefined when designing the agents. However, the more complex and flexible method is to define the interaction itself as a parameter of the reasoning process. That is our goal in the continuing development.

1.3 Thesis outline

The thesis contains a total of seven chapters. In the next three chapters, we present an overview of the modelling of the multi-agent systems (Chapter 2), especially using logical approaches (Chapter 3). In these two chapters, we aim for a 'trade-off' between the expressiveness of the modelling tools and the computational tractability of the implementation, especially when capturing the incomplete and conflicting information. With respect to this issue, defeasible logic (Chapter 4) has proven an efficient method. Our modelling technique for the agents' interaction based on defeasible logic is presented in Chapter 5, followed by an application of n-person argumentation in Chapter 6. We conclude the thesis in Chapter 7. The detailed structure of the thesis is as follows.

Chapter 2 sketches some of the basic elements of multi-agent systems. In particular, we briefly introduce the concept of intelligent agents that is the basic building block for any multi-agent system. We then elaborate the interaction among the autonomous agents. The chapter ends with the different techniques for modelling multi-agent systems. Chapter 3 provides a brief on the formal tools for the description and the control of the behaviour of the individual agents in addition to a group of agents. The chapter starts with the classes of logic to represent the different aspects of the agents' knowledge. We then present logic programming approaches to implement the multi-agent systems.

Chapter 4 introduces defeasible logic following the formalisation of Billington (1993) and the extension of the logic with ambiguity propagation. The logic provides a simple but very efficient model for confronting the problem of incomplete and conflicting information. The chapter continues with the investigation of the different implementations of defeasible logic. Finally, the chapter is concluded with a discussion on the relationship between defeasible logic and logic programming when dealing with incomplete and conflicting information.

Chapter 5 proposes a formal framework, DL-MAS, based on the defeasible logic for multi-agent systems. The framework aims to provide a declarative and executable model of agents' knowledge, in particular, the knowledge commonly shared by agents, and that obtained from other agents. In the new framework, the actions of an individual agent are constrained to a general expectation of the group of agents by balancing the desires of an individual with the beliefs of the majority. To have a fine-grained model of 'mental attitudes' and social actions, the DL-MAS is extended with modal notions including Belief, Intention and Obligation. In this model, our agents have the ability to discover the 'conventions' of the group by exploring the majority of the mental attitudes of the group.

In detail, we first introduce our modelling technique to represent the knowledge base of the agents including the meta-knowledge about the agents' importance. Also, we describe details of the DL-MAS reasoning mechanism and its implementation. We show that the extended reasoning mechanism does not increase the computational complexity of defeasible reasoning. Next, we show the integration of modal notions into our DL-MAS framework, following by the overview of research works related to our system. The chapter ends with a summary of research results.

Chapter 6 presents an application in n-person argumentation where the agents benefit from the efficiency of the representation and the reasoning technique of the DL-MAS. During the argumentation, our agents exploit the knowledge that other agents expose and, therefore, pursue a reasoning strategy to promote and defend its arguments. In this chapter, we first investigate the construction of the arguments using defeasible reasoning with respect to ambiguous information. Second, we present the technique to model n-person argumentation with regard to the DL-MAS. We relate our approach with other research works, then summarise the chapter.

Chapter 7 concludes the thesis with a summary of the main contributions and a discussion

on future research issues.

1.4 Bibliography note

Most of the research presented in the thesis has been published in some form. The main result of Chapter 5 on the method for knowledge representation and reasoning, DL-MAS, has been accepted at the RIVF 2008 IEEE International Conference in Vietnam (Pham et al., 2008a). It has also appeared as a poster paper at the Doctoral Consortium in the KR 2008 International Conference, Australia. Chapter 6 is the combination of Pham et al. (2008b,c) respectively, presented at the International Workshop on Non-monotonic Reasoning held in Australia and the Pacific Rim International Conference on Multi-agents.

The early design and implementation of the defeasible rule markup have been shown in Governatori and Pham (2005a,b). In the following development, we consider the interaction between the modal notions (Governatori et al., 2008). A simplified version of modal reasoning is used in an extension of the DL-MAS (Pham et al., 2008a).

2

Overview of multi-agent systems

Multi-agent systems have been investigated since the 1980's to investigate complex distributed problems where the approach of a single agent is not feasible, because of the limitation on knowledge and computing resources of one agent. Agents in a multi-agent system are required to interact with each other to obtain the solutions to the problems, despite the fact that they pursue their own goals and autonomously execute their tasks. Their interactions can be either to collectively work on the problems or coordinate their activities or share information. According to Sycara (1998), typical characteristics of multi-agent systems are:

- Each agent has partial information of its environment and a limited capability to solve the problem; thus the agent has a limited viewpoint.
- There is no system global control.
- Data is spread across the agents in the system.

- Computation is asynchronous.

This chapter intends to sketch some basic elements of the multi-agent systems. To start with, we briefly introduce the concept of intelligent agents, which is the basic building block for any multi-agent system. Next, we elaborate the interactions among the autonomous agents. The ability to interact distinguishes the agents from other computing entities and enables these agents to investigate complex problems. The chapter ends with the different methods for modelling multi-agent systems, in particular, capturing the interactions.

2.1 Intelligent agent

The notion of intelligent agent has attracted a great interest of researchers and developers in the field of computer science in the past decades. Recently, the concept of intelligent agent has offered a promising recipe for building highly abstract and complex systems like semantic grid systems. Arguably, one of the most important abilities of an agent is that an agent can work autonomously in a dynamic environment. In other words, an agent can accomplish a designed task without human's intervention. Agents can realise for themselves what to do on behalf of their owner to fulfil their allocated tasks, and more interestingly to cope with changes in their working environment.

2.1.1 Agent definition

Interestingly, there is no single definition of an agent. Depending on the application domains and the functionality of the agents, there are several types and definitions of agents, such as, interface agents or reactive agents. Perhaps, the definition of Wooldridge and Jennings (1995a) is the most well-known, *'An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment to meet its design objectives'*.

It is a non-trivial task to evaluate whether the behaviour of an agent is as intelligent as a human being. To some extent, it would be helpful to investigate the characters of the agents to clarify this 'magic' concept.

2.1.2 Agent attributes

According to Wooldridge and Jennings (1995a), intelligent agents should exhibit the following abilities:

- Reactivity: intelligent agents are able to perceive their environment, and respond in a timely fashion to changes that occur in it, to satisfy their design objectives.
- Pro-activeness: intelligent agents are able to exhibit goal-directed behaviour by taking the initiative to satisfy their design objectives.
- Social ability: intelligent agents are capable of interacting with other agents (and possibly humans) to satisfy their design objectives.

These attributes would not be too difficult to achieve if the agents operated in a static environment or the agents could have complete information about their world. In fact, a number of computer programs can satisfy these three attributes. For example, event-driven programs can interact not only with a human, but also with other programs to accomplish their goals. However, if the context of the programs changes, the designers are likely to restructure these programs to cope with the changes. To maintain autonomy, the agents should deliberate and then take a reasonable action to react to a change in their working environment or from other agents. It is also important that the action should be performed at an appropriate time with respects to the agents' objectives.

2.1.3 A conceptual model

The agent type we are interested in is that of rational agents, because of their abilities to reason about their working environment and also about their actions to change their environment.

The conceptual model of a rational agent is depicted in Figure 2.1. Perhaps, a hardware robot would be a very good example to describe the operations of the agent. Every robot can be equipped with sensors and actuators. Correspondingly, each individual agent has input and output modules that allow the agent to perceive information about the environment and to perform an action in response to a change in the environment. To reach a certain level of

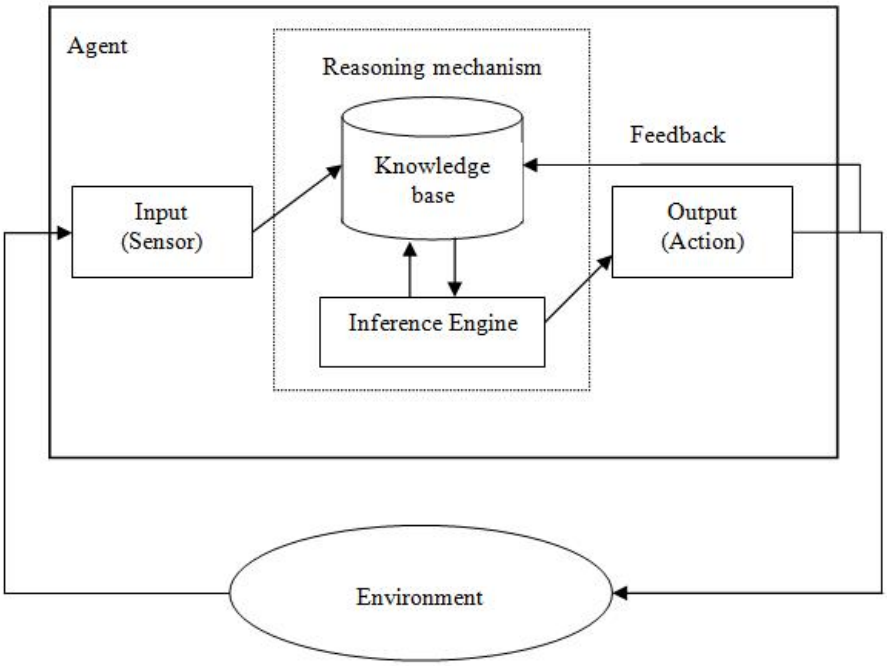

FIGURE 2.1: Conceptual model of rational agent

autonomy, the agent should consider the actions to be executed. In an abstract way, a decision could be achieved by passing perceived information through the reasoning mechanism, which is composed of an inference engine and a knowledge base. The knowledge base can be built by the designer and/or by accumulating perceptions from the environment. For agents with a learning capability, their knowledge base can be enlarged by feedbacks from the environment after their interactions.

As can be seen from the conceptual model, the working environment of an agent may be changed not only by some external events, but also by actions of the agent itself. In addition, the agent can obtain a partial picture of the environment, possibly because of its interests and/or the limitations of sensors. Those factors impose difficulties on the decision-making process of the agent. Actually, the degree of the agent's rationality mainly depends on the quality and the sophistication of this process.

2.2 Interactions of the agents

Interaction among the agents is an interesting and important phenomenon in multi-agent systems. Through interactions, agents can influence each other's decision-making process and contribute to knowledge evolution of the group. In this section, we first sketch out the essential concepts of the interaction among the agents. Next, we introduce the notion of coordination and we discuss the role of communication and possible constraints on the agents' interactions.

2.2.1 Interaction definition

Interaction is a very distinct and frequent behaviour of a living group. Members in a group can interact with one another to learn the way to the source of food or learn about the existence of dangers. On the one hand, this behaviour allows the group to collect individual resources or to have the capacity to fulfil tasks that the individual members cannot afford to achieve. On the other hand, interaction allows the knowledge within a group to evolve by passing information from member to member.

The interacting ability of autonomous computational entities is a key topic in the field of distributed artificial intelligence, in particular, multi-agent systems. This ability of the individual agents allows the multi-agent systems to model and tackle very complex problems, such as air traffic management, and distinguishes multi-agent systems from other systems in the distributed artificial intelligent field. Abstractly, interaction among agents can be considered as a sequence of actions executed by the agents to influence one another in future behaviours (Ferber, 1999; Weiss, 1999, pp 2–3). These actions can impact directly on the working environment observed by the agents or change the 'mind' of involved agents as in information exchange.

According to Ferber (1999); Huhns and Stephens (1999), the motivations of an interaction between the agents can result from dependency among the goals, resources and the capacity of these agents. During interactions, the actions are not randomly performed by the agents, but are deliberated based on their understanding about other agents for obtaining a desired state. However, the goal state can be private to a single agent and is not necessarily shared by all the other agents participating in the interaction.

2.2.2 Coordination

Coordination is an important type of interaction that has attracted much effort and the intentions of the multi-agent systems research community. This special activity among agents is either to collect more knowledge about their working environment for pursuing their goals or to gather resources and capacity from other agents to execute their tasks. Essentially, coordination can be considered as a sequence of actions involving a group of agents such that individual agents can coherently behave as a unit (Huhns and Stephens, 1999; Nwana et al., 1997).

Any individual agent in a multi-agent system has its own perception about the working environment and other agents. Also, each agent is equipped with a certain amount of knowledge and resources for executing its tasks. Therefore, coordination among these agents is needed to achieve the following main goals (Jennings, 1996; Nwana et al., 1997):

- *Avoiding anarchy or chaos*. Because individual agents have only a partial view of the environment, their goals and knowledge are likely to conflict with others. To settle conflicts, agents arrange bargaining or concessions of their knowledge, resources and competence. Without a global view of the system, these activities can result in a chaos and failure to achieve common goals.

- *Fulfilment of global constraints*. Usually, a group of agents maintain global constraints that require any single agent's compliance. Agents must be aware of these constraints and coordinate their activities to balance between individual interests and the success of the group.

- *Collecting distributed knowledge or resources*. In multi-agent systems, individual agents may have different capabilities and specialised knowledge. Moreover, they may have different sources of information, resources, reputation levels, responsibilities etc. The goal's achievement is beyond the knowledge and competence of any single agent.

- *Tackling dependencies between agents' actions*. Although the goals of an agent are independent from the other agents' goals and that agent may not be aware of the others' goals, its actions may be influenced indirectly by the others in some situations. The agents in a group have to coordinate their activities to avoid conflicts.

There are two fundamental types of coordination namely cooperation and competition (Huhns and Stephens, 1999). Cooperative agents work as a team by sharing their knowledge and resources to accomplish common goals that the single agents cannot individually attain. The agents will succeed or fail together. In contrast, competitive agents work against one another, because of their conflicting goals. An individual agent tries to maximise its own benefit at the expense of the other agents. Hence, the success of one agent results in the failure of the others.

Agents playing a chess game is a typical example of competitive interaction where each agent tries to increase its own utility. At each move, agents compute the potential gain from that move by pondering the response of the competing/opponent agent. However, in a negotiation situation, interacting agents work together to maximise the utility of the system. Typically, the resources of the system are limited and should be shared among the agents. The sole possession of the resources can result in chaos and failure of the whole system. Consequently, agents should balance between their own interests and that of the system. Often, agents convince each other of their resource requirements by providing arguments.

Coordinating agents have to tackle the problem of managing dependencies between agents' activities (Omicini and Ossowski, 2003). In particular, an agent should ponder the dependency of its planned tasks and resources with those of other agents to attain its goals with or without conflicts with the others. Successful coordination introduces coherent behaviour between the agents without an explicit global control of an individual's actions (Huhns and Stephens, 1999). Thus, via coordination, agents are capable of achieving the global constraints and efficiently distributing their knowledge and resources (Nwana et al., 1997). Agents without the ability to coordinate may end up with wasted efforts and resources and fail to achieve their desired goals (Durfee, 2004).

2.2.3 Communication

One important property of agents in a shared environment is the ability to communicate with one another. During communication, agents can gradually construct their models of one another, thus reducing uncertainties about themselves, their world or their goal (Durfee, 1999).

As a result, communication is indispensable for the coordination of an agent's actions and behaviours.

Communication must be defined at several levels in accordance with the competence of the agents. A simple communication mechanism allows agents to exchange messages to have more information about the working environment and other agents. In a more complex mechanism, agents engage in a dialogue to express their interests. Therefore, the mechanism extends the perception of the agents by understanding the meaning of exchanging messages. The formal study of communication mechanisms has to deal with structuring messages from a set of symbols and retrieving meanings of these messages (Huhns and Stephens, 1999).

A public announcement is an important phenomenon in communication among agents. Once a message is trustfully and publicly declared within a group of agents, individual agents not only understand what they themselves know and do not know, but can also infer the knowledge and ignorance of the other agents. Furthermore, public communication is a method of establishing common knowledge within the group, which is most critical for coordinating the agents' actions. A formal model of public communication without common knowledge is proposed by Plaza (1989) and Gerbrandy and Groeneveld (1997) independently. Baltag et al. (1998) present the set of axioms for the public announcement logics with common knowledge. (van Ditmarsch, 2005, Chapter 4) provides a detailed investigation of the formal model for public announcements.

Two well-known languages for agent communication are KQML (Labrou, 1997) and FIPA-ACL (http://www.fipa.org/repository/aclspecs.html). The first stands for Knowledge Query and Manipulation Language while the second is Foundation for Intelligent Physical Agents: Agent Communication Language. The KQML is a message-based language for agent communication originally devised as a means for exchanging information between different knowledge-based systems. Because of its simplicity, the KQML is the most widely implemented and used in the agents' community.

FIPA-ACL is proposed by The Foundation for Intelligent Physical Agents (FIPA) to overcome limitations of the KQML. The language is derived from Arcol (Sadek, 1991) and uses a quantified multi-modal logic as its underlying logic. Despite its expressive power, it is very difficult to implement the full-features of the FIPA-ACL, which limits the popularity of the

FIPA-ACL in agent communities.

2.2.4 Interaction constraints

In human society, individual members adjust their behaviours upon encountering actions from other members and vice versa. They also perceive that their counterparts can react in a similar manner. Through interacting with the members of the society, an agent can discover co-relations and constraints between the individuals' activities and dynamically create a template of expected behaviours to avoid chaos and waste the resources of the society. That provides a basis for the norms and social laws of human communities, which plays a critical role in coordination (Lewis, 1969). The work of Castelfranchi (1995) also recognises the role of the individuals' commitments to the group activities.

In a similar fashion to human beings, agents' actions are not simply driven from/by their own interests but also the interests commonly recognised by a community of agents. Essentially, a community establishes norms as patterns of behaviours and places constraints on the actions of its members. In some situations, an expected course of actions is enforced by punishing the violating members. The reputation and credit of these members can be decreased from the community's point of view. Aware of norms and social conventions, agents adjust their behaviours towards interests common to the community. Therefore, conflicts between agents can be reduced and eliminated. A number of authors[1] acknowledge the necessity of social laws, conventions and norm-like mechanisms for a robust and efficient coordination in multi-agent systems. Without the specification and enforcement of the standard behaviours in the community, individual agents work inefficiently and may not able to fulfil the simplest tasks because of the conflicts and interference from other agents (Shoham and Tennenholtz, 1997).

There are three views of norms, norms as constraints on behaviour (Conte and Castelfranchi, 1995), norms as goals (Rao and Georgeff, 1995), and norms as obligations (Vázquez-Salceda et al., 2005). The simplest form of norms is the specification of the activities which that requires the agents to strictly comply with Shoham and Tennenholtz (1992, 1997). Alternatively, norms

[1]Cohen and Levesque (1990); Jennings (1993); Jennings and Mamdani (1992); Kinny and Georgeff (1991); Shoham and Tennenholtz (1992)

can be considered as a filter for goals generation and selection in the reasoning process of an agent. Norms themselves do not directly specify goals for agents to attain, but the criteria that the agents' behaviour should follow. As a result, the norms restrict the agents on possible options for their goals (Castelfranchi et al., 2000).

In general, norms shape the behaviour of an individual agent and place constraints on the goals. A norm can be represented as the obligations and rights associated with an individual member within a community of agents. It is not necessarily to enforce an agent to follow the community's obligations and rights. As an autonomous entity, an agent can ponder whether or not to comply with norms at different granules depending on its understanding of the actual situations (Alonso, 2004). On the one hand, an agent has a strong temptation to override its obligation to attain its goal rather than reconsidering its intention. On the other hand, an agent can avoid adopting its obligations to eliminate bad results caused by the incompleteness of the norms (Castelfranchi et al., 2000). Therefore, a deviant behaviour can be accepted in some situations (Dignum, 1999).

2.3 Multi-agent system models

A multi-agent system can be considered as a set of interacting agents. From this perspective, modelling a multi-agent system starts with the problem of a single agent. The main challenge for modelling multi-agent systems is to determine how an agent settles conflicting interests between itself and other agents, and also the conflicting information of a different view point. In a reasoning model, the knowledge about other agents in the system influences what an agent believes and, consequently, the actions executed by this agent.

For the rest, we present an overview of the different methods of modelling multi-agent systems, including the logical model, mental-attitude model, computationally-grounded model and the game-theory model. In these models, we focus on the expressive capability and computability features.

2.3.1 Logical model

The main idea of the logical approach for modelling agents is to take advantage of the logical tools in representing the working environment and desired behaviours of the agents (Russell and Norvig, 2002; Wooldridge and Jennings, 1995a). The designer of a multi-agent system focuses mainly on specifying what agents know about the environment possibly including other agents. Thanks to the semantics of the logics, the designer is free from constructing the mechanism of the system or inventing an algorithm for individual agents. Another advantage is that the designer can check the coherence of the agents' behaviour against the specification of these agents before agents actually go online.

The basic construction of a logical agent includes a knowledge base containing a set of logical statements describing the environment and a set of deduction rules representing its decision-making process. The decision-making process of an agent is triggered to determine an appropriate reaction, whenever the agent perceives a change in the environment. When multiple agents are involved, the knowledge about the environment also includes what an agent knows about other agents (Kowalski, 2001). In other words, other agents are considered as an integral part of the working environment. From the view of an agent, the execution of its actions can significantly influence the perception of other agents and, consequently, change the behaviours of the group. Therefore, the effect of the agents' activities is more sophisticated and complex.

To avoid chaos situations and to achieve coherent behaviour in the group, the logical model has to settle individual and collective agent semantics, for example, by introducing global constraints on the agents' behaviour or conventions in the group (Torroni, 2004).

Despite promising the capability of the logical approach, there are several difficulties that are not trivial in remedying. On the one side, representing all the properties of the dynamic and real-world environment is a challenge for the logical approach, especially when the agents are considered as part of the environment. On the other side, the computational complexity of the inferential process prevents an agent from reacting effectively and efficiently in a timely manner.

2.3.2 BDI model

One important attribute of a rational agent is the capacity to 'initiatively achieve' the goal without human intervention. This attribute imposes agent-modelling methods to capture the concept of the agents' behaviour and to provide tools for understanding and predicting the behaviour. A well-known and successful approach, the BDI model, (where BDI stands for Beliefs, Desires and Intentions), is inspired by human attitudes towards actions. At any time, the states of an agent are characterised by a tuple of mental attitudes including belief, intention and desire (Rao and Georgeff, 1991, 1995).

The BDI model is inspired by the philosophical investigation by Bratman (1987) on human practical reasoning. The beliefs of an agent essentially represent its perceptions during the interactions with the environment. The desires or goals of an agent have long-term values driving an agent to act. Basically, a goal is a state of the environment the agent wants to achieve. To fulfil the goal, an agent can derive several sub-goals or alternatives. Once the agent commits to one of these alternatives and provided that it does not conflict with the goal, the alternative can be considered as intention. Very often, an intention is likely to lead to an action by the agent. Also, an intention can be regarded as a short-term goal which constrains the agent's reactivity. In Rao and Georgeff (1991), these mental attitudes are captured by a Kripke structure (Kripke, 1963) while the dynamic activities of the agents are represented by a branching time temporal logic (Allen and Jai, 1988).

In general, given a goal, an agent generates several options (intentions) such that the goal can be attained. Based on the current knowledge (state of the environment), an agent may decide to commit itself to one alternative providing that this alternative does not conflict with the agent's goals. From this point, further actions will be derived by the chosen alternative until the alternative is fulfilled. There are various problems attached to the above process. The intentions could be inconsistent with the goals or with beliefs. In some cases, it is impossible to fulfil the intention to which an agent has committed itself. Therefore, an agent should balance between overruling the conflicts and reconsidering its goals to maintain a reasonable level of rationality.

The concepts of the BDI model have been implemented for multi-agent systems in PRS

(Georgeff and Lansky, 1987), dMARS (D'Inverno et al., 1998, 2004), Jack (Age, 2001) and AgentSpeak (Rao, 1996). However, these implementations have to relax the features of the BDI model because of the computational tractability.

Extensions of the BDI architecture arise from studying the interactions between the autonomous agents. Agents are required to balance their individual goals and the goals shared by their group. The agents' behaviour is not simply constrained simply by their internal intentions and desires, but also by their commitments to the group (Castelfranchi, 1995), common conventions and norms within the group (Broersen et al., 2001; Castelfranchi et al., 2000; Cavedon and Sonenberg, 1998; Dignum et al., 2002; Lacey and Hexmoor, 2003).

The BDI model is one of the most used models in the agent research community . This model provides an insight into the decision-making process of an agent. Furthermore, the model facilitates building agent systems, because of its clear definition of agent functionality. Despite the expressive representation of the agents' rationality, the full-featured implementation of the BDI model is still ongoing research because of the computational cost of representation and reasoning with modalities. The limitations of the formal construction of the BDI model raise the need for a method of agents' specification that can be computed by a computer program. That is the main motivation for the birth of the computationally-grounded model presented in the following section.

2.3.3 Computationally grounded model

Essentially, the model of a multi-agent system is not only required to clearly explain (model) the behaviour of the agents in the system, but also it needs to be able to 'compute' this behaviour. It is also critical that a model can be implemented and assessed by the designer. The interpreted system originated by Fagin et al. (2003); Halpern and Fagin (1989) is the very first model allowing the use of the computational properties of a computer program to give meanings to the formulas in the model. Thus, the interpreted system is a computationally-grounded model (Wooldridge, 2000).

According to the interpreted system approach, a multi-agent system may contain n different agents. Each agent has its own states (local states from the system's viewpoint), which represent

the information being accessed by the agent. Because the agents have to operate in some kind of environment, it is necessary to have an environmental state. In fact, this is the external information related to running the agents. A global state of a system with n agents is a tuple of the form $(s_e, s_1, \ldots, s_n)$, where s_e is the state of the environment and $s_i | i = 1 \ldots n$ is the local state of agent i.

At run-time, the system changes from one state to another. Then, a run is identified as a function mapping from time to global states. The initial global state is presented by $r(0)$, the next one by $r(1)$ and so on. A referenced point to a global state of the system can be represented as a pair of run and time (r, m) such that

$$r(m) = (s_e, s_1, \ldots, s_n)$$

At the time m, the references to a state of the environment, r_e, and a state of agent i, r_i, are represented respectively as follows:

$$r_e(m) = s_e$$
$$r_i(m) = s_i |\ i = 1 \ldots n$$

An interpreted system $\mathscr{I}$ consists of a pair $(\mathscr{R}, \pi)$, where $\mathscr{R}$ is a system over a set G of global states and π is an interpretation for formulas in Φ over $\mathscr{G}$, which assigns truth values to the primitive propositions at a global state. The function π is defined as

$$\forall p \in \Phi \wedge s \in \mathscr{G}\ \pi(s)(p) \in \{true, false\}$$

It is noticed that the global state s can be represented by a pair of run r and time m.

To determine the knowledge of the agents, the interpreted systems $(\mathscr{R}, \pi)$ can be linked to a Kripke structure as: a set of worlds S representing states; Evaluation function π; Binary relations $\kappa_i | i = 1 \ldots n$ over S.

If there are two existing global states s and $s^{'}$ for a relation κ_i such that $s_i = s_i^{'}$ then agent i can conclude its knowledge. Also, this condition can be rewritten as $r_i(m) = r_i^{'}(m^{'})$. Informally speaking, agent i cannot distinguish between two states s and $s^{'}$. An equivalence relation has been established between the two states. For agent i the equivalence relation (reflexive, symmetric and transitive) between those states s and $s^{'}$ is represented by $s \sim_i s^{'}$ or $(r, m) \sim_i (r^{'}, m^{'})$. With respect to the Kripke structure, agent i can derive knowledge by using the following formulas:

$$\text{Iff } (\mathscr{I}, r', m') \models \varphi \ \forall (r', m') \text{ such that } (r,m) \sim_i (r', m') \text{ then } (\mathscr{I}, r, m) \models K_i\varphi$$
$$\text{where } \varphi\text{: a formula in } \Phi.$$

Common knowledge can be modelled based-on the operator E, which means everyone knows:

$$\text{Iff } (\mathscr{I}, r, m) \models Ki\varphi \text{ for } i = 1 \ldots n \text{ then } (\mathscr{I}, r, m) \models E\varphi$$
$$\text{where } E\text{: knowledge in the system having } n \text{ agents.}$$

Thus, common knowledge about φ, $C\varphi$, is defined as

$$\text{Iff } (\mathscr{I}, r, m) \models E^k\varphi \text{ for } k = 1, 2, \ldots \text{ then } (\mathscr{I}, r, m) \models C\varphi$$

Agents perform their actions based-on their local states as specified by their protocols. Essentially, a protocol is a function that maps an agent's local states to its possible actions. An interaction between agents could be regarded as a *joint action* that is identified by a joint protocol. Basically, a *joint protocol* consists of every individual agent's protocol with respect to the environment. The effect of a joint action is captured by the transform function τ mapping a global state to new one.

$$\tau(a_e, a_1, a_2, \ldots, a_n)(s_e, s_1, \ldots, s_n) = (s_e', s_1', \ldots, s_n')$$
$$\text{where } a_e : \text{ action that changes the environment}$$
$$a_i : \text{ action of agent } i$$

Interpreted systems have introduced a novel method of representing multi-agent systems by using set of states of agents and the environment over linear time. These states can be interpreted as the computational properties of a program describing the behaviour of the agents in the system. The knowledge of an agent is modelled as states that are indistinguishable over the running steps. The agents' actions are motivated by the knowledge acquired during the course of the interactions. The success of the interactions is achieved, provided that the common knowledge states are recognised by agents.

Wooldridge (2000) extends the interpreted system model by considering the visibility function of the agents. This function reflects the individual agents' perception of the working environment. Essentially, given the actual environmental state, the individual agents can have

different evaluations. Thus, the knowledge states of the agents depend on the transparent levels of the environmental states. Similarly, the KBC model (Su et al., 2005) captures the agents' subjectivity about the perceived information from the environment including the visible and invisible parts. However, the KBC favours the internal modelling of the agents' knowledge states. An agent can *K*now or *B*elieve or feel *C*ertain about a piece of information. By observing the change in the environment, individual agents may not be aware of the imprecision of their sensor. Also, a single agent can speculate on inaccessible parts of the environment.

2.3.4 Game theory model

Game theory created by Neumann and Morgenstern (1953) models the economical behaviours of rational agents. During encounters with other agents, a rational agent tries to maximise the outcome of its actions by considering the decisions of the others in its reasoning process. The early work of Aumann (1976) provides the initial connections between game theory and agent studies. Over repeated interactions, an individual agent can perceive what the involved agents believe and discover the knowledge common to the agents; therefore, it can reason with these pieces of information. Naturally, multi-agent decision-making in strategic situations is resolved by the game theory (Osborne and Rubinstein, 1994). Representing multi-agent systems by game theory provides a meaningful and formal view for modelling the agents' interactions and predicting the agents' behaviours. Different types of interactions in multi-agent systems have been modelled because of the work of Genesereth et al. (1986); Rosenschein and Genesereth (1985); Rosenschein and Zlotkin (1994).

In what follows, we introduce an abstract representation of a multi-agent system corresponding to (Wooldridge, 2002, pp 105–128). In a multi-agent system, every agent has its own preferences and desires about how the world should be. The knowledge of the agents about the world can be represented by a set of states, Ω or outcomes.

$$\Omega = \{\omega_1, \omega_2, \ldots\}$$

Each state is associated with a real value showing how the agent prefers the state. According to this representation, states are distinguished by the associated benefit (utility value). Formally,

mapping states to the preference values is performed by the utility function:

$$U_n : \Omega \mapsto \Re$$

where n : agent n;

$\Re$: set of real number

An agent n can perform an action a_i from its action set $\mathscr{A}_n$. Then, this leads to a transition in the agents' state Ω and produces a new outcome. Since each agent can decide its own action autonomously, different combinations of agents' actions result in different outcomes.

$$(a_1, a_2, \ldots, a_N) \mapsto \omega^{'}$$

where a_i: an action performed by agent mechanism based i

The agent evaluates the benefit of an action using its decision exclusively on the actions but not the knowledge about the environment. The higher utility an action brings back, the more likely that the action will be selected. To acquire the optimal benefit, an agent should ponder outcomes of all possible actions with regard to other agents.

During the course of the actions, an individual agent can follow an action set $\mathscr{A}_n$ (strategy) that can produce an optimal outcome. That is, the agent can possibly obtain the maximum benefit from the sequence of actions. In the best case, one agent could discover such a strategy that dominates the others by producing better utility values for every single action. However, it is not always possible to detect such strategy or, simply, this strategy does not exist. Therefore, the solution can be the strategy such that none of agents in the group can improve its benefit by changing its own strategy. In the case that all agents follow that strategy, an equilibrium occurs in the system and is named the Nash equilibrium (Nash, 1950). This equilibrium has been proved to be very important to predict the behaviours of rational agents.

The game-theory model is traditionally based on the assumption that every agent in the system shares the common knowledge about strategies (possible sequences of actions) and the utility scheme (benefit/cost for actions). Also, an agent does not have any limitation on its computational resources to determine the optimal outcome of its actions. These assumptions are not always practical in multi-agent systems and raise concerns of computer scientists on the Nash equilibrium (Halpern, 2008) despite the refinements, such as that of Osborne and

Rubinstein (1994). In general, it is not adequate to simulate human-like preferences with a simple order over states of 'affairs', especially in multi-criteria preference (Russell and Norvig, 2002). Therefore, the game theory model has limitations in capturing the cooperative behaviour of agents.

However, the strategic reasoning of the game theory is most desirable in the study of multi-agent systems. Pondering the effect of individual actions on other members is very critical. There are several works that integrate the game theory in their logical formalisms. Boella and van der Torre (2006, 2007) consider whether or not agents adopt their obligations as a *violation game* played by agents within a community. Meanwhile, Roth et al. (2007) examines the effect of exposing the evidence in a legal game that enables the players to construct their winning strategy. The approach for the combination is alternating-time temporal logics.

Alternating-time temporal logics, ATL, is first introduced in Alur et al. (1997) and extended in Alur et al. (2002). The ATL is one method incorporating the game-theoretic evaluation into the quantification of the time-branching computational paths that are considered as the possible outcomes of the interactions of the players in the concurrent game settings. The ATL provides a natural tool for specifying multi-agent systems. Several useful concepts, such as safety, liveness and the fairness of an action, can be formally specified by the ATL operators. In multi-agent systems, the ATL can be used to describe, synthesise and verify the general properties of the system. Nevertheless, the ATL approach requires complete information on the game that an individual agent totally knows about the utility function used by the other agents. In other words, all agents have the same configuration of the game. This requirement is not realistic in many practical situations. In addition, the computational cost for the ATL is very expensive. The complexity of the satisfiability problem is EXPTIME-complete (van Drimmelen, 2003; Walther et al., 2006).

2.3.5 Discussions

It is undeniable that game theory has proved to be very useful for modelling the agents' behaviour by using formal and sound notions that allow heuristic and clear-cut experiments. The

game theory models well the interactions among the agents and provides a meaningful explanation of the motivations of the agents towards their interactions. However, the game theory has serious limits when representing multi-agent systems, imposing shortcomings on the agent's reasoning about the interactions, especially cooperation.

According to game theory, the interactions among the agents are represented as a set of states, where every state is associated with a utility or benefit. To obtain an optimal action, the agents should investigate all possible actions taken by every agent in the system. Then, the size of the state space is exponential to the number of agents and possible actions in the systems. It is really impractical to represent a large system. Moreover, this representation cannot capture all aspects of an agent's information (Halpern, 2003), such as the beliefs or goals or strategies followed by other agents. In other words, the game theory only captures the knowledge about rewards, which agents can acquire from their actions. The agent turns out to consider the benefit/cost instead of the reasoning about the goals, preferences and motivations. As a result, game theory does not guarantee a true cooperation, because a satisfactory notion of the cooperation needs the modelling of the agent's cognitions, and especially of its goals (Castelfranchi, 1997).

Differing from game theory, the logical methods can describe the cognitive properties of the agents by using the corresponding modalities. Therefore, these methods ensure rich and expressive representations for different kinds of agents' information, such as beliefs, goals and intentions. The meaning of the representation is determined by the possible world semantics, where different pieces of agents' information are formalised by a set of possible worlds with an accessibility relation holding between them. Despite the fact that possible world semantics is well studied and can well capture the agents' knowledge and uncertainty, it is not yet clear how to define the mappings from the abstract accessibility relations, used to characterise the agents' state, to concrete computational models. Regarding this issue, Wooldridge (2000) has showed that interpreted systems are computationally-grounded. That is, given an interpreted system $\mathscr{I}$, characterised by a set of states over linear time, and a description system Φ, possibly verified by $\mathscr{I}$. If system I constitutes a set of computer programs then system Φ is known as computationally-grounded. This property of interpreted systems is very important from the

engineers' view. It is possible to build a computer system from the specifications of the conceptual model of the multi-agent system. Furthermore, these specifications can be mapped to logical properties and they can be verified.

The complexity of the logical model including the BDI model depends on the types of logics being used to model agents' actions and states of the environment. To obtain the tractability of the agents' rationality, there are trade-offs between expressive capability and computational complexity (Dantsin et al., 2001). Regarding this issue, the cost of computing the equilibrium among the agents' actions amounts to the space of possible states. Essentially, the problem can be seen as a search to find an optimal path through all possible combinations of the agents' state. That is a NP-hard problem.

3

Logics for multi-agent systems

The concept of an agent facilitates the representation and manipulation of large and complex systems that are composed of interacting and evolving computational entities. However, this requires a formal tool that eases the description and the control of the behaviour of the individual agents as well as a group of agents. Logical approaches are most suitable to tackle these requirements, because they are unambiguous and flexible in expressing specification. Furthermore, the logical approaches are endowed with computational models for verifying the specification.

In this chapter, we first provide an overview of the classes of logics to represent different aspects of the agents' knowledge. At the end, we present logic programming approaches for implementing the multi-agent systems.

3.1 Logics as knowledge representation

As an autonomous computational entity, an agent performs an action based on what the agent knows about its working environment and the other agents. In addition, by taking an action, an agent tries to fulfil its desires and satisfy the constraints with other agents. During the interaction with the working environment and other agents, an agent can obtain more information that enriches its knowledge base. However, the knowledge base is just a model (often incomplete) of the actual world. That supposes an agent has to confront the problem of the emerging information in the reasoning process.

In this section, we first present the modal logics that have been proved to be useful and powerful in representing the attitudes of the agents towards actions. Using modal logics enables an agent to differentiate the epistemic states of information (belief and knowledge). An agent can reason about the change in its epistemic structure and that of the other agents. Furthermore, modal logics can capture concepts such as conventions and norms, which put some levels of constraints into the behaviour of individual agents. Finally, non-monotonic logics directly approach the problem of the incompleteness of the agents' knowledge.

3.1.1 Modal logics

A modal logic, originally invented by Lewis (1918), qualifies the truth of the modal expressions, such as possibility and necessity p (p is a statement). Informally, modal logics deal with the logical expressions that could be true (possibility) in some cases or always true (necessity) in every case, such as the beliefs and knowledge of an agent. An understanding of modal logic is particularly valuable in the formal analysis of a philosophical argument, where expressions from the modal family are both common and confusing. To some extent, modal logic is similar to classical logic except for two new modal operators: $\Box$ for Necessarily; $\Diamond$ for Possibly. Typically, these operators relate to each other by:

$$\Box p \rightarrow \neg\Diamond\neg p \text{ and } \Diamond p \rightarrow \neg\Box\neg p$$

In addition to necessity and possibility, the modal operators are used to capture different notions, such as knowledge, change and obligation. In other words, necessity and possibility can have

different interpretations depending on the domain of interest. Logics for knowledge are known as epistemic logics and those for change are called dynamic logics. Logics for obligation are named as deontic logics. The designer of the modal logics should provide a method to interpret the properties of the modal operators. The properties of the modal operators can be investigated by a syntactical approach using modal axioms. Essentially, the modal axioms describe the modal formulas that are valid according to some criteria. The criteria of validity are often constructed by the designer of the modal systems that represent the interesting consequences of the systems. In the next section, we introduce essential axioms of the modal systems.

Modal axioms

Modal logics are equipped with typical axioms as below:

TABLE 3.1: Essential axioms of modal logics

Name	Axiom
K	$(\Box p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)$
T	$\Box p \rightarrow p$
4	$\Box p \rightarrow \Box\Box p$
D	$\Box p \rightarrow \Diamond p$
B	$p \rightarrow \Box\Diamond p$
5	$\Diamond p \rightarrow \Box\Diamond p$

These axioms are used as important seeds to build up a logical system. A composition of modal axioms provides a particular meaning for the modal operators. Therefore, different compositions create several logical systems. For example, epistemic logic is based on the $KT45$ axioms. Within the context of the agent systems, the T axiom can be interpreted as 'an agent knows what is true'. The 4 axiom (known as positive-introspection axiom) can be expressed as 'agents know about what they know' whilst the D axiom - agents 'believe in what they know'. In normative concepts, $\Box p$ means an agent ought to comply with the obligation p. But it is questionable that the agent believes in the obligation that an agent ought to adopt as represented by the 4 axiom. However, it makes sense that anything that is obliged is then permitted as by

the D axiom.

A logical system is to prove exactly the valid statements stable in the language describing the specifications of the system. The soundness and completeness are important properties of a logical system. That is, every statement proven using the logical system is valid and every valid statement has a proof in the system. Eventually, this is a powerful point for the logical approach to model a system in the sense that the correctness of the system's behaviour is always guaranteed.

Formal semantics for a logic system provide a definition of the validity by characterising the truth behaviour of the sentences in a system. The traditional truth tables, in the propositional logic, have limitations in the interpretation of modal logics. Simple true and false values could not fully represent the nature of the modality. Instead of truth tables, the semantics are represented by possible worlds. Possible worlds can be illustrated as a graph of points with directed edges. A point in the graph represents a world where a logic sentence is true. However, this sentence may be invalid in some other worlds. To validate the agent's arguments, the systems apply an evaluation function for each individual argument over the possible worlds. *necessity* p, denoted as $\Box p$, is valid in all possible worlds, while a *possibility* p, denoted as $\Diamond p$, is valid in at least one world. In the following section, we present the semantics of modal logics by using the Kripke possible worlds.

Kripke Structures

The semantics of modal logics can be interpreted by using the Kripke structure (Kripke, 1959). Basically, a Kripke structure M for a set of logic statements Φ is defined by a tuple of W, υ, R as follows:

$$M = (W, \omega, R)$$

where W : a non empty set

$$\upsilon : \upsilon(\omega, p) \mapsto \{True, False\} | \omega \in W \text{ and } p \in \Phi$$

$$R \subseteq W \times W$$

W is a set of worlds or possible worlds, υ is evaluation function, R are sets of binary relations over W. The proposition p in Φ is true at a world ω in a Kripke structure M is denoted as:

$$(M,\omega) \models p \text{ iff } \upsilon(p,\omega) = \text{ true}$$

Then, we have

$$(M,\omega) \models p \wedge q \text{ iff } (M,\omega) \models p \text{ and } (M,\omega) \models q$$
$$(M,\omega) \models p \vee q \text{ iff } (M,\omega) \models p \text{ or } (M,\omega) \models q$$
$$(M,\omega) \models \Box p \text{ iff } (M,\omega^{'}) \models p \; \forall\omega^{'} \text{ such that } (\omega,\omega^{'}) \in R$$
$$(M,\omega) \models \Diamond p \text{ iff } (M,\omega^{'}) \models p \; \exists\omega^{'} \text{ such that } (\omega,\omega^{'}) \in R$$

A Kripke structure can be represented by a directed graph. Each node in the graph represents a world in the Kripke structure and is labelled with propositions that are true in this world. The nodes are connected by directed edges. For instant, given a structure:

$$\begin{aligned} M =& (W,\upsilon,R) \text{ such that} \\ & W = \{s_1,s_2,s_3\}; \\ & \Phi = \{p_1,p_2\}; \\ & \upsilon(p_1,s_1) = \upsilon(p_2,s_2) = \upsilon(p_2,s_3) = \text{true}; \\ & R = \{(s_1,s_2)(s_1,s_3)(s_2,s_2)\} \end{aligned}$$

Then the graph for M is in the Figure 3.1. As shown in the graph, an agent knows p_1 is true at s_1 and p_2 is true at both of s_2 and s_3. However, the agent knows that $\Box p_2$ is valid at s_2 because of the self-loop link. That is all nodes connected to s_2 make true p_2.

There is correspondence between the relations of the Kripke worlds and the modal axioms. The behaviour of the modal systems can be characterised either by a set of modal axioms or by the attributes of the relations between the Kripke worlds. The correspondence is shown in Table 3.2. In the Table, x, y, and z represent worlds in a Kripke structure, whilst R is the set of relations between worlds.

By introducing concepts of possible worlds and defining the nature of relations between those worlds, Kripke structures provide a simple but efficient tool for the representation semantics of modal logics.

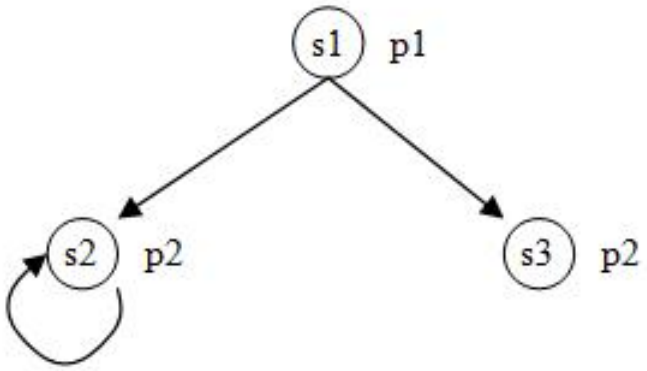

FIGURE 3.1: A simple Kripke structure

TABLE 3.2: Typical axioms with condition and type of relations

Name	Axiom	Condition	Relation attribute
T	$\Box p \rightarrow p$	$\forall x\, x\mathbf{R}x$	Reflexive
4	$\Box p \rightarrow \Box\Box p$	$\forall x,y,z\, x\mathbf{R}y \,\&\, y\mathbf{R}z \Rightarrow x\mathbf{R}z$	Transitive
D	$\Box p \rightarrow \Diamond p$	$\forall x \exists y\, x\mathbf{R}y$	Serial
B	$p \rightarrow \Box\Diamond p$	$\forall x,y\, x\mathbf{R}y$	Symmetric
5	$\Diamond p \rightarrow \Box\Diamond p$	$\forall x,y,z\, x\mathbf{R}y \,\&\, x\mathbf{R}z \Rightarrow y\mathbf{R}z$	Euclidean

3.1.2 Dynamic epistemic logic

Dynamic epistemic logic (van Ditmarsch et al., 2007) is an extension of epistemic logic, rooted in the work of Hintikka (1962), with dynamic operators that deal with the change of information. Dynamic epistemic logic is built upon the dynamic modal logic in the sense that the logic describes the change of information and provides epistemic operators to reason about information and its change. The concerns of dynamic epistemic logic are not only the truth condition of a formula, but also the changes of the knowledge states of those agents involved in the occurrence of that particular formula.

Communication among the agents is one of the most typical actions causing changes in the information states of the individual agents. Through communication, an agent does not simply verify the status of the content of a message, but also constructs a knowledge state about the other agents. The work of van Ditmarsch et al. (2007) investigates dynamic operators

for modelling epistemic actions. Public announcement is the simplest epistemic action, which allows establishing an information state being common to all involved agents (van Ditmarsch et al., 2007, Chapter 4). The language of public announcement logic is defined for a finite set of agents N and a finite set of atoms P as:

$$\varphi ::= p|\neg\varphi|(\varphi \wedge \psi)|K_n\varphi|C_B\varphi|[\varphi]\psi$$

where:

- $p \in P$, $n \in N$, and $B \subset N$ are arbitrary.
- $K_n\varphi$ indicates that agent n knows about φ.
- $C_B\varphi$ denotes that a group of B agents commonly know about φ.
- $[\varphi]\psi$ means that after announcement of φ formula ψ is true. It is noticed that, in formula $[\varphi]\psi$, operator $[\varphi]$ is $\Box$ modal. Owing to the partiality of the announcements, the interpretation of $[\varphi]\psi$ is identical to 'after every announcement of φ, ψ is hold'. The dual of $[\varphi]$ is $\langle\varphi\rangle$.

An epistemic model is a Kripke model where all accessibility relations are equivalence ones. Formally, an epistemic model $M = (S, \sim, V)$ contains a domain S of states (or worlds), accessibility $\sim$: $N \to \mathbf{P}(S \times S)$, where each $\sim_n$ is an equivalence relation, and a valuation V : $P \to \mathbf{P}(S)$. For $s \in S$, (M,s) is an epistemic state. Given two states $s, s' \in S$, $s \sim_n s'$ denotes that agent n cannot distinguish between s and s' based on its information. The group accessibility relation $\sim_B \equiv \bigcup_{n\in B} \sim_n$ is defined by the union of these of individuals in the group. $\sim_B$ is used to represent the common knowledge within a group.

The semantics for public announcement logic is defined as:

Iff $s \in V(p)$ then $(M,s) \models p$
Iff $s \not\models \varphi$ then $(M,s) \models \neg\varphi$
Iff $M,s \models \varphi$ and $M,s \models \psi$ then $(M,s) \models \varphi \wedge \psi$
Iff $\forall t \in S : s \sim_n t$ implies $M,t \models \varphi$ then $(M,s) \models K_n\varphi$
Iff $\forall t \in S : s \sim_B t$ implies $M,t \models \varphi$ then $(M,s) \models C_B\varphi$
Iff $M,s \models \varphi$ implies $M|\varphi,s \models \psi$ then $(M,s) \models [\varphi]\psi$

The model $M|\varphi$ is determined as an epistemic model $(S^{'}, \sim^{'}, V^{'})$ where

$$S^{'} = [\![\varphi]\!]_M = \{s \in D(M) | M, s \models \varphi\}$$
$$\sim^{'}_n = \sim_n \cap ([\![\varphi]\!]_M \times [\![\varphi]\!]_M)$$
$$V^{'}(p) = V(p) \cap [\![\varphi]\!]_M$$

The dynamic modal operator $[\varphi]$ acts as an epistemic state transformer. Trustful and public announcement within a group of agents eliminates uncertainty states of information conveyed in the announcement. Hence, individual agents can establish their knowledge as epistemic states that are indistinguishable the individuals. After the announcement, the common knowledge is not directly obtained as a consequence of an announcement but by a derivation rule, such as:

If $\chi \rightarrow [\varphi]\psi$ and $(\chi \wedge \varphi) \rightarrow E_B\chi$ are valid, then $\chi \rightarrow [\varphi]C_B\psi$ is valid.

The dynamic modal operator $[\varphi]$ can be extended as an action or event to capture more complex epistemic actions rather than the public announcement. Individual agents obtain different information from the complex actions. That refines the accessibility relations while the model of the domain remains intact. Moreover, these complex actions can extend the domain of the model. An elaboration of extending $[\varphi]$ as epistemic actions and action model can be seen at (van Ditmarsch et al., 2007, Chapter 5 and Chapter 6).

In recent developments, dynamic epistemic logic incorporates the factual change (change of agents' knowledge) (van Benthem et al., 2006) and preference-based modelling of belief revision with dynamic modal operators (Aucher, 2004; van Benthem and Liu, 2005; van Ditmarsch, 2005).

3.1.3 Deontic logics

Deontic logic found by von Wright (1951) is a useful tool for modelling normative concepts such as permission and obligation in multi-agent systems (Hilpinen, 1971; Meyer and Wieringa, 1994). The intuition of normative concepts is to develop the ideal behaviour of agents. That is, to specify what an agent should be obliged to do in ideal situations. However, the actual actions of the agents can deviate from their ideal behaviour, possibly because of the actual situation.

Normative behaviour, its violations and possible sanctions, can be explicitly captured in deontic logic (Dignum, 1999).

Standard Deontic Logic (SDL) can be viewed as a branch of the modal logics, where the operators of obligation O and permission P are introduced. These two are modal operators whose characteristics are identified by the following axioms:

1. $Op \rightarrow Pp$
2. $Pp \rightarrow \sim O \sim p$ and $Op \rightarrow \sim O \sim p$
3. If $\rightarrow p$ then $\rightarrow Op$

The first axiom ensures the consistence of the system and shows what is obliged is then permitted. The second means what is permitted or obliged, then it is not possible to do the opposite. The first two axioms present the principle that if an agent is given an obligation to do a thing then this agent either has the permission to perform it or does not have to do the opposite. The last axiom can be interpreted as, if anything can be proved, agents cannot disregard this thing. That essentially means an agent should be realistic.

SDL has three properties as shown in Table 3.3. The first two properties are about the derivability of the logic regards to the conjunction whilst the last property represents possible violations in inference. SDL has been attributed several paradoxes in capturing normative con-

TABLE 3.3: Properties of Standard Deontic Logic

Weakening	$O(p \wedge q) \rightarrow Op$
And	$(Op \wedge Oq) \rightarrow O(p \wedge q)$
Violation	$p \wedge O\neg p$

cepts (see Hansen et al. (2007)). These limitations are due to the pragmatic nature of the norms. That is, normative statements are not simply true or false. Among those paradoxes, the contrary-to-duty (Chisholm, 1963) is very important in multi-agent systems, because it instructs an agent on the new obligation when the ideal behaviour is no longer reachable. There are several approaches to restoring the consistency of the normative statements in the case of violations, such as preference-based deontic logic Hansson (1990) or defeasible reasoning (McCarty, 1994; Ryu and Lee, 1993, 1997).

Deontic logic is a useful tool for knowledge representation when modelling violations and contrary-to-duty obligations. Deontic notions have been the basis for agent framework BOID (Broersen et al., 2001) and BIO (Governatori and Rotolo, 2008). The argument is that agents often work in a group, therefore, they have to consider the norms and obligations shared by the group (Dignum, 1999) and, thus, must reason about these constraints.

3.1.4 Non-monotonic logics

In multi-agent systems, an agent often perceives a partial picture of its working environment. In addition, an agent can receive new information from other agents. Therefore, what an agent believes can have some levels of uncertainty. Basically, the closure property of the reasoning process of an agent is not adequate to represent the incomplete and conflicting information with which an agent deals. An agent should alter its conclusion to match the new information. For example, a bird can generally fly, but a broken-wing bird cannot. The new information about the wing situation prevents an agent from concluding that the bird can fly.

Non-monotonic logics are symbolic approaches to confront the problem of uncertainty without any quantitative representation. These approaches formalise the plausible and commonsense reasoning (Morgenstern, 1999). The classical approach to non-monotonic logics is to extend classical logic by adding meta-information about the dynamic knowledge.

Default logic by Reiter (1980) contains two knowledge components, predicate logic formulas as in classical logic and default rules. The logic formulas represent what is always true, while default rules represent what is usually true, given some known conditions.

Definition 1. *A default rule δ is represented as*

$$\delta : \frac{\varphi : \psi_1, \ldots, \psi_n}{\chi}$$

where $\varphi, \psi_1, \ldots, \psi_n, \chi$ are predicate logic formulas, and $n > 0$. φ is called the prequisite*; $psi_1, \ldots, \psi_n$ the* justifications*; and χ the* consequent *of δ*

The default rule about 'a bird generally flies' is denoted as $\frac{bird(X):flies(X)}{flies(X)}$. The semantics of default logics is obtained by computing the extension of the knowledge base of an agent with default rules, provided that inconsistency does not occur. Because a default theory is constructed

from a set of predicate logic formulas, the expressive capability and computability are inherited from those of the predicate logic. From computational complexity, satisfiability problems in default logic are harder than those of the predicate logic (Antoniou, 1997, p 87).

Auto-epistemic logic firstly developed by Moore (1985) uses a single modal operator L to represent knowledge states of an ideal agent with perfect introspection. An agent can examine its own beliefs and knows about what it knows and it does not know. Given a total belief set E, $L\phi$ means an agent knows ϕ and $\phi \in E$. In contrast, if $\phi \notin E$, $\neg L\phi$ holds, this means the agent does not know about ϕ. The knowledge operator has the property $L\phi \leftrightarrow \neg L\neg\phi$ and can be nested.

The logic of circumscription by McCarthy (1980) formalises non-monotonicity by adding second-order axioms that limit the extension of certain 'abnormal' predicates. The predicate representing flying birds can be rewritten as: $\forall X(bird(X) \wedge \neg abnormal(X) \rightarrow flies(X))$

The above expression shows that if the bird is normal, then the bird can fly. If there is any evidence showing an abnormal bird then the predicate *abnormal(X)* is added into the system to prevent the conclusion of *flies(X)*.

All three approaches to non-monotonicity extend the predicate logic in different ways. Default logic augments predicate logic by adding default rules, while the auto-epistemic logic exploits the concept of possible worlds constructed from a predicate knowledge base of an agent and the knowledge modal operators. Circumscription uses second-order axioms and the minimal model to retrieve the semantics of formulas. In all of these approaches, the logic theory representing the knowledge of an agent essentially does not remove redundant information that is no longer valid. Consequently, agents have to recompute their theories when new information occurs.

3.2 Logic programming languages

Logic programming plays an important role in building up multi-agent systems, because it provides a declarative method and formal semantics. In this section, we introduce programming languages that are explicitly designed to incorporate the notions of rational agents. Essentially, these languages convey the idea of an intention-based agency. Answer-set programming is

not specialised for multi-agent systems. However, the language provides useful features for implementing agents, such as negation as failure, handling multiple views.

3.2.1 Agent-0

Shoham (1993) proposes a programming framework, AGENT-0, for agents where an agent can be described by its mental attitudes such as belief, desire, and intention. That is the behaviour of an agent can be explained by human-like properties and can be captured by a logic program. (Thomas, 1995) develops an interpreted language for programming those agents.

AGENT-0 models an agent by a tuple $CAN, B, COMM$ where CAN: a set of capacities; B: a set of initial beliefs; $COMM$: a set of commitment rules. The notions of capability, belief, and commitment are captured by a quantified multi-modal logic. Time reference is associated with every element in a logical formula.

Example 1.

$$CAN_A^j submit(paper)^{j+4} \Rightarrow B_B^j CAN_B^{j+5} review(paper)^{j+6}$$

The above expression is interpreted as if agent A at time j can submit $paper$ at time $j+4$ then at time j agent B believes that B at time $j+5$ can review $paper$ at $j+6$.

A commitment rule determines an agent's action depends on the condition of the incoming message and its mental situation. In AGENT-0, agents can perform communicative actions including 'inform', 'request', and 'unrequest'. Inform messages convey information among agents and consequently update their belief base whilst request and unrequest messages change the commitments of the agents. The update operator is rather primitive and limited by restricting the representation of logical sentences (no logical connectives other than negation and no nested modalities) so that the operator can be tractable. Although the link between the logic and programming language is not formally defined (Wooldridge and Jennings, 1995b), the idea of AGENT-0 has been adapted by the research community and inspired many other languages for programming rational agents.

3.2.2 AgentSpeak

AgentSpeak(L) by Rao (1996) provides a high-level programming language that systematically adopts the BDI principles. The language of AgentSpeak(L) extends that of the first-order logic with events and actions by integrating the notions of beliefs, goals and intentions. These notions are not captured as modal formulas, but as a program written in AgentSpeak(L). An agent program consists of:

- B: set of grounded logical predicates known as the belief base. This set reflects the knowledge of an agent about itself, its environment and other agents.
- I: set of goals that determines what an agent wants to achieve depending on the external and internal inputs. An intention can be considered as an adoption of the program to meet the specification of these inputs.
- E: set of events that respond to changes of the working environment. Each event typically leads to a modification in the sets of beliefs and goals.
- A: set of actions that represents a change in the state of the working environment when taken by an agent.
- P: set of plans that aims to hierarchically decompose goals and to execute actions so that an agent can attain its desires. A plan has a rule-like format where the body contains goals or actions while the head has event and belief literals known as a context. Informally, a plan specifies what an agent should meet to fulfil its desires in the case of an event.

The behaviour of an agent is dictated by an AgentSpeak(L) program. Whenever there is a change in the working environment, an external event is generated. Also, an internal event can be produced by an agent from its own mental states. Both internal and external events are put into E. The agent can select an event from E and determines plans in P relevant to this event. The applicable plans are determined among the relevant plans if the contexts of these plans are logical consequences of the belief base B. For each event, there may be more than one applicable plan or option. The agent selects an option in response to that event. Therefore, the applicable plan becomes the intended mean. The agent pushes this plan on the top of the

existing intentions (for internal events) or creates a new intention (for external events). Finally, the agent selects an intended plan (a 'true' intention) for the execution. The selection functions for events, plans, and intentions are abstract and not specified by the language. The designer of the system has to write the code for these functions.

AgentSpeak(L) is provided with an interpreter and multi-agent platform known as Jason (Bordini and Hübner, 2006). The operational semantics for the interpreter has been extended by Bordini et al. (2005). The existence of AgentSpeak(L) facilitates the practical use of the abstract BDI architecture.

3.2.3 3APL

3APL (An Abstract Agent Programming Language) language is a hybrid approach of imperative and logic programming for constructing cognitive agents by combining Prolog and Java. Therefore, the 3APL platform exploits regular programming constructs and logical proof for querying states of agents. To some extents, the 3APL can balance the practical view of the software engineers and the formal view of the computer scientists in building multi-agent programs. The authors of the 3APL also investigate the features of Dribble (van Riemsdijk et al., 2003) and GOAL (Hindriks et al., 2001).

Initially, the 3APL by (Hindriks et al., 1998, 1999) includes programming constructs for implementing the notions of beliefs, plans and rules for plan revision. The language has then been extended with declarative goals, which can be updated via the set of reasoning rules (Dastani et al., 2003a, 2005c), and with communication (Dastani et al., 2003b). An agent can perform different types of actions including *mental actions* that modify the mental states, *communication actions* for exchanging information, *external actions* which change the working environment, *test actions* that verify the validity of an atomic formula against the belief base.

An agent in the 3APL consists of beliefs, plans, goals and reasoning rules. The reasoning process of the 3APL agents is captured by an interpreter, which determines deliberation operations to modify the programming constructs, such as applying a rule or adopting a goal or revising a plan. Typically, the deliberation cycle starts with finding an applicable planning rule to construct a plan. A plan is subject to apply plan-revision rules if the plan is not executable (it

does not contain any action). The first executable plan found is run by the interpreter.

3.2.4 Answer set programming

Answer-set programming (ASP), a logic programming language with answer-set semantics formalised as AnsProlog (Baral, 2003), intends to facilitate the knowledge representation and declarative problem-solving. There are several efficient implementations for the ASP, (Eiter et al., 1998; Niemelä and Simons, 1997). The ASP provides powerful and useful features for modelling multi-agent systems, such as the ability to cope with incomplete information, and handling multiple world views. Non-monotonic reasoning is stimulated by using the notion of Negation As Failure during reasoning. The statement of 'a bird normally can fly' can be expressed as $fly(X) \leftarrow bird(X)$, ***not*** $ab(X)$. The ***not*** operator 'tells' the program to check the validity of the literal in the current answer set. In the example, if there is no evidence against $ab(X)$, $fly(X)$ is added into the answer set.

Since an ASP program can have more than one stable model, epistemic operators (belief and knowledge) can be captured (Gelfond, 1994). The meaning of these operators can be interpreted as in possible worlds. An agent knows a piece of information if and only if this information is hold in every model of its program.

The ASP also supports reasoning about actions (Gelfond and Lifschitz, 1992) and recently intended actions (Baral and Gelfond, 2005). Depending on the situations, intended or planned actions are not actually performed by an agent. However, these actions are still persistent in the agent's *mind*, and can happen at an opportune moment in future time. These properties represent the notion of intention (Wooldridge, 2002, pp 65–70). Regarding this development, the ASP provides an alternative for capturing the operational semantics of the BDI agents.

3.3 Discussions

Modal logic has been heavily used as a conceptual tool for establishing the foundations of the analysis of epistemic and doxastic (that is, knowledge and belief) notions in terms of modal operators, thus paving the way to the field of agents and multi-agent systems. In this field, modal

operators proved to be very powerful conceptual tools for describing the internal (mental) states of the agents as well as the interactions among the agents. Also, modal logic is appropriate for providing a conceptual model for describing normative notions, such as obligations, permissions, rights, which influence the internal states of an agent in reasoning about an action. A more detail review on the different combinations of modal logics for building a theory for an agent can be found in van der Hoek and Wooldridge (2003).

In addition to modelling knowledge states of an agent, modal logics provide a tool for investigating the dynamic aspect of the agents' knowledge states on new events, such as a coming message. This new information reduces the ignorance of an agent about its working environment, including other agents. Eventually, that modifies the knowledge structures of an agent as well as its perception of the other agents. Non-monotonic logics cope with new information by adding meta-information on top of traditional propositional logic. Therefore, the original theory of an agent does not change over time.

ASP is a logic programming language that uses negation as failure to cope with incomplete information. Intentional notions can be simulated by special predicates in the language. This approach is not uncommon in the logic programming. However, the usage of these predicates limits the expression of the modal operators and may confuse the designers in specifying the behaviour of the agents.

An important requirement of logics for multi-agent systems is the computational tractability. Those modal logics that satisfy the equivalence (reflexive, symmetric and transitive) relation can be computationally-grounded. Hence, there are some compromises in the expressive capability of the programming languages to meet the tractability of the agents' program. More comprehensive lists of programming languages and platforms for multi-agent systems are presented in Bordini et al. (2006); Fisher et al. (2007).

A meaningful logic with a practical inferential engine should feature expressive and computational tractable in the modelling states of the agents. Furthermore, the designer of the logic should consider that an agent operating in a dynamic environment (an agent may be influenced by actions of other agents) and has a partial image of the environment.

4

Defeasible Logic

Defeasible logic is a logical formalism designed to cope with the problem of incomplete and conflicting information. Among the approaches to non-monotonic reasoning without negation as failure, such as courteous logic programs (Grosof, 1997) and LPwNF (Dimopoulos and Kakas, 1995), defeasible logic provides a simple but often more efficient and expressive, especially with regard to sceptical reasoning as shown by (Antoniou et al., 2000b).

In this chapter, we introduce defeasible logic following the formalisation of Billington (1993) owing to its simplicity and efficiency and the extension of the logic with ambiguity propagation. Next, we present different implementations of defeasible logic reasoning engines. Finally, we conclude the chapter by discussing the relationship between defeasible logic and logic programming when dealing with incomplete information.

4.1 Introduction

As an approach to non-monotonic reasoning (Antoniou, 1997; Marek and Truszczynski, 1993), defeasible logic is a very promising tool, able to efficiently cope with the issue of partial and conflicting knowledge. The origin of defeasible logic goes back to Nute (1987, 1994), when the logic had been designed with a particular concern about computational efficiency (Maher, 2001; Maher et al., 2001) and has been very well developed over the years (Antoniou et al., 2000a, 2001; Billington, 1993). The main idea of defeasible logic is to produce a plausible conclusion given a reasonable amount of information. The conclusion is considered to hold if and only if there is no counter-evidence or the counter-evidence is not strong enough to defeat the conclusion. Potential conflicts between pieces of information are handled by a superiority relation. This provides a compact representation and an effective way to accommodate new information.

In addition to the ease of deployment, defeasible logic has been used in various application domains, including the modelling of regulations and business rules (Antoniou et al., 1999a; Grosof et al., 1999), modelling of contracts (Governatori, 2005; Governatori and Pham, 2005a; Reeves et al., 1999), the integration of information from various sources (Antoniou et al., 1999b; Lee et al., 2006), and semantic web (Antoniou and Bikakis, 2007; Bassiliades et al., 2006; Governatori and Pham, 2005b; Kontopoulos et al., 2008). In the agent research domain, there is a line of works which proposes to use defeasible logic to model rational agents on: the deliberation process (Dastani et al., 2007; Falappa et al., 2004; Governatori et al., 2006b; Rotstein et al., 2007); normative and social aspects (Governatori and Rotolo, 2004; Governatori et al., 2006b); actions and planning (Dastani et al., 2005a,b; Ferretti et al., 2007; García et al., 2007; Simari et al., 2004); and communication (Boella et al., 2007a).

4.2 Defeasible logic

In the section, we present the essential concepts of defeasible logic, including the language, proof conditions and the principle of strong negation. The logical formalism is based on Billington (1993), which provides a simple and effective representation for dealing with conflicting and

incomplete information.

4.2.1 Basic concepts

The basic components of defeasible logic are: *facts*, *strict rules*, *defeasible rules*, *defeaters*, and a *superiority relation*.

Facts are undeniable statements, which are always true.

Strict rules, are similar to rules in classical logics. Given enough evidence, the conclusions produced by strict rules are unquestionable. In other words, the conclusions are considered as new facts.

Defeasible rules are different from strict rules in the way that their conclusions can be overridden by contrary evidences. Defeasible rules capture those statements that are usually true.

Defeaters are rules that cannot be used to draw any conclusion but to prevent some conclusions from some defeasible rules by producing evidence to the contrary.

The *superiority relation* defines priorities among the defeasible rules. That is, one defeasible rule may override the conclusion of another rule when we have to solve a conflict between rules with opposite conclusions. Strict rules always have priority over defeasible ones. However, the priority is not defined among the strict rules.

4.2.2 Formal definitions

A defeasible theory D is a triple $(F,R,>)$ where F is a finite set of facts, R a finite set of rules, and $>$ a superiority relation on R.

The language of defeasible logic consists of a finite set of literals. Given a literal l, we use $\sim l$ to denote the propositional literal complementary to l, that is if $l = p$ then $\sim l = \neg p$, and if $l = \neg p$ then $\sim l = p$.

A rule r in R is composed of an antecedent or body $A(r)$ and a consequent or head $C(r)$. $A(r)$ consists of a finite set of literals while $C(r)$ contains a single literal. $A(r)$ can be omitted from the rule r if it is empty. The connective between two parts of a rule represents the type of the rule, in particular, $A(r) \rightarrow C(r)$ for a strict rule; $A(r) \Rightarrow C(r)$ for a defeasible rule; while $A(r) \rightsquigarrow C(r)$ is for a defeater rule.

The set of rules R can include all three types of rules, namely R_s (strict rules), R_d (defeasible rules), and R_{dft} (defeaters). We will use R_{sd} for the set of strict and defeasible rules, and $R[q]$ for the set of rules whose head is q.

A conclusion derived from the theory D is a tagged literal and is categorised according to how the conclusion can be proved:

- $+\Delta q$: q is definitely provable in D
- $-\Delta q$: q is definitely unprovable in D
- $+\partial q$: q is defeasibly provable in D
- $-\partial q$: q is defeasibly unprovable in D.

Example 2. In considering the following statements, 'Penguin is certainly a type of bird; normally a bird can fly; actually, penguin cannot fly'. These statements can be formulated using defeasible logic.

Because we are certain about the biological classification of the penguin, we have a strict rule $r_1 : penguin \rightarrow bird$. For the last two statements we can have $r_2 : bird \Rightarrow fly$ and $r_3 : penguin \Rightarrow \sim fly$. Rule r_2 denotes that typically a bird can fly if there is no evidence against this conclusion. Rule r_3 gives a conclusion that a penguin does not have flying capability.

The set $R = \{r_1, r_2, r_3\}$ works well if we only have the fact of *bird*. From R, we can derive the conclusion of fly. Now, the fact of *penguin* is introduced to R, it causes a conflict between fly and $\sim fly$ since both r_2 and r_3 are triggered. This conflict can be solved by the superiority $>= \{r_3 > r_2\}$. That is r_3 overrides the conclusion from that of r_2.

4.2.3 Proof conditions

Provability is based on the concept of a derivation (or proof) in a defeasible theory $D = (F, R, >)$. Informally, definite conclusions can derive from strict rules by forward chaining, while defeasible conclusions can obtain from defeasible rules if and only if all possible 'attacks' are rebutted because of the superiority relation or defeater rules. The set of conclusions of a defeasible theory is finite. This set is the Herbrand base that can be built from the literals occurring in the rules and the facts of the theory.

A derivation is a finite sequence $P = (P(1), \ldots, P(n))$ of tagged literals satisfying proof conditions (which correspond to inference rules for each of the four kinds of conclusions). $P[1..i]$ denotes the initial part of the sequence P of length i. In the follow, we present the proof for definitely and defeasibly provable conclusions by Antoniou et al. (2001).

The definition of Δ describes just forward chaining of strict rules. For a literal q to be definitely provable there is a strict rule with head q, of which all antecedents have been definitely proved previously.

Definition 2. *The condition for a conclusion with tag $+\Delta$ is defined as:*

$+\Delta$*: If* $P(i+1) = +\Delta q$ *then*

(1) $q \in F$ *or*

(2) $\exists r \in R_s[q]\ \forall a \in A(r) : +\Delta a \in P[1..i]$

To show that q cannot be proven definitely, q must not be a fact. In addition, we need to establish that every strict rule with head q is known to be inapplicable. Thus, for every such rule r there must be at least one antecedent a for which we have established that a is not definitely provable $-\Delta q$.

Definition 3. *The proof for $-\Delta$ conclusion is defined as:*

$-\Delta$*: If* $P(i+1) = -\Delta q$ *then*

(1) $q \notin F$ *and*

(2) $\forall r \in R_s[q]\ \exists a \in A(r) : -\Delta a \in P[1..i]$

To show that q is provable defeasibly is more complicated, because the opposing chains of reasoning against q must be considered, (1) q is already definitely provable or, (2) the defeasible part of D is investigated. In particular, it is required that a strict or defeasible rule with head q that can be applied is in the theory (2.1). In addition, the possible 'attacks' must be taken into account. To be more specific: q is defeasibly provable providing that $\sim q$ is not definitely provable (2.2). Also (2.3), the set of all rules supporting $\sim q$ are considered. Essentially, each such a rule s attacks the conclusion q. The conclusion q is provable if each such rule s is not applicable or s must be counter-attacked by a rule t with head q and t must be stronger than s.

Definition 4. *The proof for a $+\partial$ conclusion is as follows:*

$+\partial$*:If* $P(i+1) = +\partial q$ *then either*

(1) $+\Delta q \in P[1..i]$ *or*

(2.1) $\exists r \in R[q]\ \forall a \in A(r) : +\partial a \in P[1..i]$ *and*

(2.2) $-\Delta{\sim}q \in P[1..i]$ *and*

(2.3) $\forall s \in R[{\sim}q]$ *either*

(2.3.1) $\exists a \in A(s) : -\partial a \in P[1..i]$ *or*

(2.3.2) $\exists t \in R[q]$ *such that* $t > s$ *and*

$$\forall a \in A(t) : +\partial a \in P[1..i]$$

The similar explanation is applied for proving $-\partial q$. In short, the theory D does not have any strict rule supporting q and one of following conditions: all defeasible rules for q are not applicable; there is a strict support for ${\sim}q$; at least one defeasible rule for ${\sim}q$ is applicable and successfully overrides the 'attack' from those rules for q.

Definition 5. *The condition for a $-\partial$ conclusion is constructed as:*

$-\partial$*:If* $P(i+1) = -\partial q$ *then either*

(1)$-\Delta q \in P[1..i]$ *and*

(2.1) $\forall r \in R[q]\ \exists a \in A(r) : -\partial a \in P[1..i]$ *or*

(2.2) $+\Delta{\sim}q \in P[1..i]$ *or*

(2.3) $\exists s \in R[{\sim}q]$ *such that*

(2.3.1) $\forall a \in A(s) : +\partial a \in P[1..i]$ *and*

(2.3.2) $\forall t \in R[q]$ *either* $t \not> s$ *or*

$$\exists a \in A(t) : -\partial a \in P[1..i]$$

Example 3. This example illustrates the reasoning process for a defeasible theory. Considering

the defeasible theory D in Example 2. We have a defeasible theory as follows.

$$D = \{F, \{R_s, R_d\}, >\}$$

where

$$F = \{penguin\}$$

$$R_s = \{r_1 : penguin \rightarrow bird\}$$

$$R_d = \{\ r_2 : bird \Rightarrow fly;$$
$$r_3 : pengin \Rightarrow \sim fly\}$$

$$>= \{r_3 > r_2\}$$

Theory D derives sequence P of tagged conclusions as

$$P = +\Delta penguin, +\Delta bird, +\partial penguin, +\partial bird, +\partial \sim fly$$

The conclusion of $+\Delta penguin$ is derived, because *penguin* is in the set of facts F. Also, this conclusion results in rule r_1 being triggered. Therefore, $+\Delta bird$ is added into the derivation sequence P. Owing to the definition of ∂, we have $+\partial penguin$ and $+\partial bird$ from $+\Delta penguin, +\Delta bird$.

The occurrence of two defeasible conclusions turns r_2 and r_3 to be applicable. Because of the superiority relationship the conclusion of fly is withdrawn. Finally, $+\partial \sim fly$ is included in P.

4.2.4 Strong negation principle

The principle of strong negation defines the relationship between positive and negative conclusions. Hence, enforcing the principle preserves the coherence and consistency of the conclusions. The meaning of tags $-\Delta$ and $-\partial$ is that it is not possible to obtain a proof for the corresponding literals. As shown in the proof conditions of these tags, a negative conclusion is made if and only if all possible proofs for the positive conclusion are investigated. Therefore, conclusions with tags $-\Delta$ and $-\partial$ are the outcome of a constructive proof that the corresponding positive conclusion is not provable.

The structure of the proof conditions for a pair of conflict tags $+\partial, -\partial$ or $+\Delta$ $(-\Delta)$ is the same, but the conditions are negated in some sense. We claim that the proof condition for a tag is the strong negation of the proof condition for its complement.

To show the principle, we first present the strong negation of a formula. Essentially, the strong negation can be simulated by a function that simplifies a formula by moving all negations to an innermost position in the resulting formula. This function behaves as follows:

$$
\begin{aligned}
&sneg(+\partial p \in X) && = -\partial p \in X \\
&sneg(-\partial p \in X) && = +\partial p \in X \\
&sneg(A \wedge B) && = sneg(A) \vee sneg(B) \\
&sneg(A \vee B) && = sneg(A) \wedge sneg(B) \\
&sneg(\exists x A) && = \forall x sneg(A) \\
&sneg(\forall x A) && = \exists x sneg(A) \\
&sneg(A) && = \neg sneg(A) \\
&sneg(A) && = \neg A \text{ if A is a pure formula}
\end{aligned}
$$

Definition 6. *The principle of the strong negation is that for each pair of tags such as $+\partial$, $-\partial$, the inference rule for $+\partial$ should be the strong negation of the rule of $-\partial$ and vice versa.*

4.3 Ambiguity propagation extension

Defeasible logic can be extended by an ambiguity propagating variant (see (Antoniou et al., 2000a; Governatori et al., 2004)). The superiority relation is not considered in the inference process of this variant. The extension introduces a new tag Σ, which shows a support for a literal in a defeasible theory. The tag $+\Sigma p$ means that there is a monotonic chain of reasoning that would lead to conclude p in the absence of conflicts. Thus, a defeasibly provable literal tagged with $+\partial$ is also supported. In contrast, a literal may be supported even though it is not defeasibly provable. Therefore, support is a weaker notion than defeasible provability. In the following, we present the extension conditions for $\pm\Sigma$ conclusions with respect to the superiority relationship among defeasible rules.

There is a positive support for a literal if a strict or defeasible rule supports this literal and all rules against this literal are either weaker or inapplicable.

Definition 7. *The positive support for a literal is defined as:*

$+\Sigma$: *If* $P(i+1) = +\Sigma q$ *then*

$\Delta q \in P[1..i]$ *or*

$\exists r \in R_{sd}[q]$:$\forall a \in A(r) : +\Sigma a \in P[1..i]$ *either*

$\forall s \in R_{sd}[\sim q]$:$\exists a \in A(s) : -\partial a \in P[1..i]$ *or*

$\exists t \in R_{sd}[q]$ *and* $t > s$ *and* $\forall a \in A(t) : +\Sigma a \in P[1..i]$

There is a negative support for a literal if all strict and defeasible rules for this literal are not supported or defeated by an applicable rule.

Definition 8. *The negative support for a literal is constructed as:*

$-\Sigma$:*If* $P(i+1) = -\Sigma q$ *then*

$-\Delta q \in P[1..i]$ *and*

$\forall r \in R_{sd}[q]$:$\exists a \in A(r) : -\Sigma a \in P[1..i]$ *either*

$\exists s \in R_{sd}[\sim q]$: $\forall a \in A(s) : +\partial a \in P[1..i]$ *or*

$\forall t \in R_{sd}[q]$ *and* $t \not> s$ *or* $\exists a \in A(t) : -\Sigma a \in P[1..i]$

We can achieve ambiguity propagation behaviour by making a minor change to the inference conditions for $+\partial_{AP}$ and $-\partial_{AP}$. That is attacks from other rules are not considered in the proof.

Definition 9. *The condition for a positive defeasible conclusion with respect to ambiguity is defined as:*

$+\partial_{AP}$:*If* $P(i+1) = +\partial_{AP} q$ *then either*

(1) $+\Delta q \in P[1..i]$ *or*

(2.1) $\exists r \in R_{sd}[q]$ $\forall a \in A(r) : +\partial_{AP} a \in P[1..i]$ *and*

(2.2) $-\Delta{\sim}q \in P[1..i]$ *and*

(2.3) $\forall s \in R_{sd}[\sim q]$ $\exists a \in A(s) : -\Sigma a \in P[1..i]$

By considering the principle of the strong negation, we derive the condition for $-\partial_{AP}$.

Definition 10. *The condition for an unprovable defeasible conclusion with respect to ambiguity is constructed as:*

$-\partial_{AP}$*:If* $P(i+1) = -\partial q$ *then*

(1) $-\Delta q \in P[1..i]$ *either*

(2.1) $\forall r \in R_{sd}[q]\ \exists a \in A(r) : -\partial_{AP} a \in P[1..i]$ *or*

(2.2) $+\Delta{\sim}q \in P[1..i]$ *or*

(2.3) $\exists s \in R_{sd}[{\sim}q]$ *such that* $\forall a \in A(s) : +\Sigma a \in P[1..i]$

In the following example, we illustrate the use of support notion and the inference with ambiguity.

Example 4. Considering a defeasible theory D as follows:

$$R_d = \{r_1 :\Rightarrow a; r_2 :\Rightarrow {\sim}a; r_3 :\Rightarrow b; r_4 : a \Rightarrow {\sim}b\}$$

Without the superiority relationship, there is no means of deciding between a and $\sim a$ and both r_1 and r_2 are applicable. In a setting where the ambiguity is blocked, b is not ambiguous, because r_3 for b is applicable. The rule r_4 is not, because its antecedent is not provable. If the ambiguity is propagated, we have evidence supporting all of four literals, because all of the rules is applicable. The tags $+\Sigma a, +\Sigma{\sim}a, +\Sigma b$ and $+\Sigma{\sim}b$ are included in the conclusion set. Moreover, we can derive $-\partial a$, $-\partial{\sim}a$, $-\partial b$ and $-\partial{\sim}b$ showing that the resulting logic exhibits an ambiguity propagating behaviour. In the second setting b is ambiguous, and its ambiguity depends on that of a.

4.4 Inferential engines

Among the non-monotonic logics, defeasible logic provides a method to deal with the problem of incomplete and conflicting information. The syntax of the logic is very simple and intuitive. However, there is a challenge in building an inferential mechanism for the logic so that this mechanism can be computationally efficient and flexible for dealing with practical problems. In this section, we present several implementations of the defeasible reasoning including d-Prolog, Deimos, Delores, and the DR-Family. Only Delores can compute all answers, while the other implementations are built as answering query systems. Furthermore, the design of Delores focuses mainly on tackling the computational cost.

4.4.1 d-Prolog

d-Prolog by (Covington et al., 1987) is a query-answering interpreter for defeasible logic constituting about 300 lines of Prolog. Originally, the system was designed for mostly small, non-recursive inheritance problems. The strict rules are represented directly as Prolog rules while defeasible rules are a new component in Prolog. The d-Prolog implementation differs from the formal description of the logic:

- d-Prolog does not implement loop-checking
- d-Prolog is not closed under strict rules. Also, defeasible logic is not closed. In addition to a literal and its complement, the conflicts between the literals can also defined by the special predicate *incompatible*. Therefore, the inference for strict rules has to be modified to keep the original semantics. In the case of two competing strict rules that are defeasibly satisfied, both of their heads are not added to the sequence of conclusions. The literals supported by these rules are not derivable.
- The superiority relation can be defined implicitly by specificity.

However, the d-Prolog implementation of defeasible logic has an issue (Maher et al., 2001) that is inherited from Prolog. Specially, the computation of conclusions depends on the order in which the rules are given. This effect becomes more obvious when experiencing theories containing cyclic dependencies among literals.

4.4.2 Deimos – A query answering defeasible logic system

Deimos (Maher et al., 2001) implements defeasible reasoning as a query-answering system by using the Haskell programming language. The design of Deimos meets logical and computational requirements, including correctness, traceability, efficiency, flexibility and maintainability. Therefore, this tool provides a promising environment for both of the ongoing research and solving practical problems in defeasible logic.

The central part of the system is the prover using a backward-chaining strategy to derive a literal, tagged with different proof strengths including definite, defeasible and support. The proof

of a conclusion is accomplished with a depth-first search with the memorisation of already-proved conclusions and the detection of duplicated conclusions. To improve the efficiency, the system deploys balanced binary trees and indexed structures, which represent literals and rules in a defeasible theory, in deploying memorisation and loop-checking.

Because of the functional programming of Haskell, expressions of inference conditions in Deimos precisely and directly correspond to what is specified by the logical formalism. That facilitates the implementations of different extension of defeasible logic in addition to the verification of these implementations. In addition, Deimos can provide the proof history that traces all the sub-goals in the evaluation for a query. That is helpful for examining the traces of reasoning process of the logic.

The user can interact with Deimos via a command line or web pages and investigate several pre-prepared defeasible theories. The present system now consists of about 4000 lines of Haskell code.

4.4.3 DELORES – DEfeasible LOgic REasoning System

Delores (Maher et al., 2001), implemented in about 4000 lines of C, is based on forward chaining, but this is only for the positive conclusions. The negative conclusions are derived by a dual process. The system relies on the transformation of a general defeasible theory to a basic theory, which only contains defeasible rules. The details of transformation are presented in (Antoniou et al., 2001).

Given a basic defeasible theory, the system constructs data structures based on indexed lists to facilitate the access from literals to rules and vice versa. A literal has indexed lists indicating its positive/negative occurrences in the head or body of a rule. The algorithm for proof conditions works as follows.

- Assert each fact (a literal) as a conclusion and removes this literal from the rules, where the literal positively occurs in the body, and 'deactivate' the rules where either its complements or its conflicting literals occur in the body.
- Scan the list of active rules for rules with the empty body. Take the literal from the head, remove the rule, and put the literal into the pending facts. The literal is removed from the

pending facts and added to the list of facts, if there is no such rule whose head contains the complements of the literal or it is impossible to prove these literals.

- It repeats the first step.
- The algorithm terminates when there is no more fact or rule with an empty body.

This algorithm outputs $+\partial$; $-\partial$ can be computed by an algorithm similar to this with the 'dual actions'. For $+\Delta$ we have just to consider similar constructions where we examine only the first parts of step 1 and 2. $-\Delta$ follows from $+\Delta$ by taking the dual actions. The computational complexity of the algorithm is $O(N)$, where N is the multiplication of the number of literals and rules in D (Maher et al., 2001).

Considering the transformations of the defeaters and superiority relation (see Antoniou et al. (2001) for details), the number of literals can be increased, at the most, 12 times. Furthermore, the time taken to produce the transformed theory is linear in the size of the input theory. Consequently, the implementation of full defeasible logic is still linear. A complete analysis of correctness and complexity is presented in (Maher, 2001).

4.4.4 DR-Family: defeasible reasoning for the web

Antoniou and Bikakis initiate a line of works that provides a defeasible reasoning mechanism for Semantic Web applications. The representation of defeasible theory is compliant with the RuleML (Rule Mark-up Language) format so that it facilitates the exchange of defeasible theories over the Web. A major effort of these systems puts the transformation of knowledge bases in Web formats, such as RDF/S (Resource Description Format/Schema), into defeasible knowledge. The reasoning core relies on the existing rule-programming systems, such as C Language Integrated Production System (CLIPS, 1992). DR-families can be extended to capture modal deontic defeasible logic (Governatori and Rotolo, 2004). In this section, we present the DR-Prolog (Antoniou and Bikakis, 2007) and the DR-Device (Bassiliades et al., 2004).

DR-Prolog

The design of DR-Prolog by Antoniou and Bikakis (2007) focuses on the following:

- the compliance with RuleML so that the defeasible theory can be easily portable between Web systems
- the basis of the translation of defeasible logic into the meta-program from (Antoniou et al., 2000a; Maher and Governatori, 1999)
- the flexibility of the reasoning mechanism that can perform either ambiguity blocking or ambiguity propagating
- integration with Semantic Web technologies. The system can work with rules and knowledge base in RDF/S (Resource Description Format/Schema) and OWL (Web Ontology Language) constructs.

DR-Prolog is a system for answering queries whose answers can have a level of strength, definite/defeasible provability. The user gives a query as a defeasible theory in RuleML format. The query is evaluated by a knowledge base in RDF/S & OWL typically built from the Internet. The system relies on the translation of different knowledge formats into logic programs. The core of the reasoning engine is built on XSB-Prolog due to the use of tabled predicates and the *sk_not* operator. The usage of XSB-Prolog provides the well-founded version of the meta-program; however, the complexity of the reasoning mechanism is quadratic (Witteveen, 1996). Furthermore, XSB-Prolog facilitates the integration with other systems such as RDF/S & OWL.

DR-Device

The DR-DEVICE system by Bassiliades et al. (2004) has the ability to reason about RDF data over multiple Web sources using defeasible logic rules. The main features of the DR-Device are,

- compliance with the representation of defeasible logic by allowing three types of rules, strict, defeasible, and defeaters
- support for both classical (strong) negation and negation-as-failure. Also, the system allows conflicting literals that are derived objects that exclude each other.
- accept multiple formats of knowledge representation such as RDF, RuleML.

The core of the system is implemented by the CLIPS production rule system and the R-DEVICE deductive rule system (Bassiliades and Vlahavas, 2004). In addition, the core accepts knowledge in different formats by converting the knowledge to a set of CLIPS-like or object-oriented ‘ deductive rules. The defeasible reasoning is deployed by the compilation into the generic rule language of R-DEVICE. This approach is extended to capture modal defeasible deontic logic (Governatori and Rotolo, 2004).

4.5 Discussions

Defeasible logic is a simple but efficient approach to cope with the problem of incomplete and conflicting information. The logic and its extensions attract the interest of research communities, especially in knowledge representation and reasoning, and multi-agent systems.

Defeasible Logic Programming DeLP, by García and Simari (2004), captures the concept of defeasible logic using logic programming as a declarative language and a defeasible argumentation (Simari and Loui, 1992) as the proof mechanism. The DeLP treats sets of strict and defeasible rules as two separated logic programs, but does not support defeater rules. Instead of using the explicit superiority relationship to solve conflicts, DeLP deploys dialectical analysis over proof trees of conflicting conclusions. The complexity of DeLP is attributed to the retrieval of a proof for a conclusion and the assessment of the acceptability of that proof, especially in the presence of contradictory conclusions. Cecchi et al. (2006) show that given a logic program composed of finite ground facts, strict and defeasible rules the complexity of the former task is $P-complete$ whilst that of the latter is $NP-complete$. This proof procedure is very expensive in comparison with that of Antoniou et al. (2001).

In this chapter, we follow (Billington, 1993) for the representation of defeasible logic, where the superiority relation among the rules is described by its strength (strict or defeasible) as well as its ability to override the attack from other rules. It is noted that the second condition only applies for the defeasible rule.

Compared to the approaches to non-monotonic reasoning by a logic program without negation as failure (Dimopoulos and Kakas, 1995), defeasible logic is more powerful (Antoniou et al., 2000b). A defeasible theory derives more desirable conclusions than a program without

negation as failure.

Antoniou et al. (2006) have shown a close relationship between the semantics of defeasible logic and those of extended logic programs. Given a translation into a program $P(D)$, a defeasible theory D exactly expresses the sceptical behaviour of $P(D)$ under the answer-set semantics with respect to the decisiveness condition. Without this condition, the intersection of all answer sets of program $P(D)$ is equal to the defeasible conclusions of theory D. However, under Kunen semantics (Kunen, 1987), the translation shows an equivalence between the defeasible theory D and program $P(D)$.

Governatori et al. (2004) show that the grounded semantics of argumentation systems in (Dung, 1993) can be characterised by defeasible logic with ambiguity propagation. Vreeswijk (1997) adapts Dung's argumentation systems and their semantics can be characterised by defeasible logic with ambiguity blocking. The grounded semantics and the notion of the acceptability of arguments (Dung, 1995) are characterised by the defeasible logic with ambiguity propagating and ambiguity blocking, respectively. Informally, an argument is accepted if it is not possible to provide counter-evidence to deny this argument. In the case of grounded (skeptical) semantics, it is not conclusive if the system contains arguments providing evidence against each other.

There are common concerns on the defeasible reasoning within approaches to non-monotonic reasoning such as floating conclusions (Horty, 2002; Makinson and Schlechta, 1991), argument reinstatement (Horty, 2001). Also, Brewka (2001) claims the superiority of the well-founded semantics of the extended logic program against defeasible logic. These issues are discussed in (Antoniou, 2006).

Justifying floating conclusions, which are conclusions supported by conflicting information, is an open debate. Horty (2002) claims that floating conclusions can be reasonable but non-monotonic reasoning pattern fails to justify those conclusions. However, Prakken (2002) shows that floating conclusions are often desirable and these conclusions can be controlled by the default rules of the reasoning system. The variants of defeasible logic in Antoniou et al. (2000a) do not support these conclusions. In fact, these conclusions can be controlled if there is explicit preference over conflicting information. The opinion of Horty (2001) on argument reinstatement requires further discussion. Defeasible logic can tackle the reinstatement easily because to trigger a rule r, every attack from other rules must be directly refuted by the stronger

rule s that supports the same conclusion as r.

In response to Brewka (2001), Antoniou (2006) admits that a meta-program, used to translate a defeasible theory into a logic program, does not fully handle the cyclic theory as well-founded semantics for extended logic programs, but it is not the case with the defeasible logic framework. Moreover, the direct use of well-founded semantics to translate strict rules can provide a counter-intuitive result. This translation can result in the implicit reference of the conclusions supported by the strict rules over those of defeasible rules. A strict rule when applying for defeasible facts should be treated as a defeasible one. Also, the representation of defeasible logic (Antoniou et al., 2000a) only defines a superiority relation over the rules that individually support a conclusion and its complements.

Within non-monotonic logics, defeasible logic is an interesting approach to the problem of incomplete and conflicting information because of its simple but effective representation. The logic is also equipped with efficient implementations containing flexible models of ambiguity. The complexity class of the reasoning mechanism is linear to the size of the defeasible theory (number of rules and literals), which allows having an executable and portable logic mechanism in many research problems, such as agent modelling (Governatori and Rotolo, 2008), web ontology reasoning (Antoniou and Bikakis, 2007) or process compliance modelling (Governatori and Milosevic, 2006; Governatori et al., 2006a).

5

Multi-agent framework based on defeasible logic

Agents within a group can have different perceptions of their working environment and autonomously fulfil their goals. However, they can be aware of beliefs and goals of the group as well as other members and they can adjust their behaviours accordingly. To model these agents, we propose a framework based on defeasible logic, DL-MAS, where we explicitly include knowledge commonly shared by the group and knowledge obtained from other agents. Agents demonstrate their social commitment to the group by avoiding actions that violate the 'mental attitudes' shared by the majority of the group.

Defeasible logic is chosen as our representation formalism for its computational efficiency, and for its ability to handle incomplete and conflicting information. In addition, it provides specifications for an agent that is both conceptual and executable. Hence, our agents can enjoy

the low computational cost while performing 'reasoning about others'. Finally, we present our DL-MAS framework and its extension with modal notions including Belief, Intention, and Obligation.

5.1 Introduction

In multi-agent systems, interactions between agents are often related to cooperation or competition in such a fashion that they can fulfil their tasks. Successful interactions often require agents to share common and unified knowledge about their working environment. However, autonomous agents observe and judge their surroundings according to their own view. Consequently, agents possibly have partial and sometimes conflicting descriptions of the world. In scenarios where they have to coordinate, they are required to identify the shared knowledge in the group and to be able to reason with the available information. Modelling those agents requires representing and reasoning with incomplete and conflicting information, which is beyond the classical logics and monotonic reasoning.

Recently, defeasible logic (Nute, 1987, 1994) has attracted considerable interest from the research community (Antoniou et al., 2000a, 2001; Billington, 1993; Chesñevar et al., 2003; Dix et al., 1999; García and Simari, 2004). In the agent research domain, there is a line of works that proposes to use defeasible logic to model the deliberation process of the rational agents (Dastani et al., 2007; Falappa et al., 2004; Governatori et al., 2006b; Rotstein et al., 2007); to capture normative and social aspects (Governatori and Rotolo, 2004; Governatori et al., 2006b); to reason about actions and planning (Dastani et al., 2005a,b; Ferretti et al., 2007; García et al., 2007; Simari et al., 2004); and to model communication (Boella et al., 2007a).

Defeasible logic is an elegant and computationally efficient tool (Maher, 2001; Maher et al., 2001) to deal with partial and conflicting knowledge. The key advantage of defeasible logic is being able to draw a plausible conclusion from a reasonable amount of information. In addition, defeasible logic provides a compact representation and an effective way to accommodate new information.

In this chapter, we propose a formal framework, DL-MAS, based on defeasible logic to describe the knowledge commonly shared by agents, and that obtained from other agents. The

new model enables an agent to reason about the environment and intentions of other agents in the group. Actions of an individual agent are constrained to a general expectation of the group of agents by balancing between the desires of an individual and the beliefs of the majority. To achieve that, we extend the reasoning mechanism of defeasible logic with the notion of *superior knowledge*. The extended mechanism allows an agent to integrate its mental attitude with a more trustworthy source of information such as the knowledge shared by the majority of other agents.

In the extension of the DL-MAS, we add modal notions including Belief, Intention and Obligation to have a fine-grained model of 'mental attitudes' and social actions. In this model, our agents have the ability to discover the 'conventions' of the group by exploring the majority of the mental attitudes of the group.

In the rest of this chapter, we introduce our modelling technique and discuss how to represent the knowledge base of the agents including the meta-knowledge about the agents' importance in Section 5.2. In addition, we outline the strategies to allow the agents to discover the approximate 'common attitudes' among the group. Next, Section 5.3 describes details of the DL-MAS mechanism for reasoning with a priority source of knowledge. Section 5.4 presents the defeasible rule markup, which originates from the Rule Markup Language as a knowledge representation tool and the algorithm of the reasoning engine. In Section 5.5, we show the integration of modal notions into our DL-MAS framework. We provide an overview of research works related to our system in Section 5.6. Finally, Section 5.7 concludes the chapter.

5.2 DL-MAS multi-agent framework

In this section, we introduce DL-MAS, a framework for multi-agent systems, where we use the defeasible logic to represent the knowledge structure of an agent. We motivate the role of the knowledge shared by the agents' group in placing constraints and expectations on the behaviour of an individual agent. In addition, we present the concept of majority knowledge and the strategies of an agent in using this special knowledge in the presence of conflicts. Finally, we define the method enabling an agent to ponder the 'prevalent opinion' of the majority of agents with its internal attitude.

5.2.1 Knowledge representation

In general, each individual agent can take any action by balancing its desires, its knowledge about the environment and the perception of other agents' behaviours. As a member of a group, each agent is aware of the mental attitudes commonly held among the group and should avoid actions that can violate the general expectation of the group. The reputation of an agent can be decreased owing to its violation. On the other hand, the behaviour of an individual can be significantly influenced by either members with a high reputation or the majority of the group. That means an agent's perception of others either strengthens its current knowledge or introduces new information. An agent can adjust its behaviours accordingly by considering its knowledge and the 'conventional wisdom' of the group. To capture this concept, we propose a knowledge structure for an agent that which consists of three components of *background*, *other members*, and *internal knowledge*, that is, the agent's own knowledge.

Definition 11. *Given a group of agents,* $\mathscr{A} = \{A_1, \ldots, A_{n+1}\}$ *, and a weight function* w_A*, which maps an agent* A_i *in* $\mathscr{A}$ *to a real value representing importance of the agent to the group:*

$$w_A : \{A_1, \ldots, A_{n+1}\} \mapsto \mathbb{R}^+$$

An individual agent in the group, considering itself as A_{me}*, has the knowledge structure* $\mathscr{T}$*, represented by a set of defeasible theories.*

$$\mathscr{T} = \{T_{bg}, T_{me}, \mathscr{T}_{other}\}$$

In detail, the elements of the knowledge structure are described as follows:

- T_{bg} is the *background theory* representing the background knowledge. This knowledge represents information commonly shared by all agents, which motivates general (social) behaviours. In addition, this knowledge can represent desires or restrictions popularly recognised among agents [1].

- T_{me} is the *internal theory* representing the own knowledge of A_{me}, which describes its own view about the working environment. This knowledge enables A_{me} to achieve its goals autonomously and distinctively.

[1] The terms of 'knowledge' and 'theory' are interchangeable in our framework, because a knowledge source is modelled by a defeasible theory.

- $\mathscr{T}_{other} = \{T_i : 1 \leq i \leq n+1 \ \& \ i \neq me\}$ where T_i is a defeasible theory that A_{me} obtains from A_i in $\mathscr{A}$. A weight function w_T determines the importance of a theory in $\mathscr{T}_{other}$

$$w_T(T_i) = w_A(A_i) | \ A_i \in \mathscr{A}$$

  That is the importance of a theory is derived from that of the corresponding agent. The knowledge of *other agents* provides a rough understanding of possible behaviours of individuals. This information could be learnt from experience or via information exchange. For example, Boella et al. (2007a) introduce a method for agents to construct rule-based knowledge about others via the communication activities.

Our approach deals with how an agent 'computes' collective wisdom based on the individual opinions of single agents. In particular, the agent A_{me} replicates the reasoning of the other agents individually and combines the results. This process is based on A_{me}'s perception of the other agents, where the perceived knowledge of the others is considered private.

Measuring the trustfulness of an agent during interaction is a major challenge in the multi-agent research. Within our framework, information from any agent is validated against the background knowledge. In effect, any information that violates the commonly known information is dropped. Moreover, it would be acceptable for an agent if new information is validated by a large number of agents and complies with the commonly-shared knowledge.

Our approach favours the internal view approach (Kinny et al., 1996) to the agents' behaviours in the sense that an individual agent can adopt or react to events depending on what the agent knows about the environment and the other agents. We believe that our proposed framework can be used as a tool for the external modelling. Interactions between agents in the group can be fully investigated and verified when every individual agent is equipped with complete knowledge of the other agents.

Example 5. In this example, we sketch out a basic scenario where the agents in a group make use of different types of knowledge. Suppose we have a group of animal rescue agents including

A_1, A_2, and A_3. These agents share background knowledge:

$$\begin{aligned} T_{bg} = \{R_d = \{ & r_1 :\Rightarrow catInDanger; \\ & r_2 : catInDanger \Rightarrow rescue; \\ & r_3 : risk \Rightarrow \sim rescue\}; \\ >= & \{r_3 > r_2\}; \end{aligned}$$

Essentially, if the background knowledge is the only source available the expected conclusions for the agents are $+\partial catInDanger$ and $+\partial rescue$.

Assume that A_3 identifies a risk as $\{R_d = \{r_1 :\Rightarrow risk\}\}$ and perceives some knowledge from A_2 but nothing from A_1. A_3's knowledge structure includes A_3's private knowledge and $\mathscr{T}_{other} = \{T_1, T_2\}$ where $T_1 = \emptyset$; $T_2 = \{R_d = \{r_1 :\Rightarrow risk\}\}$. The combination between A_3's knowledge and the background knowledge can result in $+\partial risk$ and $+\partial \sim rescue$.

By replicating the reasoning of A_1 and A_2 (based on A_3's perception of these agents), A_3 presumes A_1 supports $+\partial rescue$ while A_2 holds $+\partial \sim rescue$. Now, A_3 can either pursue its conclusions regardless of the knowledge from other agents or alter its conclusions by pondering the opinions perceived from the others. The next section presents our approach to this issue.

5.2.2 Majority knowledge

The *majority rule* from Lin (1996) retrieves a maximal amount of consistent knowledge from a set of agents' knowledge. Conflicts between agents can be approached by considering not only the number of agents supporting that information but also the importance [2] of the individual agents. This approach provides a useful method to discover information largely held by the agents. The majority knowledge can be used either to reinforce the current knowledge of an agent or to introduce new information into the agent's knowledge.

Owing to possible conflicting information within a source, the merging operator, by majority, cannot directly be applied to our framework. Instead, the majority rule pools potential joint conclusions derived by the defeasible reasoning, which resolves possible conflicts.

Considering the knowledge structure of an agent, A_{me}, C_i denotes the set of tagged conclusions that can be derived by the reasoning mechanism from the corresponding theory $T_i \in \mathscr{T}_{other}$,

[2] The importance can be interpreted as the reputation or the reliability of an agent.

obtained from the agent A_i. The level that the theory T_i supports a literal l is derived from the weight of the corresponding theory as follows:

$$support(l,T_i) = \begin{cases} w_T(T_i) & l \in C_i \\ 0 & \text{otherwise} \end{cases}$$

The *support* function shows that a literal has a support value only if this literal is provable by a theory. The strength of the literal, which enables the literal to override other conflicts within the theory, is not considered by the function. That is the reputation of the source is more important than the proof within the source.

The majority knowledge from the others, T_{maj}, whose elements are inferred from $\{C_1,\ldots,C_n\}$ by the majority rule, is determined by the formula:

$$T_{maj} = \left\{ c : \sum_{T_i \in \mathscr{T}_{other}} support(c,T_i) > \frac{W - w_T(T_{me})}{2} \right\}$$

where W, the total weight of the group, is defined as

$$W = \sum_{A_i \in \mathscr{A}} w_A(A_i)$$

Each conclusion in T_{maj} can have different support levels accumulated from individual theories. Hence, the weight of the majority conclusions is a set, $\{w_{maj}\}$, whose members have values ranging from $W - w_T(T_{me})$ to $\frac{W-w_T(T_{me})}{2}$. Example 6 illustrates the selection by the majority over sets of tagged conclusions.

Example 6. Suppose that three agents namely A_1, A_2, and A_3 are tackling the 'cat in danger' situation, which is presented in Example 5. The importance levels of these agents are identified as $w_A(\{A_1,A_2,A_3\}) \mapsto \{4,3,1\}$, therefore, the total weight is $W = 8$. Assume that A_3 has obtained sets of tagged conclusions C_1 and C_2 from theories T_1 and T_2 respectively. It is noticed that the importance levels of these sets are inherited from those of corresponding agents A_1 and

A_2. Details are as follows:

$$\begin{aligned} C_1 = \{ & +\Delta catOnTree, +\partial catInDanger, \\ & +\partial climbTree, +\partial risk, +\partial{\sim}rescue\} \\ w_T(T_1) = & w_A(A_1) = 4 \\ C_2 = \{ & +\Delta catOnTree, +\partial catInDanger, \\ & +\Delta climbLadder, +\Delta{\sim}risk, +\partial rescue\} \\ w_T(T_2) = & w_A(A_2) = 3 \end{aligned}$$

To retrieve majority conclusions from the others, A_3 combines C_1 and C_2 with respect to the importance of these sets. The superscript presents the weight of a conclusion accumulated from sources support that conclusion.

$$\begin{aligned} C_1 + C_2 = \{ & +\Delta catOnTree^7, +\partial catInDanger^7, \\ & +\partial climbTree^4, +\partial risk^4, +\partial{\sim}rescue^4 \\ & +\Delta climbLadder^3, +\Delta{\sim}risk^3, +\partial rescue^3\} \end{aligned}$$

A_3 now obtains the prevalent opinion T_{maj} and its weight $\{w_{maj}\}$ from A_1 and A_2 by applying the threshold 3.5 (owing to $\frac{w_A(A_1)+w_A(A_2)}{2}$) over the combination as:

$$\begin{aligned} T_{maj} = \{ & +\Delta catOnTree^7, +\partial catInDanger^7, \\ & +\partial climbTree^4, +\partial risk^4, +\partial{\sim}rescue^4 \\ \{w_{maj}\} = & \{7,4\} \end{aligned}$$

Proposition 1. *For any literal q, it is impossible to have both $+\partial q$ and $+\Delta{\sim}q$ in T_{maj}*

Proof. We can assume without any loss of generality that every defeasible theory has the same weight. Suppose that there is a pair $+\partial q$ and $+\Delta{\sim}q$ in T_{maj}. According to the proof condition of $+\partial$, $+\partial q$ is due to $+\Delta q$ or $-\Delta{\sim}q$. That means the number of theories, which definitely support $+\Delta q$ or do not have any strict proof for (that is, $-\Delta{\sim}q$), must be more than one-half of the theories. From this we infer that the number of theories, where $+\Delta{\sim}q$ holds, is less than one-half of the group. Consequently, $+\Delta{\sim}q$ is not in T_{maj}. Clearly, this contradicts the assumption.

A similar argument holds for the case of $+\partial{\sim}q$ and $+\Delta q$ in T_{maj}. □

Owing to the nature of defeasible logic proofs and the conflicts between knowledge sources, there can be strict and defeasible conclusions of a literal and its complement in the inference from individual sources. However, the outcome of the majority rule is still coherent.

Because T_{maj} is derived from what the agent A_{me} knows about the other agents, T_{maj} can conflict with T_{me} (the internal knowledge of A_{me}). In the case that A_{me} joins the majority pool, the greater importance (weight) A_{me} acquires, the greater influence it has on the joint knowledge. If the weight of A_{me} is greater than $W/3$, A_{me}'s support for any conclusion c is tantamount to half of the others' support for c. Hence, A_{me} has an opportunity to significantly influence the joint knowledge.

Two possible strategies can be applied by A_{me} to handle conflicts with the joint knowledge of the other agents:

1. *Adaptive strategy* if $w_{me} \not> \max(\{w_{maj}\})$. In this situation A_{me} should take into account the conclusions from the others, because it is unlikely that A_{me} can successfully override the conflicts from the joint knowledge. That also means T_{maj} can introduce new information to A_{me}.

2. *Collective strategy* if otherwise. A_{me} can defeat the conflicts from the other agents if A_{me} joins the pool. Hence the joint knowledge from the others reinforces the current knowledge of A_{me}. To obtain more knowledge, A_{me} should collect all possible consistent knowledge from the others, which is consistent with respect to the reliability of the owner.

In summary, the majority rule enables the individual agents to identify knowledge shared by the group. Depending on the weight, the agent can pursue one of two strategies, adapting to the majority or collecting all possible knowledge. In both strategies, the *background knowledge* commonly shared by the group is respected; that is, in case of a conflict between a conclusion from the background knowledge and either from the majority or the agent's knowledge, the conclusions, which are supported by the background part, prevail.

Example 7. Considering the knowledge structure of the agent A_3 in the Example 5, A_3's perception of other agents reveals that A_1 supports $+\partial rescue$ whilst A_2 holds $+\partial rescue$. The adaptive strategy guides A_3 to take $+\partial rescue$ no matter what A_3 thinks about the situation. In contrast,

the collective strategy requires A_3 to combine the knowledge from other into its reasoning process.

5.2.3 Defeasible reasoning with superior knowledge

In this section, we propose a simple method to integrate two independent defeasible theories, which have different levels of reliability. It is noticed that a defeasible theory has finite sets of facts and rules, and a derivation from the theory can be computed in linear time (Maher, 2001). Suppose that an agent considers two knowledge sources represented by defeasible theories labelled as T_{sp} – the superior theory, and T_{me} – the agent's internal theory. The agent considers that T_{sp} is more important than T_{me}. Thus, conclusions from the internal theory should be withdrawn if they conflict with the superior theory; the agent prefers the superior theory's conclusions to its own. In the following, we present the proof conditions extended from the standard defeasible reasoning (Billington, 1993) and the properties of the extended conditions.

Proof conditions

Owing to the transformations of the superiority relation and the defeater rules (Antoniou et al., 2001), we can assume that the two theories contain only strict and defeasible rules represented. To perform the defeasible reasoning, the agent generates a superiority relation over sets of rules as in $R_s^{sp} > R_s^{me}$ and $R_d^{sp} > R_d^{me}$. In this scheme, the subscript denotes the type of rules while the superscript indicates the type of the theory, which contains the rules. Meanwhile, the subscript represents the type of rules: s for the strict rules while d for the defeasible rules.

A derivation from the two theories is a finite sequence $P = (P(1), \ldots, P(n))$ of tagged literals satisfying proof conditions (which correspond to inference rules for each of the four kinds of conclusions). $P[1..i]$ denotes the initial part of the sequence P of length i.

The definite conclusion, $+\Delta q$, will be derived by performing forward chaining with the strict rules in the superior theory or in the internal theory if the complementary literals cannot be positively proved by the superior theory.

$+\Delta$: If $P(i+1) = +\Delta q$ then

(1) $q \in F$ or

(2) $\exists r \in R_s^{sp}[q]\ \forall a \in A(r) : +\Delta a \in P[1..i]$ or

(3) $\exists r' \in R_s^{me}[q]\ \forall a \in A(r') : +\Delta a \in P[1..i]$ and

$\forall r \in R_s^{sp}[\sim q]\ \exists a \in A(r) : -\Delta a \in P[1..i]$

The conclusions tagged with $-\Delta$ mean that the extended mechanism cannot retrieve a positive proof for the corresponding literals from the strict parts of both theories.

$-\Delta$: If $P(i+1) = -\Delta q$ then

(1) $q \notin F$ and

(2) $\forall r \in R_s^{sp}[q]\ \exists a \in A(r) : -\Delta a \in P[1..i]$ and

(3) $\forall r \in R_s^{me}[q]\ \exists a \in A(r) : -\Delta a \in P[1..i]$ or

$\exists t \in R_s^{sp}[\sim q]\ \forall a \in A(t) : +\Delta a \in P[1..i]$

The proof for $-\Delta$ satisfies the principle of strong negation (Antoniou et al., 2000a). Every statement of the proof for $-\Delta$ is the exact complement of that for $+\Delta$.

A defeasible conclusion $+\partial q$ can either be drawn directly from definite conclusions, or by investigating the defeasible part of the integrated theory. In particular, it is required that a strict or defeasible rule with an 'applicable' head q is in the theory (2.1). In addition, the possible 'attacks' must be either unprovable (2.2 and 2.3.1) or counter-attacked by 'stronger' rules (2.3.2).

$+\partial$: If $P(i+1) = +\partial q$ then either

(1) $+\Delta q \in P[1..i]$ or

(2.1) $\exists r \in R_{sd}^{sp}[q] \cup R_{sd}^{me}[q]\ \forall a \in A(r) : +\partial a \in P[1..i]$ and

(2.2) $-\Delta \sim q \in P[1..i]$ and

(2.3) $\forall s \in R_{sd}^{sp}[\sim q] \cup R_{sd}^{me}[\sim q]$ either

(2.3.1) $\exists a \in A(s) : -\partial a \in P[1..i]$ or

(2.3.2) $\exists t \in R_{sd}^{sp}[q] \cup R_{sd}^{me}[q]$ such that $t > s$ and

$\forall a \in A(t) : +\partial a \in P[1..i]$

The conclusions tagged with $-\partial$ mean that the extended mechanism cannot retrieve a positive proof for the corresponding literals from the strict and defeasible rules of both theories

or these conclusions are rebutted because of 'stronger' conclusions. The proof for $-\partial$ derives from that of $+\partial$ by using the strong negation principle.

$-\partial$: If $P(i+1) = -\partial q$ then
(1) $-\Delta q \in P[1..i]$ and
(2.1) $\forall r \in R^{sp}_{sd}[q] \cup R^{me}_{sd}[q]\ \exists a \in A(r) : -\partial a \in P[1..i]$ or
(2.2) $+\Delta{\sim}q \in P[1..i]$ or
(2.3) $\exists s \in R^{sp}_{sd}[{\sim}q] \cup R^{me}_{sd}[{\sim}q]$ such (2.3.1) $\forall a \in A(s) : +\partial a \in P[1..i]$ and
(2.3.2) $\forall t \in R^{sp}_{sd}[q] \cup R^{me}_{sd}[q]$ either $\exists a \in A(t) : -\partial a \in P[1..i]$

Example 8. In this example, we illustrate the use of the extended conditions in the context of our rescue agents. Suppose that the agent A_{me} has two knowledge sources T_{sp} and T_{me} as:

$$\begin{aligned} T_{sp} = \{R_d = \{\ & r_1 :\Rightarrow catInDanger; \\ & r_2 : catInDanger \Rightarrow rescue; \\ & r_3 : risk \Rightarrow {\sim}rescue\}; \\ >= & \{r_3 > r_2\}; \\ T_{me} = & \{R_d = \{r_4 :\Rightarrow risk\}\} \end{aligned}$$

At first, the integration process derives $+\partial risk$ because of the empty body rules r_4 and no 'attack' against the conclusion of *risk* from T_{sp} and T_{me}. The conflict between *rescue* and ${\sim}rescue$ is solved by the superiority of r_3 over r_2. That is A_{me} infers $+\partial{\sim}rescue$.

The conclusion of *catInDanger* will be changed by the extended reasoning if A_{me} holds a strong evidence against *catInDanger* from T_{sp}. This evidence can be represented as a definite conclusion $+\Delta{\sim}catInDanger$ derived from T_{me}. Now the conclusion $+\partial catInDanger$ is overridden by the stronger evidence $+\Delta{\sim}catInDanger$ from a 'weaker' theory T_{me}. The intuition is that conclusions supported by T_{sp} generally override those from T_{me} but a definite conclusion from T_{me} is only rejected by definite ones from T_{sp}.

Given two defeasible theories T and S and a proof tag #, we use $T \vdash \#q$ to mean that $\#q$ can be proved from theory T using the basic proof conditions of the defeasible logic (see Section 4.2 on Chapter 4), while $T \ni S \vdash \#q$ means that there is a derivation of $\#q$ from the theory integrating

T and S using the proof conditions given in this section and where T plays the role of T_{sp} and S the role of T_{me}.

Properties of extended reasoning

A defeasible theory has two essential properties: coherent and consistent, which are defined as follows.

Definition 12. *A defeasible theory is* coherent *if it is impossible to derive from it a pair* $-\Delta q$ *and* $+\Delta q$*, or* $-\partial q$ *and* $+\partial q$*.*

Definition 13. *A defeasible theory is* consistent *if it is possible to derive* $+\partial q$ *and* $+\partial{\sim}q$ *if and only if the theory derives both* $+\Delta q$ *and* $+\Delta{\sim}q$*.*

The extended defeasible reasoning with the superior knowledge has the properties:

1. The extended mechanism does not provide any proof against a strict conclusion derived from the superior theory (see Proposition 2).
2. The conclusions from the extended mechanism overrides defeasible conclusions obtained from the superior theory by a strict counter-evidence from its internal knowledge (see Proposition 3).
3. The extended reasoning mechanism is coherent and consistent (see Proposition 4).

Proposition 2. *If* $T_{sp} \vdash +\Delta q$ *then* $T_{sp} \ni T_{me} \nvdash +\Delta{\sim}q$ *and* $T_{sp} \ni T_{me} \nvdash +\partial{\sim}q$

Proof. This result directly draws from the proof conditions of our reasoning mechanism. Assume that q is held by the strict rules of the superior theory, $T_{sp} \vdash +\Delta q$, while $+\Delta{\sim}q$ is computed by the integrated theory, $T_{sp} \ni T_{me} \vdash +\Delta{\sim}q$. With regard to the proof for $+\Delta$, $+\Delta{\sim}q$ can be derived either from (1) or (2) that is $T_{sp} \vdash +\Delta{\sim}q$. Nevertheless, this violates the assumption. If $+\Delta{\sim}q$ is proved by (3), this proof condition requires the strict rules from the superior theory does not have any support q, $T_{sp} \nvdash +\Delta q$. Again, this contradicts the assumption. Thus, the first part of the proposition is proved.

$T_{sp} \vdash +\Delta q$ blocks the derivation of $+\Delta{\sim}q$ and $-\Delta{\sim}q$ from the integrated theory. Therefore, (1) and (2.2) are not satisfied in the proof of $+\partial$. Consequently, $+\partial{\sim}q$ is blocked. □

This proposition states if a strict conclusion is derived from the superior theory, the extended mechanism does not provide any proof for its negation.

Proposition 3. *If $T_{sp} \vdash \sim\Delta\sim q$ and $T_{me} \vdash +\Delta q$ then $T_{sp} \ni T_{me} \vdash +\partial q$*

Proof. Assume that the integrated theory derives the contradiction, $T_{sp} \ni T_{me} \vdash +\partial\sim q$, given the conditions of the proposition. According to the proof of $+\partial$, $+\partial\sim q$ requires $T_{sp} \ni T_{me}$ to hold either $+\Delta\sim q$ or $-\Delta q$, because the conditions of (2.1) and (2.3) can be met owing to the superiority of T_{sp} over T_{me}. As shown in the proof of Δ, the derivation of $+\Delta\sim q$ and $-\Delta q$ from $T_{sp} \ni T_{me}$ respectively requires $T_{sp} \vdash +\Delta\sim q$ or $T_{me} \vdash +\Delta\sim q$, $T_{sp} \nvdash +\Delta q$ and $T_{me} \nvdash +\Delta q$. Clearly, these requirements violate the assumptions of $T_{sp} \vdash \sim\Delta\sim q$ and $T_{me} \vdash +\Delta q$. □

This proposition claims the conclusions from the extended mechanism can violate defeasible conclusions obtained from the superior theory if the agent has a strong evidence of the contradiction in its internal knowledge.

Proposition 4. *The defeasible reasoning with the superior knowledge is coherent and consistent.*

Proof. First, we investigate the derivation of strict conclusions. By the definition of the standard defeasible logic, the strict rules and facts of individual theories do not concurrently hold $-\Delta q$ and $+\Delta q$. The only case leading to conflicting conclusions is where the superior theory supports q, $T_{sp} \vdash +\Delta q$, while the internal theory holds the contradiction, $T_{me} \vdash +\Delta\sim q$. According to the proof condition for $+\Delta$, the contradiction is always rejected. Therefore, the derivation of strict conclusions is coherent.

The coherence of the defeasible part of the reasoning with the superior theory directly inherits from that of the standard defeasible reasoning in the case that both theories have empty sets of facts and strict rules. The only case that can lead to the incoherence is $T_{sp} \vdash +\partial q$ and $T_{me} \vdash +\Delta\sim q$. That means the defeasible rules of the superior theory prove a conclusion violating that supported by the strict rules of the internal theory. However, as mentioned above, the defeasible part of the superior theory is defeated by the strict part of the internal theory. That is, the derivation of defeasible conclusions is also coherent.

The consistency property follows from the coherence property. It is impossible to have conflicting conclusions from the proof conditions for defeasible conclusions. The only source of inconsistency comes from the proof for strict conclusions. □

The extended mechanism goes beyond the standard defeasible reasoning, because it extends the superiority relation of rules to that of theories. This only increases the size of the theory to be investigated but not the complexity of the reasoning process. Hence the complexity class of the reasoning algorithm (Maher, 2001) remains unchanged. Remarkably, the extended mechanism keeps the two theories intact. This feature facilitates the further manipulation of knowledge about others, because the agents' knowledge is likely to evolve from interactions such as communication. Moreover, this provides the agents with flexibility to select different inference mechanisms to handle conflicting information.

5.3 DL-MAS reasoning mechanism

In Section 5.2, we present different types of knowledge, which can be modelled by an agent, and a method to combine two knowledge sources. Based on these notions, we are now able to describe how to incorporate the majority rule into a reasoning mechanism.

The reasoning mechanism operates in two steps as:

1. In the first step, an agent identifies the majority knowledge from the other agents.

2. In the second step, the agent performs either adaptive or collective reasoning depending on its weight to obtain the final conclusions.

We show that the complexity of DL-MAS reasoning mechanism is still linear as in Proposition 5.

5.3.1 Identify the majority knowledge

This step is completed by applying the standard defeasible reasoning over the individual knowledge sources and by following the majority rule over the set of tagged conclusions.

1. *Draw defeasible conclusions from the others.*

   We run the extended defeasible reasoning over theories obtained from the others. The background theory T_{bg} is considered superior for every theory. Formally, this step is described by:

   $$T_{bg} \ni T_i \vdash C_i : 1 \leq i \leq n$$

2. *Establish the majority knowledge T_{maj}.*

   The extended defeasible reasoning already guarantees that conflicts are removed from the final set of conclusions. Hence, the majority knowledge can be determined by applying the *majority rule* (Lin, 1996) over the sets of defeasible conclusions, $\{C_i : 1 \leq i \leq n\}$, from the previous step. The conclusions with support from the majority will be projected to the joint knowledge.

5.3.2 Reasoning strategies

At this stage, the set of knowledge sources is reduced to the background, the majority, and the agent's own knowledge. Depending on the weight, an individual agent can either follow the majority knowledge or collect all possible information. The two strategies are implemented by the defeasible reasoning with the superior knowledge.

Adaptive reasoning

It takes two steps to derive the final conclusions as shown in Figure 5.1. First, the agent combines the background and its own knowledge by considering these two as the superior knowledge and the internal knowledge respectively. Next, the joint knowledge from the other agents is used to adjust the derivation from the first step. That is, the agent withdraws conclusions, which violate the joint knowledge.

$$T_{maj} \ni (T_{bg} \ni T_{me}) \vdash C'_{me}$$

Example 9 demonstrates how an agent applies the adaptive reasoning strategy. Technically, this process requires transformations from defeasible conclusions to rules as in Antoniou et al. (2001). A definitely provable conclusion can be converted to a strict rule whose head contains

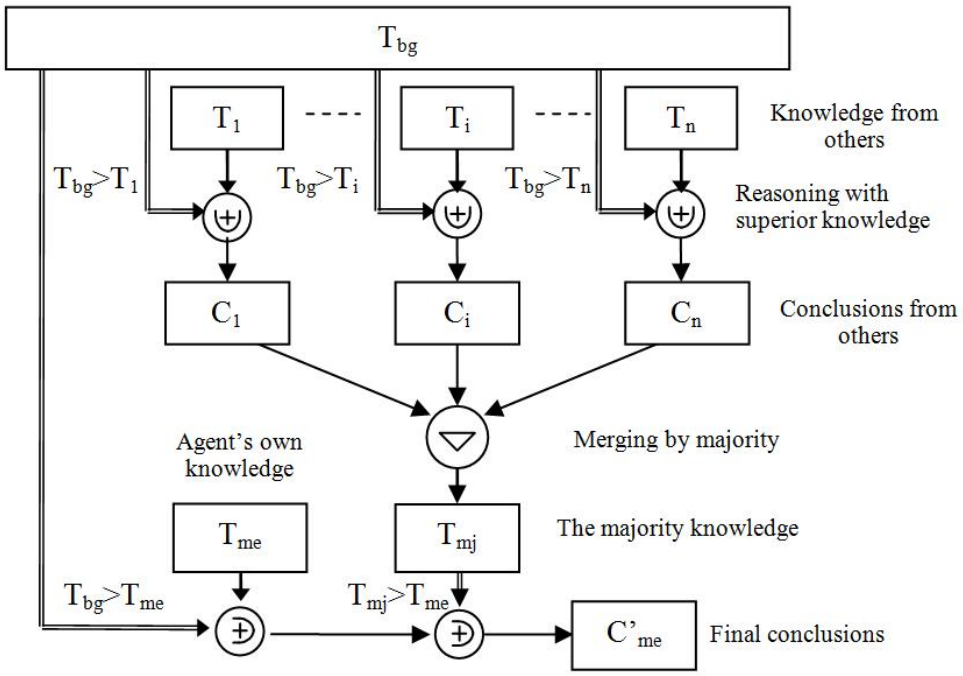

FIGURE 5.1: Adaptive reasoning

the corresponding literal, and whose body is empty. Similarly, a defeasibly provable conclusion can be transformed into a defeasible rule.

Example 9. In this example, we show how an agent uses the majority knowledge by the adaptive reasoning. Consider a rescue team of three agents $\{A_1, A_2, A_3\}$ having weights of $\{4, 3, 1\}$ respectively. The following knowledge is shared by the group:

$$
\begin{aligned}
T_{bg} = \{ & F = \{catOnTree\}; \\
& R_d = \{\ r_1 : catOnTree \Rightarrow catInDanger; \\
& \qquad r_2 : catInDanger \Rightarrow rescue; \\
& \qquad r_3 : risk \Rightarrow \sim rescue\}; \\
& >= \{r_3 > r_2\}\}
\end{aligned}
$$

This knowledge provides basic information that a cat is in danger and the team should do the rescue provided that it is safe. Suppose that A_3, considering itself as A_{me}, just arrives and does not have any knowledge about the field, that is A_3's internal knowledge is empty $T_{me} = \emptyset$.

However A_3 knows about A_1 and A_2 respectively:

$$\begin{aligned} T_1 =& \{R_d = \{\ r_1 :\ \Rightarrow climbTree; \\ & \qquad r_2 : climbTree \Rightarrow risk; \\ & \qquad r_3 : keenAtClimbing \Rightarrow \sim risk\}; \\ & >= \{r_3 > r_2\}\} \\ T_2 =& \{\ F = \{climbLadder\}; \\ & \quad R_s = \{r_1 : climbLadder \rightarrow \sim risk\}\} \end{aligned}$$

By combining the background knowledge with those from A_1 and A_2 respectively, A_3 can derive following knowledge:

$$\begin{aligned} C_1 = \{\ & +\Delta catOnTree^4, +\partial catInDanger^4, \\ & +\partial climbTree^4, +\partial risk^4, +\partial \sim rescue^4\} \\ C_2 = \{\ & +\Delta catOnTree^3, +\partial catInDanger^3, \\ & +\Delta climbLadder^3, +\Delta \sim risk^3, +\partial rescue^3\} \end{aligned}$$

The superscript of a literal presents the support level (weight) from the corresponding theory. Based on the support, the majority rule projects the knowledge as

$$\begin{aligned} T_{maj} = \{\ & +\Delta catOnTree^7, +\partial catInDanger^7, \\ & +\partial climbTree^4, +\partial \sim rescue^4\} \\ \{w_{maj}\} = \{7,4\} \end{aligned}$$

The superscript of each majority conclusion represents the support level accumulated from the weight of theories. Because A_3 has $T_{me} = \emptyset$, the combination with the background infers

$$C_{me} = \{+\Delta catOnTree, +\partial catInDanger, +\partial rescue\}$$

By considering the majority knowledge as the superior knowledge, A_3 adjusts its derivation to:

$$\begin{aligned} C'_{me} = \{ & +\Delta catOnTree, +\partial catInDanger, \\ & +\partial climbTree, +\partial \sim rescue\} \end{aligned}$$

Now A_3 adapts its behaviour toward that of the majority by dropping its *rescue* conclusion.

Collective reasoning

Owing to its importance, the agent considers itself as dominant over the other agents. Instead of reinforcing its current knowledge, the agent derives new knowledge by accumulating all possible consistent knowledge from the others. This is achieved by performing the sequence of inference processes with the superior theory as shown in Figure 5.2.

The sequence starts with the theory having the minimum weight and takes the next theory in the order of the weight as the superior theory. The sequence ends with the background theory, which implicitly has the maximum weight. This reasoning strategy requires a total order over the weights of agents in the group. This reasoning strategy is illustrated in Example 10.

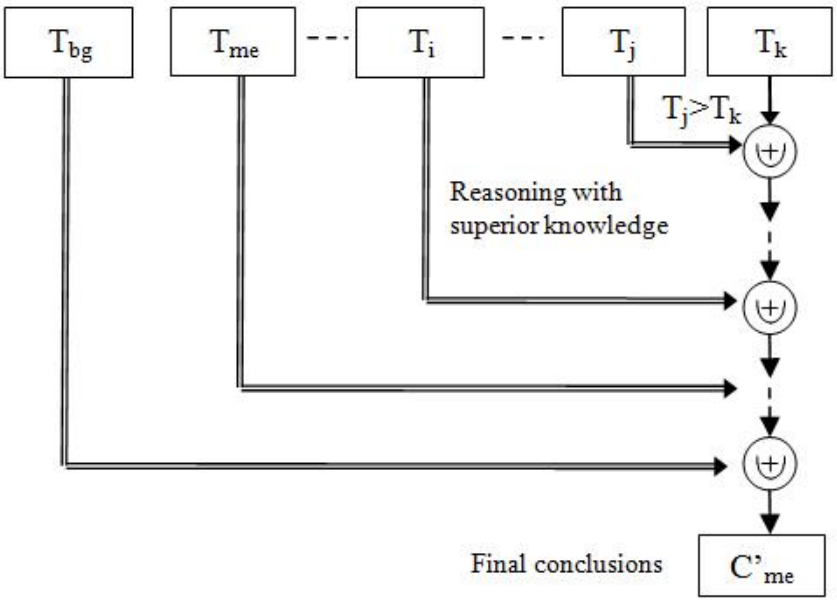

FIGURE 5.2: Collective reasoning

Example 10. This example explains the idea of the collective reasoning strategy. Consider the agent group in the Example 9 again, but with the importance level of A_3 increased to 8 so that the weights of $\{A_1, A_2, A_3\}$ are now $\{4, 3, 8\}$. The weight of the majority conclusions ranges from 4 to 7 as found in $\{w_{maj}\} = \{7, 4\}$.

$$\{+\Delta catOnTree^{7}, +\partial catInDanger^{7},$$
$$+\partial climbTree^{4}, +\partial{\sim}rescue^{4}\}.$$

Directly, the importance of A_3 is higher than $max(\{w_{maj}\})$. A_3 employs collective reasoning by generating the sequence of $T_2 + T_1 + T_{bg}$, because the internal theory of A_3, T_{me}, is empty. The integration of $T_2 + T_1$ derives the conclusions:

$$\{+\Delta climbLadder, +\Delta{\sim}risk, +\partial climbTree\}$$

The conclusion of $+\partial risk$ from the superior T_1 is overridden by the stronger $+\Delta{\sim}risk$ from the inferior T_2. In the next step, the combination with the background knowledge, A_3 infers

$$\{+\Delta climbLadder, +\Delta{\sim}risk, +\partial climbTree,$$
$$+\Delta catOnTree, +\partial catInDanger, +\partial rescue\}$$

A_3 now accepts the conclusion of *rescue* from A_2 even it conflicts with A_1's belief and A_1 is considered more reliability than A_2. In this case, the proposition 3 holds. A_1 does not have a counter-evidence strong enough to defeat that from A_2.

Proposition 5. *The complexity of the proposed mechanism is in the $O(n)$ class, where n is the size of the theory.*

Proof. The low computational cost of the proposed mechanism is due to the efficiency of the majority rule and the defeasible reasoning. As in Lin (1996), the process to determine the information set supported by the majority has linear complexity depending on the number of conclusions derived from all of agents' theories. This efficiency also owes it to the efficient resolution of conflicts of the defeasible reasoning, which produces conflict-free conclusions. Therefore, agents do not have to check the consistency of the conclusions from their knowledge components.

Compared to the standard defeasible reasoning mechanism, reasoning under a superior theory simply increases the size of the theory, which equals the number of rules in both theories (the superior and the internal). In the case that an agent follows the adaptive strategy, the total size of the theories, which are investigated by $(n+1) \times |T_{bg}| + |T_{me}| + \sum_{i=1}^{n} |T_i|$, where n is the number of theories from the other agents. The size of the majority knowledge is omitted, because this knowledge is derived from $\{T_1, ..., T_n\}$. The upper bound is $2 \times (n+1) \times max(\{|T_{bg}|, |T_{me}|, |T_1|, ..., |T_n|\})$.

If the collective strategy is selected, the size of the theories investigated by the agent equals $|T_{bg}| + |T_{me}| + \sum_{i=1}^{n} |T_i|$. Considering the cost of retrieving the majority knowledge, the upper bound for the collective reasoning is double that of the adaptive strategy.

However, this cost can be reduced if the agent applies a fixed threshold (weight), instead of the maximum weight of majority conclusions, to select the reasoning strategy.

Therefore, the computational cost of the mechanism is linearly proportional to the total number of literals and rules, which are presented in the knowledge base of the agent. □

The defeasible reasoning with the superior knowledge offers a method to accumulate knowledge from multiple sources. It relies on the weight of the sources and the ambiguous blocking (Antoniou et al., 2001) to deal with the potential conflict within/among the sources. Intuitively, the collective reasoning should allow the ambiguities to be propagated along the sequence of reasoning with the superior knowledge. Conflicts between the sources of knowledge can be approached by pondering the strength of the proof. This approach can increase the quality of the integration but boosts the computational cost. At the end, this possibility is well worth further investigation.

5.4 DL-MAS Implementation

In this section, we present our modification of RuleML to capture defeasible theories and our extension of DELORES for the defeasible reasoning with the superiority knowledge. We chose DELORES because of its ability to compute all conclusions from a defeasible theory.

5.4.1 DRM - Defeasible rule markup

The Rule Markup Language (RuleML) is an eXtended Markup Language (XML) dialect for representing rules. It offers facilities for specifying different types of rules from derivation rules to transformation rules to reaction rules. RuleML already supports derivation rules via the *Imp* element. However, we need to define a new syntax to represent the strength of the rules and superiority relations. In this section, we present our defeasible rule markup language. The syntax of the language is shown in Figure 5.3. Following the proposal of DR-DEVICE

(Bassiliades et al., 2004), every rule in the knowledge structure now has a *@ruletype* attribute taking one of three values: *strictrule*, *defeasiblerule* or *defeater*.

Example 11. The defeasible rule $r_2 : catInDanger \Rightarrow rescue$ is represented in RuleML format as:

```
<Imp label = "r2" ="defeasiblerule">
<head>
<Atom><Rel>rescue</Rel></Atom>
</head>
<body>
<Atom><Rel>catInDanger</Rel></Atom>
</body>
</Imp>
```

The conclusions from the corresponding theories, represented by the *Conclusion* element, are also stored for exchanging knowledge or explaining the agents' behaviour. Each conclusion includes the literal and the strength of the proof.

Example 12. The conclusion $+\partial rescue$of $r_2 : catInDanger \Rightarrow rescue$ is represented as:

```
<Conclusion ruletype="defeasiblerule">
<Tag>defeasible</Tag>
<Atom><Rel>rescue</Rel></Atom>
</Conclusion>
```

The DR-DEVICE expresses the superiority relation by using the `@superior` attribute on the superior rule as a link to the `@ruleID` label of the inferior rule. We found this unsuitable, because we may need to mark a rule as superior to more than one other rule and an XML element can only bear a single `@superior` attribute. Using the scheme from (Governatori, 2005) instead, we explicitly represent the superiority relation using the distinguished predicate `Sup`.

Example 13. For example, $r_1 \succ r_2$ is represented as

```
<Imp label = "r1" ruletype="defeasiblerule" > .... </Imp>
<Imp label = "r2" ruletype="defeasiblerule" > .... </Imp>
<Sup superior="r1" inferior="r2">
</Sup>
```

Finally, every defeasible theory in the knowledge structure, containing a collection of rules, facts, and superiority, is represented by the *SDLTheory* element having two attributes, namely *source*, and *weight*, corresponding to the source name and the weight of the theory.

Example 14. For example, $r_1 \succ r_2$ is represented as

```
<SDLTheory source = "A1" weight = "7">
<Imp label = "r1" ruletype="defeasiblerule" > .... </Imp>
<Imp label = "r2" ruletype="defeasiblerule" > .... </Imp>
<Sup superior="r1" inferior="r2">
</Sup>
</SDLTheory>
```

The complete definitions of elements of defeasible theory are presented in Figure 5.3 using Document-Type-Declaration syntax.

5.4.2 Algorithm for the extended mechanism

The process of the extended defeasible reasoning with superiority theory operates with a pair of theories in three phases:

1. In the pre-processing phase, the theory in the RuleML format is loaded into the mechanism and is transformed into an equivalent theory without superiority relation and defeaters. Also, this phase combines a pair of theories by using the same technique for the priority between them.

```
<!ELEMENT SDLTheory ((Rule|Fact|Sup|Conclusion)*)>
<!ATTLIST SDLTheory source CDATA #REQUIRED>
<!ATTLIST SDLTheory weight CDATA #IMPLIED>

<!ELEMENT Atom (Not?,Rel,(Ind|Var)*)>
<!ELEMENT Not EMPTY>
<!ELEMENT Rel (#PCDATA)>
<!ELEMENT Var (#PCDATA)>
<!ELEMENT Ind (#PCDATA)>
<!ATTLIST Ind href CDATA #IMPLIED>

<!ELEMENT Conclusion (Tag,Atom)>
<!ELEMENT Tag (#PCDATA)>

<!ELEMENT And (Atom)*>
<!ELEMENT body (And)>
<!ELEMENT head (Atom)>

<!ELEMENT Rule (head?,body?)>
<!ATTLIST Rule strength CDATA #REQUIRED>
<!ATTLIST Rule href CDATA #IMPLIED>
<!ATTLIST Rule label ID #REQUIRED>
<!ATTLIST Rule time CDATA #IMPLIED>

<!ELEMENT Sup EMPTY>
<!ATTLIST Sup superior IDREF #REQUIRED
       inferior IDREF #REQUIRED>

<!ELEMENT Fact (Atom)>
<!ATTLIST Fact href CDATA #IMPLIED>
<!ATTLIST Fact label ID #IMPLIED>
```

FIGURE 5.3: Data Type Definition of Defeasible Rule Markup

2. In the next phase, the rule loader, which parses the theory obtained in the first phase, generates the data structure for the inferential phase.

3. In the final phase, the inference engine applies modifications to the data structure, where, at every step, it reduces the size of the data structure.

Theory transformation. The transformation flattens the superiority structure of an individual theory by removing the defeaters rules and the superiority relation among the rules by applying the transformation rules in Antoniou et al. (2001).

Because the mechanism works with multiple defeasible theories, the transformation is recalled to remove the superiority relation between the theories. For every conflict between the internal theory T_{in} and its superior one T_{sp}, the transformation function assumes rules from T_{sp} have the priority over those from T_{in}. It is noticed that the process of conflict detecting excludes the temporary literals generated by the transformation rules.

Rule loader. The rule loader creates a data structure as shown in Figure 5.4. For every literal in the theory, the loader creates an entry whose structure includes:

- a list of (pointers to) rules having the literal in the head. To simplify the data structure, a literal from the head of a rule is built from the head atom of the corresponding rule.

- a list of (pointers to) rules having the literal in the body

- a list of (pointers to) entries of complements of the literal (incompatible ones).

To improve the computational performance, every list in the data structure is implemented as a hash table. The list of complements of a literal provides the flexibility for further development. That is, the negation of a literal can be flexibly defined outside of the reasoning mechanism.

Inferential engine. The engine is based on an extension of the Delores algorithm proposed in Maher et al. (2001) as a computational model of Basic Defeasible Logic. In turn, the engine:

- asserts each fact (as a literal) as a conclusion and removes the literal from the rules, where the literal positively occurs in the body, and 'deactivates' the rules where either of its complements occurs in the body.

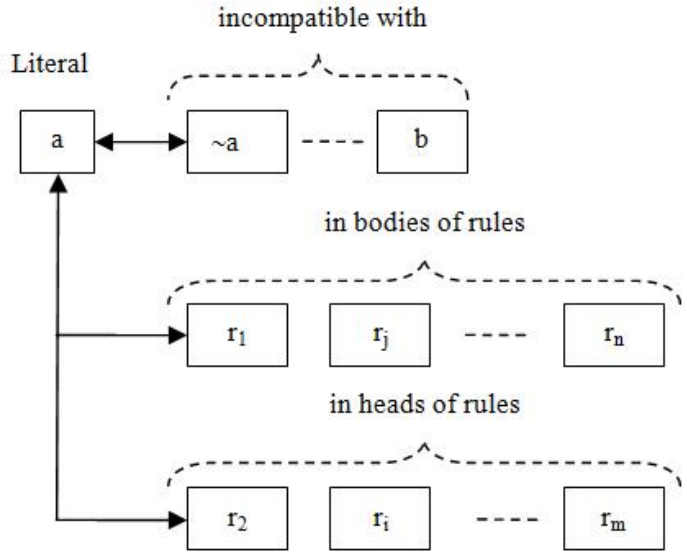

FIGURE 5.4: Data structure for a literal

- scans the list of active rules for the rules with the empty body. Take the literal from the head, remove the rule, and put the literal into the pending facts. The literal is removed from the pending facts and added to the list of facts if either there is no such rule whose head contains the complements of the literal or it is impossible to prove these literals.
- repeats the first step
- terminates when one of the two steps fails.

For $+\Delta$, we have merely to consider similar constructions where we examine only the first parts of steps 1 and 2. It is noticed that this algorithm outputs positive conclusions; negation conclusions can be computed by an algorithm similar to that of the positive with the 'dual actions'. Essentially, for negative conclusions the engine:

1. asserts a literal as a negative conclusion if there is no rule for the literal,
2. scans the list of active rules for the rules have the negative conclusion in the body. These rules are deactivated.
3. Repeats the first step,
4. terminates when one of the two steps fails.

On termination, the algorithm outputs the set of conclusions from the list of facts in the RuleML format.

The extended inference process flattens the superiority relation between the theories to apply Basic Defeasible Logic. Differing from Maher et al. (2001), literals proved in the strict rules can be defeated by definite conclusions derived from the superior theory. Hence, the inference can be used for both strict and defeasible rules by separately investigating those rules. The outcome from processing the strict rules is considered as facts when examining the defeasible rules.

5.5 MDL-MAS: DL-MAS extension with modal notions

As can be observed from a society, individual members can take any action driven by their desires. However, the individuals are often required to comply with the society 'conventions'. Essentially 'conventions' could be norms, constraints or desires that are popularly recognised by the society. Being aware of those conventions, individual members can strengthen their social relationships and coordinate well with other members. Within a group of agents, an agent maintains its social commitments by discovering the 'common attitudes' and fulfils its own demands whilst obligating to these attitudes.

In MDL-MAS, we extend the DL-MAS framework by introducing the modal notions for a finer model of interacting agents. We favour the argument in Governatori and Rotolo (2008), therefore, we create three layers including Belief, Intention and Obligation for every theory in the knowledge structure of an individual agent. The DL-MAS framework enables an agent to discover and approximate the attitudes shared by the majority of the group.

5.5.1 MDL-MAS architecture

Our MDL-MAS has two major components, as shown in Figure 5.5. The first component is the repository of the agents' knowledge, which is built by the designers. To facilitate the interactions between the designers and the agents, the RuleEditor module provides a Java user interface to create defeasible theories representing the knowledge of the individual agents. Once the designers finish composing the sets of knowledge, including the individual's knowledge

and the meta-knowledge of the agents' weights, this knowledge is stored in the RuleML-like repository (see Section 5.5.2). Only the background knowledge and the meta-knowledge are accessible to all agents.

The second component in the dashed-line box in Figure 5.5 presents the essential modules of an individual agent. RuleLoader parses the defeasible theories into Java objects that are suitable for the ReasoningEngine. KnowledgeExchange performs the communication with the other agents in the group. Incoming information is stored in the internal repository as Java objects.

The ReasoningEngine performs the decision-making process by using the extended defeasible reasoning (see Section 5.2.3). Decisions are stored in the internal repository for knowledge exchange or for further investigation. The action module, essentially, provides the connections between an agent and its working environment.

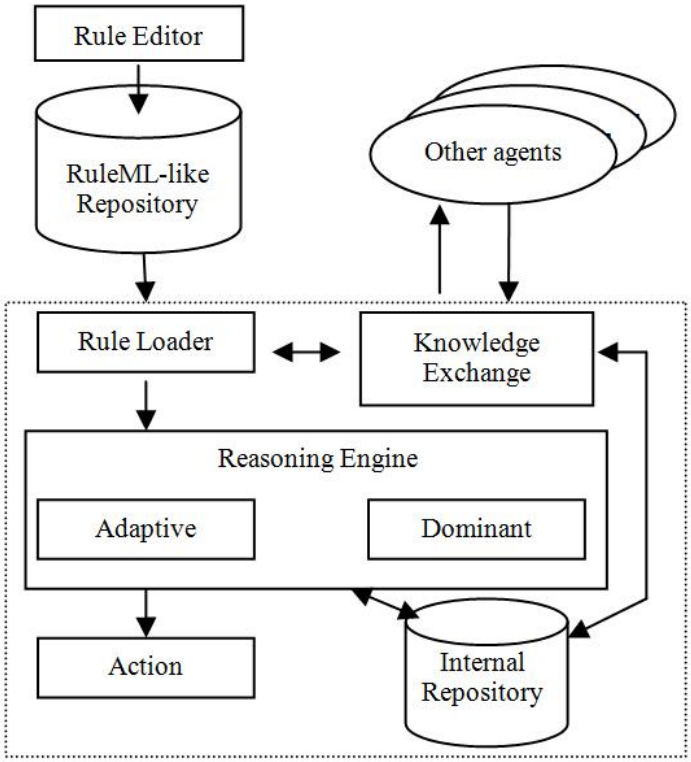

FIGURE 5.5: MDL-MAS architecture

5.5.2 Knowledge representation

As presented in Section 5.2.1, an individual agent has a knowledge structure as a tuple $\mathscr{T} = \{T_{bg}, T_{me}, \mathscr{T}_{other}\}$, whose elements are defeasible theories. To better capture the social actions, we introduce modal notions including Belief, Intention and Obligation. These notions allow the agents to explicitly reason not only about the beliefs of the other agents, but also about their goals, resulting in a stronger social behaviour (Castelfranchi, 1998). Now every $T_i \in \mathscr{T}$ ($T_i \neq T_{bg}$) has two independent sets of defeasible theories T_i^B and T_i^I that represent the set of beliefs and the set of intentions correspondingly. Meanwhile, T_{bg} has all three modal notions $T_{bg} = \{T_{bg}^B, T_{bg}^I, T_{bg}^O\}$. Essentially, the beliefs represent what the agents believe to be true; the intentions represent what the agents want to achieve; and the obligations represent what the agents should commit to the group.

With regard to the implementation, every defeasible theory in the knowledge structure is represented by using a defeasible rule markup with the modal label on top of the theory.

Example 15. There is a man in a sinking boat and three agents, A_1, A_2, A_3, observe the situation, having the weights of $\{6, 3, 1\}$ respectively. Knowledge commonly shared among the agents is:

$$T_{bg}^B = \{R_s = \{r_1 :\rightarrow manOnSinkingBoat\}\}$$
$$T_{bg}^I = \{R_d = \{\ r_1 : manOnSinkingBoat \Rightarrow manInDanger$$
$$r_2 : manInDanger \Rightarrow rescue\}$$
$$T_{bg}^O = \{R_d = \{r_1 : risk \Rightarrow \sim rescue\}\}$$

The background knowledge states that the man is in danger, the rescue should be performed if it is safe to do so. In addition, A_3 knows about the intentions of A_1, A_2 and its own:

$$T_1^I = \{R_d = \{r_1 :\Rightarrow swim, r_2 : swim \Rightarrow risk\}\}$$
$$T_2^I = \{R_s = \{r_1 :\rightarrow throwRope, r_2 : throwRope \rightarrow \sim risk\}\}$$
$$T_{me}^I = T_3^I = \{R_d = \{r_1 :\Rightarrow surf, r_2 : surf \Rightarrow \sim risk\}\}$$

Essentially, the knowledge structure is interpreted as, A_1 wants to swim directly to the sinking boat, while A_2 intends to throw a rope to the boat, and A_3 plans to approach the boat on a surfboard.

5.5.3 Reasoning engine

Social categories

As in Section 5.2.2, an individual agent can adapt to the majority by dropping its own beliefs and intentions in favour of those popularly recognised by the group. However, the agent can dominate the group by promoting its own intentions and rejecting contradictory beliefs and intentions from the majority of the group. In both situations, the obligations from the background knowledge play as 'filter' so that any behaviour violating these obligations is cancelled by the individual agent.

We categorise our agents into two types of social behaviours, entitled 'majority' and 'obedience'. Agents in the first category totally commit to the group, avoiding conflicts with the majority and the group's 'common conventions', represented by T^O_{bg}. These agents collect the majority beliefs and intentions from others by running the reasoning mechanism in Section 5.3 over the belief and intention elements of the knowledge structure, respectively:

1. $T^B_{maj} \ni (T^B_{bg} \ni T^B_{me}) \vdash C'^B_{me}$
2. $T^I_{maj} \ni (T^I_{bg} \ni T^I_{me}) \vdash C'^I_{me}$
3. $\{T^O_{bg}; C'^B_{me}\} > C'^I_{me}$

In contrast, obedient agents only commit to 'common conventions' and perform the reasoning process:

1. $(T^B_{bg} \ni T^B_{me}) \ni T^B_{maj} \vdash C'^B_{me}$
2. $(T^I_{bg} \ni T^I_{me}) \ni T^I_{maj} \vdash C'^I_{me}$
3. $\{T^O_{bg}; C'^B_{me}\} > C'^I_{me}$

Example 16. Reconsidering Example 15, because A_3 does not know about the beliefs of the other agents, the majority belief equals the derivation of the background belief T^B_{bg}. That is $T^B_{maj} = \{+\Delta manOnSinkingBoat\}$.

A_3 identifies intentions of A_1 and A_2 by integrating what A_3 knows with T^I_{bg}, $T^I_{bg} \ni \mathscr{T}^I_i \vdash C^I_i$: $i = 1,2$:

$$C^I_1 = \{+\partial manInDanger^6, +\partial swim^6, +\partial risk^6, +\partial rescue^6\}$$
$$C^I_2 = \{+\partial manInDanger^3, +\Delta throwRope^3, +\Delta{\sim}risk^3, + \partial rescue^3\}$$

The superscript of defeasible conclusions represents the weight inherited from the corresponding knowledge source. The majority intentions from others are:

$$T^I_{maj} = \{+\partial manInDanger^9, +\partial swim^6, +\partial risk^6, +\partial rescue^9\}$$
$$w_{maj} = \{9,6\}$$

The superscript of a majority conclusion shows the weight accumulated from that of the sources supporting the conclusion. Because A_3's weight is the smallest, A_3 adapts its intentions to the majority $I_{A_3} = T^I_{maj}$. In the final step, A_3 drops the intention of doing the rescue because of r_1 in T^O_{bg}.

Suppose the weight of the group changes from $\{6,3,1\}$ to $\{6,3,5\}$. By integrating A_3's intentions with that of the background, $T^I_{bg} \ni T^I_{me}$, A_3 derives:

$$C^I_{me} = \{+\partial manInDanger^5, +\partial surf^5, +\partial{\sim}risk^5, +\partial rescue^5\}$$

Clearly, if A_3 joined the majority pool, the majority conclusions would favour those from A_3. A_3 now rejects conflicts from the majority intentions, T^I_{maj}, and persists with its own intentions with respect to group obligations. That is, obedient agents only maintain their commitments to the group by eliminating the intentions against the obligations specified in the background knowledge.

We believe that the 'majority agents' can express a strong social commitment to the group. Being aware of the others' knowledge, these agents dynamically learn new 'conventions' recognised by the majority and change their intentions toward this knowledge. On the other hand, 'obedient agents' can introduce 'new values' into the group. Thanks to their high weight, these agents could take leading actions so that other agents could follow.

Algorithm

As in the previous section, every agent in the MAS-LM has three knowledge layers corresponding to the Belief, Intention and Obligation notions. Eliminating conflicts with the Obligation layer from the agents' intentions is achieved by the standard defeasible reasoning.

The key component for the MDL-MAS engine is the mechanism for integrating with a superior knowledge source (Section 5.3), which operates on the Belief and Intention layers to determine the beliefs and intentions of the majority. The engine allows an agent either to adapt to or to override the mental attitudes of the majority by implementing adaptive or dominant strategies.

Because of the conflict resolution of the reasoning mechanism, the implementation of the majority rule is straightforward. Therefore, this part focuses on the implementation of the reasoning with the superior knowledge. The algorithm for the reasoning mechanism extended from Maher et al. (2001) takes the theories in the RuleML format as input to create the data structure for the inference process. The inference process flattens the superiority relation between the theories to apply Basic Defeasible Logic. Differing from Maher et al. (2001), literals proved in the strict rules can be defeated by definite conclusions derived from the superior theory. Hence, the inference can be used for both strict and defeasible rules by separately investigating those rules. The outcome from processing the strict rules is considered as facts when examining the defeasible rules.

5.6 Related work

In this section, we relate our work in following perspectives: knowledge representation and reasoning. We show that our approach can maintain the expressiveness and efficiency of defeasible logic in dealing with incomplete situation and conflicting information from different sources. Our reasoning mechanism can work as an information fusion tool and support coordination between agents in the group.

5.6.1 Knowledge representation

In the framework, we favour the logic-based approach to modelling the knowledge of the agents. However, our framework differs from BDI-like agents (Kinny et al., 1996; Rao and Georgeff, 1991) in the consideration of incomplete and conflicting information. Despite the knowledge structure of the framework, which explicitly captures the shared knowledge and the knowledge about the other agents, our agents can only provide an approximate representation of the common knowledge and the 'knowing about what others know'. The formal definitions can be found in Fagin et al. (2003).

Interpreted systems (Fagin et al., 2003) consider multi-agent systems as systems containing n different agents. A global state of the system consists of the agents' states, representing the information accessible by the agents, and an environment state. Knowledge is established if the agents cannot distinguish a given state over all of their runs. Essentially, the knowledge of an agent is captured by modal logics and interpreted as a sequence of snapshots of the world. However, the reasoning mechanism is not designed to cope with incomplete and conflicting information.

With regard to incomplete and conflicting information, Sakama et al. (2000) design multi-agent systems, whose individual agent contains three knowledge components, knowledge base, default information (assumptions) about others and responses from others. The knowledge base is a logic program, whose predicates can be commonly shared by the agents. The default information is consistent with the logic program, but possibly conflicts with the responses. The belief of an agent corresponds to the maximum answer-set computed from an extended logic program, whose predicates are transformed from the knowledge components by using situation calculus. Defeasible reasoning is captured by the rule: counter-evidence obtained from the responses rebuts the previous assumptions. The computational cost of the mechanism is dominated by the computing answer-sets from the extended logic program.

Rotstein et al. (2007) propose to use defeasible logic programs (García and Simari, 2004) to represent the BDI agents. The knowledge portion of an agent is represented as normal logic programs provided this portion does not contain conflicting information. Meanwhile, the knowledge portion with the possible conflicts is represented by the defeasible logic programs. These

conflicts can be solved by dialectical analysis over arguments for and against a piece of information. A set of filters, expressed as logic programs, is used to eliminate the impractical desires and intentions. During the reasoning process, these filter programs are merged with the belief programs using the operator of Fuhrmann (1997). Hence, the approach defines the different types of agents according to the logic programs filtering desires and intentions.

One major stream in the multi-agent systems to capture social commitments is to modify the BDI architecture (Rao and Georgeff, 1991) by introducing deontological properties such as, laws, norms and obligations to place constraints on the agents' behaviours. The deontological properties are considered as external influences on an individual's decision-making and the commitment to other members. This idea is supported by several authors in (Broersen et al., 2001; Castelfranchi et al., 2000; Dignum et al., 2002).

Clearly modal logics are very powerful in representing these concepts. Our approach differs from the BDI-like agents in the consideration of the incomplete and conflicting information. The social commitment is implemented by pondering the conflicts with the 'desires' commonly shared by the group and the 'desires' shared by the majority. That is, our agents demonstrate a social ability via their commitments to the beliefs and goals of the group (Castelfranchi, 1998).

In our approach, the agents generally adjust their behaviour to the majority attitudes, which are dynamically discovered during the interactions with other agents. However, if an agent has a strong belief contrary to the common (shared) desires, the agent can break its commitment. This exception can be against the goals of the group, but offers the agent some levels of autonomy and flexibility in making a decision.

5.6.2 Reasoning mechanism

From the perspective of the information fusion based on logic programs, our mechanism significantly differs from the simple union of logic programs (surveyed in Brogi (2004)) and the semantic-based approaches (reviewed in Grégoire and Konieczny (2006)) owing to the non-monotonic nature of defeasible logic. The reasoning mechanism with the superior theory offers the main method for combining the knowledge sources (represented as logic programs). The majority is considered as a special source of knowledge that is superior to the internal

knowledge of the individual agent. The extended mechanism guarantees the consistency of the computed conclusions, but does not limit those conclusions to the knowledge of the majority. Notably, the sources of the knowledge are still intact after integration. Furthermore, the computational cost of our mechanism is in the $O(n)$ class, linearly proportional to the size of the knowledge base of the individual agents.

Our framework uses the majority rule to combine the conclusions from different sources representing the belief of corresponding agents. The well-known problem for the majority rule is doctrinal paradox or discursive dilemma (Kornhauser and Sager, 1986). Two procedures have been proposed to solve the paradox: premises-based and conclusions-based voting (Bovens and Rabinowicz, 2006; Chapman, 2002; Pettit, 2001). Pigozzi (2006) criticises these methods, because they do not consider the logical connections between premises. Therefore, Pigozzi introduces the argument-based procedure to tackle the problem. This procedure applies a model-based aggregation (Grégoire and Konieczny, 2006) to merge belief sets and includes a set of logical relations between premises and conclusions (can be seen as rules) to remove models leading to the paradox.

Our framework does not suffer the paradox if all agents are aware of the logical relations. That is the logical relations are commonly shared among these agents (background knowledge). Suppose that three agents with equal weight largely support p, q, and $\sim r$ ($\sim r$ derived by a rule in the individual beliefs of agents) and these agents commonly share the rule $r_1 : p \wedge q \rightarrow r$. Once an agent obtains the majority knowledge, the agent continues the reasoning process by applying the extended defeasible reasoning. This reasoning process rejects the majority support for $\neg r$ and retains r due to r_1 and the superiority of the background knowledge over the majority knowledge. The background knowledge in our framework plays the role of integrity constraints as in (Pigozzi, 2006). However, the main differences are:

- Our agents have belief bases that can contain incomplete and conflicting information,
- Our agents do not know exactly how other agents tackle with the conflicting information.

Retrieving a consistent belief base by the majority procedure is not a trivial task. The performance of the majority rule can depend on the type of the belief bases. It can work well with perceptual beliefs but cannot for those beliefs are strongly judged by other information in the

base (Pettit, 2006). In our framework, the outcome of the reasoning process is always consistent with the background knowledge. It is noted that the outcome of individual agents cannot always unify because of the non-monotonicity of the logic and the partial view of agents. If an agent believes that it can dominate the group, the agent does not use the majority rule but merges the belief bases by the order of weight. This strategy allows the agent to enrich its beliefs by information from the other agents.

Our work shares several similarities with the modal logic framework (Liau, 2005), in particular the meta-structure of the agents' knowledge and the reasoning strategies. This framework relies on combining the multi-agent epistemic logic and the multi-source reasoning systems (Cholvy, 1994) to reason about the multi-agent knowledge with the levels of reliability and its fusion. Two cautious strategies are devised, namely, *level cutting fusion* – the agent rejects all the conflicting beliefs of those having a lower level of reliability; and *level skipping fusion* – only the level having a conflicting belief is discarded to obtain the maximal consistent subset combination.

The framework in Liau (2005) does not differ from ours in the fusion techniques, but in the conflict handling. Because of the use of the defeasible logic, our agents' knowledge can contain conflicting information in a single source, but the information is consistent at the end of the reasoning process. Also, the conflicts between the sources can be resolved by further exploiting the superiority relation.

Another work related to our mechanism in the context of defeasible reasoning from multiple sources is the framework for the defeasible policy composition by Lee et al. (2006). In the framework, each security policy associates with a meta-policy, which describes the enforced requirements and composition preferences in the language of defeasible logic. The results of the reasoning process, to some extent, are similar to our framework when our agents have to run the sequence of reasoning with superiority theory to handle the lack of the majority knowledge. Essentially, this work can be classified as a new approach to the problem of merging knowledge bases. From defeasible reasoning, the framework performs the reasoning process over one single set of defeasible theory, whereas our agents have to cope with multiple defeasible theories. The distinctive result of the framework is the function, which automatically creates the superiority relation between the meta-policies from the hierarchy of policies reflecting the

structure.

The intuition of solving the conflicts among the knowledge sources by examining the support level for each conflicting information is also found in argumentation systems (Amgoud and Kaci, 2007). Our mechanism relies on the weight of the sources and the ambiguity blocking feature of defeasible logic (Governatori et al., 2004) to cope with the conflicts within/among the sources. That is, instead of generating all possible arguments as in Amgoud and Kaci (2007), arguments are aggregated incrementally by using the weight of the sources. After each step, only the 'stronger arguments' are forwarded to the next step. Blocking ambiguities enables our mechanism to obtain the low computation cost.

With regard to the reasoning process, our framework supports the discovery of the information consistently and is widely supported by many members in the group. Also, the approximate distributed knowledge can be explored by the sequence of reasoning with the superiority theory when it is impossible to retrieve the majority knowledge. It is noticed that both the common and distributed knowledge mentioned in both frameworks have differences from those presented in Fagin et al. (2003), because that knowledge are only approximate. Both reasoning processes do not handle the conflicts and possible inconsistence in the logic theories.

The reasoning mechanism of our framework is coordination-oriented. The behaviour of an individual agent can be motivated by the joint knowledge from other agents or by the integration of all the possible consistent knowledge from the others. This result can be interpreted as rigorous and generous coordination (Sakama and Inoue, 2005) in the case where our agents can obtain complete knowledge of the others. For general cases, the background knowledge can contribute to the success of coordination, because this knowledge is commonly known and enforced by the individual agents. In exceptional cases, an individual agent can refuse to defer to the superior knowledge, such as the majority knowledge, if the agent has a strong belief to the contrary (definitely provable information). This exception can break the coordination, but offers the agent some levels of autonomy and flexibility in making a decision.

5.7 Summary

This chapter has presented a defeasible logic framework DL-MAS for multi-agent systems that explicitly captures the background knowledge of the group and the knowledge about other agents. The joint knowledge from the other agents is a special source, because this knowledge is largely shared by the group. Considering the conflicts between the internal knowledge and this special knowledge, an individual agent can balance its mental attitudes with the 'desires' shared among the group.

We extended the reasoning mechanism by incorporating two reasoning strategies, that is, the adaptive and collective strategies. An agent can either adapt to the behaviour of the majority or collect consistent information as much as possible from other agents. The agent can have a distinct behaviour from the group in the case that it holds strong evidence contrary to the knowledge of the other agents. One important feature of the framework is maintaining the computational efficiency from both the defeasible reasoning and the majority rule. The complexity of the framework is linearly proportional to the size of the knowledge base of an individual agent.

We propose a layer approach in the MDL-MAS framework. By introducing modal notions into the knowledge structure of an agent, the layer approach is a trade-off between the 'expressibility' and the tractability. An agent can represent what it knows about the modal notions of the other agents, but not what it knows about what others know. The complexity of the extended mechanism is a linear proportion to the number of the modal notions. Being aware of the beliefs and goals of the other agents, an individual agent can discover mental attitudes, which are largely shared by the group, and balance those attitudes with its own. In our MDL-MAS, the agents are categorised into two types, majority and obedience depending, on the reasoning strategies they use. An agent can either adapt to the majority behaviours or dominate the group. In the second strategy, the agent can introduce a distinct behaviour that would lead other agents, while committing to obligations commonly recognised by the group.

6

n-Person Argumentation Game: an application

Argumentation games have proved to be a robust and flexible tool to resolve the conflicts among the agents. An agent can propose its explanation and its goal, known as a claim, which can be refuted by other agents. The situation is more complicated when there are more than two agents playing the game.

In this chapter, we propose a weighting mechanism for competing premises to cope with the conflicts from multiple agents in an n-person game. An agent can defend its proposal by giving a counter-argument to change the 'opinion' of the majority of opposing agents. In addition, an agent, whose claim is accepted by the group, can take a 'preventive' action by attacking other accepted arguments whose combination can result in damaging its main claim. During the game, our agent can exploit the knowledge that the other agents expose to promote and

defend its main claim.

6.1 Introduction

In a group of agents, there are several situations requiring the agents to settle on a common goal, despite that, each agent can pursue its goals, which may conflict with other agents' goals. A simple but efficient method to solve the problem is to give weights to the goals. However, this method is not robust and limits the autonomy of an individual agent. Also, conflicts among the agents are likely to arise from a partial view and incomplete information on the working environment of the individual agents. To settle conflicts among the agents, an agent can argue to convince others about its pursued goal and provide evidence to defend its claim.

This interaction between the agents can be modelled as an argumentation game (Amgoud et al., 2007; Jennings et al., 1998; Parsons and McBurney, 2003; Prakken and Sartor, 1996). In general, an agent can propose an explanation for its pursued goal (an argument), which can be rejected if other agents provide counter-evidence. This interaction can be iterated until an agent (the winner) successfully argues its proposal against the other agents. The argumentation game approach offers a robust and flexible tool for the agents to resolve conflicts by evaluating the status of the arguments. The argumentation semantics by Dung (1995) is widely recognised for establishing relationships among arguments. The key notion for a set of arguments is whether or not a set of arguments is self-consistent and provides the base for deriving a conclusion. A conclusion is justified, and thus provable, if there is a set of supporting arguments and all the counter-arguments are deficient when we consider the arguments in the set of the supporting arguments.

An argumentation game is more complicated when the number of participants is greater than two. It is not clear how to extend the existing approaches to cover the argumentation in groups of more than two agents, especially when the agents are equally trustful. That is, the arguments from the individual agents have the same weight. In this case, the problem amounts to how to decide which argument has precedence over the competitive arguments. In other words, the problem is to determine the global collective preference of a group of agents.

The main idea behind our approach concerns where the individual preferences of the agents

are not sufficient to solve a conflict (for example, we have several arguments without any relative preference over them). The group of agents uses the majority rule (Lin, 1996) over the initial proposals to determine the 'most common' claim known as the 'topic' of the dialogue, that is, the topic preferred by the majority of the group. An agent either supports the topic or defends its own claim against the topic. Our majority mechanism simplifies the complexity of the n-persons argumentation and provides a strategy for an agent to select an argument for defending its proposal. An argument causing more 'supporters' to reconsider 'their attitude' will be preferred by the defending agents.

Also, each of our agents is equipped with its private knowledge, the background knowledge and the knowledge obtained from the other agents. The background knowledge represents the expected behaviour of a member of the group, which is commonly shared by the group. The knowledge about the other agents increases during the interactions and enables an agent to efficiently convince others about its own goal. Essentially, the background knowledge is preferred over other sources, because it represents the common expectations and constraints of the group. Any argument violating the background knowledge is not supported by the group.

Defeasible logic is chosen as our underlying logic for the argumentation game owing to its efficiency and simplicity in representing incomplete and conflicting information. Furthermore, the logic has a powerful and flexible reasoning mechanism (Antoniou et al., 2000a; Maher et al., 2001) that enables our agents to capture the argumentation semantics of Dung (1995) by using two features of defeasible reasoning, namely, the ambiguity propagating (the preference over conflicts is unknown) and the ambiguity blocking (the preference is given).

The rest of the chapter is structured as follows. In Section 6.2, we present the construction of arguments using defeasible reasoning with respect to (w.r.t) ambiguous information. Section 6.3 presents the external model of n-person argumentation, which describes a basic procedure for an interaction between the agents. The external model also defines the majority arguments supporting the goal accepted the majority of the agents. Section 6.4 defines the internal model of n-person argumentation. The internal model illustrates how an agent can integrate its private knowledge with other sources either to defend or to convince other agents about its own goal. Section 6.5 provides an overview of the research works related to our approach. Section 6.6 concludes the chapter.

6.2 Argument construction w.r.t defeasible logic

In what follows, we briefly introduce the basic notions of an argumentation system using defeasible logic as underlying logical language. Moreover, we present the acceptance of an argument w.r.t Dung's semantics. A detailed exploration is found in Governatori et al. (2004).

6.2.1 Arguments and defeasible proofs

In general, arguments are defined to be proof trees (or monotonic derivations) from a logical theory.

Definition 14. *An argument A for a literal p based on a set of rules R is a (possibly infinite) tree with nodes labelled by literals such that the root is labelled by p and for every node with the label h:*

1. *If $b_1, \ldots, b_n$ label the children of h then there is a rule in R with body $b_1, \ldots, b_n$ and head h.*

2. *If this rule is a defeater then h is the root of the argument.*

3. *The arcs in a proof tree are labelled by the rules used to obtain them.*

The literal p is also known as the conclusion supported by A.

However, defeasible logic requires a more general notion of a proof tree that admits infinite trees, so that the distinction is kept between an unrefuted, but infinite, chain of reasoning and a refuted chain. Depending on the rules used, there are different types of arguments:

- A supportive argument is a finite argument in which no defeater is used.

- A strict argument is an argument in which only strict rules are used.

- An argument that is not strict is called defeasible.

Relationships between two arguments A and B are determined by those of the literals being composed of these arguments. An argument A *attacks* a defeasible argument B if a conclusion

of A is the complement of a conclusion of B, and that conclusion of B is not part of a strict sub-argument of B. A set of arguments $\mathscr{S}$ attacks a defeasible argument B if there is an argument A in $\mathscr{S}$ that attacks B.

A defeasible argument A is supported by a set of arguments $\mathscr{S}$ if every proper sub-argument of A is in $\mathscr{S}$. A defeasible argument A is *undercut* by a set of arguments $\mathscr{S}$ if $\mathscr{S}$ supports an argument B attacking a proper non-strict sub-argument of A.

The notion of undercut by $\mathscr{S}$ means that some premises of A cannot be proved if we accept the arguments in $\mathscr{S}$.

It is noticed that the concepts of the attack and the undercut concern only defeasible arguments and their sub-arguments; for strict arguments, we stipulate that they cannot be undercut or attacked.

6.2.2 Argument status

It is critical for argumentation systems to determine if an argument is acceptable within a set of arguments. Essentially, an argument is assessed as valid if we can show that the premises of all arguments attacking it cannot be proved from the valid arguments in $\mathscr{S}$. The concept of provability depends on the methods used by the reasoning mechanism to cope with ambiguous information. According to the features of the defeasible reasoning, we have the definition of acceptable arguments (Definition 15).

Definition 15. *An argument A for p is acceptable w.r.t. a set of arguments $\mathscr{S}$ if A is finite, and:*

1. *If reasoning with the ambiguity propagation is used, (a) A is strict, or (b) every argument attacking A is attacked by $\mathscr{S}$.*

2. *If reasoning with the ambiguity blocking is used: (a) A is strict, or (b) every argument attacking A is undercut by $\mathscr{S}$.*

The status of an argument is determined by the concept of acceptance. If an argument can resist a reasonable refutation, this argument is justified as in Definition 16. If an argument cannot overcome attacks from other arguments, this argument is rejected as in Definition 17.

Definition 16. *Let D be a defeasible theory. We define J_i^D as follows:*

- $J_0^D = \emptyset$
- $J_{i+1}^D = \{a \in Args_D |\ a \text{ is acceptable w.r.t } J_i^D\}$

The set of justified arguments in a defeasible theory D is $JArgs^D = \bigcup_{i=1}^{\infty} J_i^D$.

Definition 17. *Let D be a defeasible theory and* $\mathscr{T}$ *be a set of arguments. We define* $R_i^D(\mathscr{T})$ *as follows.*

- $R_0^D(\mathscr{T}) = \emptyset$
- $R_{i+1}^D(\mathscr{T}) = \{a \in Args_D |\ a \text{ is rejected by } R_i^D(\mathscr{T}) \text{ and } \mathscr{T}\}$.

The set of rejected arguments in a defeasible theory D w.r.t. $\mathscr{T}$ *is* $RArgsD(\mathscr{T}) = \bigcup_{i=1}^{\infty} R_i^D(\mathscr{T})$.

6.2.3 Argumentation semantics and the extended reasoning

This section shows the properties of arguments built by the extended reasoning mechanism (See Chapter 5, Section 5.2.3). First, we revise the proof for strict conclusions derived from the combination of the theories, therefore, strict arguments. Regarding the standard defeasible reasoning, we introduce the notion of defeasibility into the strict part of the combined theory. Essentially, a strict argument can be rejected if and only if that argument is constructed from a theory with a lower priority. Otherwise, the argument is not rejected by any argument. We define the acceptance of a strict argument w.r.t. the extended reasoning in Definition 18.

Definition 18. *In the extended reasoning, a strict argument A for p is strictly acceptable w.r.t. a set of strict arguments* $\mathscr{S}$ *if A is finite, and every argument attacking A is undercut by* $\mathscr{S}$.

We now present the property of strict arguments constructed by the extended reasoning over two defeasible theories T_{sp} and T_{in}, where T_{sp} has priority over T_{in}.

Proposition 6. *Let* T_{sp} *and* T_{in} *be defeasible theories such that* $T_{sp} \succ T_{in}$ *and p be a literal.*

1. $T_{sp} \ni T_{in} \vdash +\Delta p$ *iff there is a strictly acceptable argument for p from* $T_{sp} \ni T_{in}$.
2. $T_{sp} \ni T_{in} \vdash -\Delta p$ *iff there is no strictly acceptable argument for p from* $T_{sp} \ni T_{in}$.

Proof. We prove the only if ($\Rightarrow$) direction of the proposition by induction on the length of derivation P of the extended reasoning over T_{sp} and T_{in}.

At the first step of the derivation, $P(1) = +\Delta p$. That implies there is a strict rule, r, supporting p in $T_{sp} \ni T_{in}$ (Note that a fact can be considered as a strict rule with an empty body). If r is in T_{sp}, there is a strict argument for p constructed from T_{sp}. That argument is self acceptable within T_{sp}, because of the priority of T_{sp}. If r is in T_{in}, there is a strict supportive argument A for p constructed from T_{in}. Within T_{in}, there is no argument against A as the standard reasoning. Corresponding to the extended condition for $+\Delta$, $P(1)$ holds only if there is no strict rule in T_{sp} supporting $\sim p$. In other words, there is no argument supporting $\sim p$ constructed from T_{sp}. Therefore, the argument A from T_{in} is acceptable w.r.t T_{sp}.

At the first step, if $P(1) = -\Delta p$ then there is no strict rule r supporting p in both T_{sp} and T_{in}. Therefore, it is not possible to have a strict argument for p in both theories.

At the inductive step, we assume that the proposition holds for the derivation with the length up to n. $P(n+1) = +\Delta p$. That is there exists a supportive argument A for p, which is built from a strict rule $r \in T_{sp} \cup T_{in}$ such that $\forall a_r \in A(r), +\Delta a_r \in P(1..n)$. If r is in T_{sp}, A is a strict argument in T_{sp}. According to the standard defeasible reasoning, the strict argument A is self acceptable within T_{sp}. If $r \in T_{in}$, every a_r must be justified by inductive hypothesis. In addition, every literal in the bodies of the strict rules for $\sim p$ in T_{sp} does not have a strictly positive proof: $\forall r \in R_s^{sp}[\sim p]\ \exists a \in A(r) : -\Delta a \in P(1..n)$. By the inductive hypothesis for the extended negative proof, there is not any strict argument supporting these literals in T_{sp}. Therefore, the argument A is acceptable.

Assume that $P(n+1) = -\Delta p$, there are two possibilities. First, each strict rule r for p in T_{sp} and T_{in} has at least one literal a_r in the body such that $-\Delta a_r \in P(1..n)$. By the inductive hypothesis, there is no strict argument for a_r; therefore, it is not possible to built a strict proof for p from T_{sp} and T_{in}. Second, there is a rule in T_{sp} supporting the complement of p. By the inductive hypothesis for the positive proof, there is a strictly acceptable argument for $\sim p$. Hence, all of the arguments for p (from T_{in})is not acceptable.

In what follows, we prove the if direction ($\Leftarrow$) of the proposition.

In the first part of the proposition, suppose that A is a strict argument for p having the height of 1. If A is built from T_{sp}, there is a strict rule with an empty body for p in the combination

$T_{sp} \cup T_{in}$. If A is built from T_{in} and accepted by T_{sp}, there is a strict rule for p in T_{in} and no rule for $\sim p$ in T_{sp}. In both case, there is an applicable rule for p in the combination, therefore, $T_{sp} \ni T_{in} \vdash +\Delta p$.

At the inductive step, we assume that the first part holds for arguments with heights up to n and A is an argument for p. From A, we construct a strict rule as $A(r) \rightarrow p$. For every literal $a_r \in A(r)$, which is accepted by T_{sp}, we create sub-arguments of A having the height less than n. By the inductive hypothesis we obtain $+\Delta a_r$, hence, the condition for $A(r) \rightarrow p$ is satisfied. Therefore, $T_{sp} \ni T_{in} \vdash +\Delta p$.

The second part of the proposition is proved by the contradiction. Assume that $T_{sp} \ni T_{in} \nvdash -\Delta p$. That leads to:

1. $r \in R_s^{sp}[p]$ $\forall a_r \in A(r)$ $T_{sp} \ni T_{in} \nvdash -\Delta a_r$ or
2. $s \in R_s^{in}[p]$ $\forall a_s \in A(s)$ $T_{sp} \ni T_{in} \nvdash -\Delta a_r$ and $\forall t \in R_s^{sp}[\sim p]$ $\exists a_t \in A(t)$ $T_{sp} \ni T_{in} \nvdash +\Delta a_t$.

For a strict rule for p in T_{sp}, we construct a partial argument A for p by expanding r. The expansion of the argument ends with three instances:

1. A rule with the empty body. That is there a strict argument for the literal from T_{sp}. That contradicts the assumption.
2. No more rule to expand, therefore, we have $-\Delta a_r$. That also contradicts the assumption.
3. A loop. None of the literals of the loop can prove the adjacent literal. Therefore, we have $-\Delta a_r$. That also contradicts the assumption.

For a strict rule for s in T_{in}, we construct a partial argument B for p by expanding s. Considering an argument C attacking B at q. If q is supported by T_{sp}, the attack is rejected because of the priority of T_{sp}. Hence, q is supported by a rule T_{in}. If E for $\sim q$ is constructed from T_{sp}, then that violates the assumption $\forall t \in R_s^{sp}[\sim q]$ $\exists a_t \in A(t)$ $T_{sp} \ni T_{in} \nvdash +\Delta a_t$. If E for $\sim q$ is derived from T_{in}, that violates the coherence and consistency of the strict part of T_{in}. Therefore, E is not an acceptable argument. Also, B is not attacked by any argument.

For both cases, the assumption is not valid; thus, the second part of the proposition is proved.

□

Example 17. This example shows the result of extending the superiority relation between theories to the strict parts of defeasible theories. Suppose that we two defeasible theories T_{sp} and T_{in} such that $T_{sp} \succ T_{in}$:

$$T_{sp} = \{R_s = \{r_1 :\rightarrow a;\ r_2 : a \rightarrow b\}\}$$
$$T_{in} = \{R_s = \{r_1 :\rightarrow c;\ r_2 : c \rightarrow \sim b\}\}$$

The extended $T_{sp} \ni T_{in}$ reasoning proves $+\Delta a$, $+\Delta b$, $+\Delta c$, and $-\Delta\sim b$. Correspondingly, the combined theory justifies the strict arguments $JArgs_{T_{sp} \ni T_{in}} = \{\rightarrow a \rightarrow b;\ \rightarrow c\}$. Owing to the priority of T_{sp} over T_{in}, the argument $\rightarrow c \rightarrow \sim b$ is rejected and undercut by $\rightarrow a \rightarrow b$. Therefore, there is no justified argument for $\sim b$.

The combination of $T_{sp} \ni T_{in}$ extends the priority among the defeasible theories to that of rules in the combination. Therefore, the set of arguments constructed from the combination inherits the justification property from that of the standard defeasible logic (see Theorem 17 in Governatori et al. (2004)). This is also owing to the coherent property of the extended conditions for strictly provable conclusions.

Proposition 7. *In the combination of two independent theories $T_{sp} \ni T_{in}$*

1. *$T_{sp} \ni T_{in} \vdash +\partial p$ iff arguments for p are justified by $T_{sp} \ni T_{in}$. Must be considered the strict part of T_{in}.*

2. *$T_{sp} \ni T_{in} \vdash -\partial p$ iff arguments for p are rejected.*

Owing to coherent and consistent properties of the extended defeasible reasoning, the set of arguments constructed from the combination of two independent theories satisfies the proposition 8. The extended reasoning over the combination does not simultaneously provide proof for $\pm\partial p$ or $\pm\Delta p$. As a result, it is not possible to construct the arguments both for and against a literal and its complement.

Proposition 8. *In the integration of two defeasible theories $T_{sp} \ni T_{in}$*

1. *No argument is both justified and rejected.*

2. *No literal is both justified and rejected.*

6.3 External model of n-person argumentation

In an argumentation game, a group of agents $\mathscr{A}$ shares a set of goals $\mathscr{G}$ and a set of external constraints T_{bg} represented as a defeasible theory, known as a background knowledge. This knowledge provides common expectations and restrictions in $\mathscr{A}$. An agent has its own view of the working environment, therefore, it can autonomously pursue its own goal.

In this section, we model the interactions between the agents to settle on the goals commonly accepted by the group. Also, at each step of the game, we show how an agent can identify a goal and the sub-goals for its counter arguments. This information is critical for those agents whose main claims are refuted either directly by arguments from other agents or indirectly by the combination of these arguments. The external model presents a simple mechanism for an agent to select its argument, which relies on the majority rule.

6.3.1 Settling on common goals

An agent can pursue a goal in the set of common goals $\mathscr{G}$ by proposing an explanation for its goal. The group justifies proposals from individual agents to identify commonly-accepted goals using a dialogue as follows:

1. Each agent broadcasts an argument for its goal. The system can be viewed as an argumentation game with n-players corresponding to the number of agents.
2. An agent checks the status of its argument against those from the other agents. There are three possibilities:
   (a) *Directly refuted* if its argument conflicts with those from others
   (b) *Collectively refuted* if its argument does not conflict with individual arguments but violates the combination of individual arguments (See Section 6.4.2)
   (c) *Collectively accepted* if its argument is justified by the combination (see Section 6.4.3).
3. According to the status of its main claim, an agent can, (a) defend its claim, (b) attack a claim from other agents, (c) rest. An agent repeats the previous step until the termination conditions of the game are reached.

4. The dialogue among agents is terminated if all agents can pass their claims. For a dispute, agents stop arguing if they do not have any more argument to propose.

6.3.2 Weighting opposite premises

In a dialogue, at each iteration an agent is required to identify the goals and sub-goals that are largely shared by other agents. This information is highly critical for those agents whose main claims are refuted either directly by other agents or collectively by the combination of arguments from others to effectively convince other agents. To achieve that, an agent, A_{me}, identifies a sub-group of agents, namely an 'opp-group', which directly or collectively attacks its main claim. A_{me} creates $Args^{opp}$ as the set of opposing arguments from the opp-group and P^{opp} as the set of premises in $Args^{opp}$. Essentially, $Args^{opp}$ contains arguments attacking A_{me}'s claim. Each element of P^{opp} is weighted by its frequency in $Args^{opp}$. We define the preference over P^{opp} as given $p_1, p_2 \in P^{opp}$, $p_2 \succeq p_1$ if the frequency of p_2 in $Args^{opp}$ is greater than that of p_1. Basically, the more frequent an element $q \in P^{opp}$, is the more the agents use this premise in their arguments. Therefore the refutation of q challenges other agents better than the premises having lower frequency, because this refutation causes a larger number of agents to reconsider their claims.

6.3.3 Defending the main claim

At iteration i, $Args_i^{opp}$ represents the set of arguments played by the opp-group:

$$Args_i^{opp} = \bigcup_{j=0}^{|\mathscr{A}|} Args_i^{A_j} | Args_i^{A_j} \text{ directly attacks } Args_i^{A_{me}}$$

where $Args^{A_j}$ is the argument played by agent A_j. If A_j rests at iteration i, its last argument (at iteration k) is used $Args_i^{A_j} = Args_k^{A_j}$. The set of opposite premises at iteration i is:

$$P_i^{opp} = \{p | p \in Args_i^{opp} \text{ and } p \notin Args_i^{A_{me}}\}$$

The preference over elements of P^{opp} provides a mechanism for A_{me} to select arguments for defending its main claim.

Example 18. Suppose that agent A_1 and A_2 respectively propose $Args^{A_1} = \{\Rightarrow e \Rightarrow b \Rightarrow a\}$ and $Args^{A_2} = \{\Rightarrow e \Rightarrow c \Rightarrow a\}$ whilst agent A_3 claims $Args^{A_3} = \{\Rightarrow d \Rightarrow \sim a\}$. From A_3's view, its claim directly conflicts with those of A_1 and A_2. The arguments and premises of the opp-group are:

$$Args_i^{opp} = \{\Rightarrow e \Rightarrow b \Rightarrow a;\ \Rightarrow e \Rightarrow c \Rightarrow a\}$$
$$P_i^{opp} = \{a^2, b^1, c^1, e^2\}$$

The superscript of elements in P_i^{opp} represents the frequency of a premise in $Args_i^{opp}$. A_3 can defend its claim by providing a counter-argument that refute $\sim a$ – the major claim of the opp-group. Alternatively, A_3 can attack either b or c or e in the next step. An argument against e is the better selection compared with those against b or c since A_3's refutation of e causes both A_1 and A_2 to reconsider their claims.

6.3.4 Attacking an argument

In this situation, the individual arguments of the other agents do not conflict with that of A_{me} but the integration of these arguments does. Agent A_{me} should argue against one of these arguments to convince others about its claim.

At iteration i, let the integration of arguments be $T_{INT}^i = T_{bg} \bigcup_{j=0}^{|\mathscr{A}|} T_j^i$, where T_j^i is the knowledge from agent j supporting agent j's claim, and $JArgs^{T_{INT}^i}$ be the set of justified arguments from integrated knowledge of other agents (see Section 6.4.3). The set of opposite arguments is defined as:

$$Args_i^{opp} = \{a | a \in JArgs^{T_{INT}^i} \text{ and } a \text{ is attacked by } Args_i^{A_{me}}\}$$

and the set of opposite premises is:

$$P_i^{opp} = \{p | p \in Args_i^{opp} \text{ and } (p \notin Args_i^{A_{me}} \text{ or } p \text{ is not attacked by } Args_i^{A_{me}})\}$$

The second condition is to guarantee that A_{me} is self-consistent and does not play any argument against itself. To convince the other agents about its claim, A_{me} is required to provide arguments against any premise in P^{opp}. In fact, the order of the elements in P^{opp} offers a guideline for A_{me} for selecting its attacking arguments.

6.4 Internal model of n-person argumentation

This section presents the internal model of an individual agent participating in an n-person argumentation. In particular, the internal model defines the knowledge structure of an agent and uses the argumentation semantics in Section 6.2 to build up the set of arguments w.r.t the knowledge from other agents being exposed after every step of the dialogue.

6.4.1 Knowledge representation

Agent A_{me} has three types of knowledge including the background knowledge T_{bg}, its own knowledge about the working environment T_{me}, and the knowledge about others:

$$\mathscr{T}_{other} = \{T_j : 1 \leq j \leq |\mathscr{A}| \text{ and } j \neq \text{me}\}$$

where T_j is obtained from agent A_j during iterations and T_j is represented in DL. At iteration i, the knowledge obtained from A_j is accumulated from the previous steps:

$$T_j^i = \bigcup_{k=0}^{i-1} T_j^k + Args_i^{A_j}$$

In our framework, an agent's knowledge can be rebutted by other agents. It is reasonable to assume that defeasible theories contain only defeasible rules and defeasible facts (defeasible rules with an empty body).

6.4.2 Knowledge integration

To generate arguments, an agent integrates knowledge from different sources. Given ambiguous information between two sources, there are two possible methods for combining them: ambiguity blocking is selected if the preference between these sources is known; otherwise, ambiguity propagation is applied.

Ambiguity blocking integration

This method extends the standard defeasible reasoning by creating a new superiority relation from that of the knowledge sources, that is, given two sources as T_{sp} – the superior theory, and

T_{in} – the inferior theory, we generate a new superiority relation $R_{\mathrm{d}}^{\mathrm{sp}} > R_{\mathrm{d}}^{\mathrm{in}}$ based on the rules from the two sources. The integration of the two sources is denoted as $T_{INT} = T_{sp} \ni T_{in}$. Now, the standard defeasible reasoning can be applied for T_{INT} to produce a set of arguments $Args_{\mathrm{AB}}^{T_{INT}}$.

Example 19. Given two defeasible theories

$$\begin{aligned} T_{bg} = \{R_d = \{\ & r_1 : e \Rightarrow c; \\ & r_2 : g, f \Rightarrow \sim c; \\ & r_3 : \Rightarrow e\}; \\ > = \{r_2 > r_1\}\} & \end{aligned}$$

$$\begin{aligned} T_{me} = \{R_d = \{\ & r_1 : \Rightarrow d; \\ & r_2 : d \Rightarrow \sim a; \\ & r_3 : \Rightarrow g\}\} \end{aligned}$$

The integration of $T_{bg} \ni T_{me}$ produces:

$$\begin{aligned} T_{INT} = \{R_d = \{\ & r_1^{T_{bg}} : e \Rightarrow c; \\ & r_2^{T_{bg}} : g, f \Rightarrow \sim c; \\ & r_3^{T_{bg}} : \Rightarrow e; \\ & r_1^{T_{me}} : \Rightarrow d; \\ & r_2^{T_{me}} : d \Rightarrow a; \\ & r_3^{T_{me}} : \Rightarrow g\}; \\ > = \{r_2^{T_{bg}} > r_1^{T_{bg}}\}\} & \end{aligned}$$

The integrated theory inherits the superiority relation from T_{bg}. That means the new theory reuses the blocking mechanism from T_{bg}.

Ambiguity propagation integration

Given two knowledge sources T_1 and T_2, the reasoning mechanism with ambiguity propagation can directly apply to the combination of the theories denoted as $T_{INT}^{'} = T_1 + T_2$. The preference between the two sources is unknown; therefore, there is no method to solve conflicts between them. The supportive and opposing arguments for any premise are removed from the final set of arguments. The set of arguments obtained by this integration is denoted by $Args_{\mathrm{AP}}^{T_{INT}^{'}}$.

6.4.3 Argument justification

The motivation of an agent to participate in the game is to promote its own goal. However, its claim can be refuted by different agents. To gain the acceptance of the group, at the first iteration, an agent should justify its arguments by the common constraints and expectations of the group governed by the background knowledge T_{bg}. The set of arguments justified by T_{bg} determines the arguments that an agent can play to defend its claim. In subsequent iterations and even if the proposal does not conflict with other agents, an agent should consider the knowledge from the others to determine the validity of its claim. That is, an agent is required a justification by collecting the individual arguments from the others.

Justification by background knowledge.

Agent A_{me} generates the set of arguments for its goals by combining its private knowledge T_{me} and the background knowledge T_{bg}. The combination is denoted as $T_{INT} = T_{bg} \ni T_{me}$ and the set of arguments is $Args^{T_{INT}}$. Owing to the non-monotonic nature of DL, the combination can produce arguments beyond individual knowledges. From A_{me}'s view, this can bring more opportunities to fulfil its goals. However, A_{me}'s arguments must be justified by the background knowledge T_{bg}, because T_{bg} governs the essential behaviours (expectations) of the group. Any attack on T_{bg} is not supported by the members of $\mathscr{A}$. A_{me} maintains the consistency with the background knowledge T_{bg} by the following procedure:

1. Create $T_{INT} = T_{bg} \ni T_{me}$. The new defeasible theory is obtained by replicating all the rules from the common constraints T_{bg} into the internal knowledge T_{me} while maintaining the superiority of the rules in T_{bg} over that in T_{me}.

2. Use the ambiguity blocking feature to construct the set of arguments $Args^{T_{bg}}$ from T_{bg} and the set of arguments $Args^{T_{INT}}_{\text{AB}}$ from T_{INT}.

3. Remove any argument in $Args^{T_{INT}}_{\text{AB}}$ attacked by those in $Args^{T_{bg}}$, obtain the justified arguments through the background knowledge $JArgs^{T_{INT}} = \{a \in Args^{T_{INT}}_{\text{AB}}$ and a is accepted by $Args^{T_{bg}}\}$.

Example 20. Consider two defeasible theories:

$$\begin{aligned} T_{bg} = \{R_d = \{\ & r_1 : e \Rightarrow c; \\ & r_2 : g, f \Rightarrow \sim c; \\ & r_3 : \Rightarrow e\}; \\ > = \{r_2 > r_1\}\} & \\ T_{me} = \{R_d = \{\ & r_1 : \Rightarrow d; \\ & r_2 : d \Rightarrow \sim a; \\ & r_3 : \Rightarrow g\}\} \end{aligned}$$

We have sets of arguments from the background theory and the integrated theory:

$$\begin{aligned} Args^{T_{bg}} &= \{\Rightarrow e; \Rightarrow e \Rightarrow c\} \\ Args^{T_{INT}} = Args^{T_{bg} \Rightarrow T_{me}} &= \{\Rightarrow e; \\ &\quad \Rightarrow e \Rightarrow c; \\ &\quad \Rightarrow d; \\ &\quad \Rightarrow g; \\ &\quad \Rightarrow d \Rightarrow \sim a\} \end{aligned}$$

In this example, there is no attack between the arguments in $Args^{T_{bg}}$ and $Args^{T_{INT}}_{\text{AB}}$. In other words, the arguments from $Args^{T_{INT}}$ are acceptable by those from $Args^{T_{bg}}$. The set of justified arguments w.r.t. $Args^{T_{bg}}$ is $JArgs^{T_{INT}} = Args^{T_{INT}}_{\text{AB}}$.

Collective justification

During the game, A_{me} can exploit the knowledge exposed by other agents to defend its main claims. Owing to possible conflicts in the individual proposals, an agent uses the sceptical semantics of the ambiguity propagation reasoning to retrieve the consistent knowledge. Essentially, given the competing arguments, an agent does not have any preference over them; therefore, these arguments will be rejected. The consistent knowledge from the others allows an agent to discover the 'collective wisdom' distributed among the agents to justify its claim.

The justification for the collective arguments, which are generated by integrating all knowledge sources, is achieved by the arguments from the background knowledge $Args^{T_{bg}}$. The procedure runs as follows:

1. Create a new defeasible theory $T'_{INT} = T_{bg} \ni T_{me} + \mathscr{T}_{other}$
2. Generate the set of arguments $Args_{\text{AP}}^{T'_{INT}}$ from T'_{INT} using the feature of ambiguity propagation
3. Justify the new set of arguments $JArgs^{T'_{INT}} = \{a | a \in Args_{\text{AP}}^{T'_{INT}} \text{ and } a \text{ is accepted by } Args^{T_{bg}}\}$.

$JArgs^{T'_{INT}}$ allows A_{me} to verify the status of its arguments for its claim $JArgs^{T_{INT}}$. If the arguments in $JArgs^{T'_{INT}}$ and $JArgs^{T_{INT}}$ do not attack one another, A_{me}'s claims are accepted by the other agents. Any conflict between two sets shows that accepting the arguments in $JArgs^{T'_{INT}}$ stops A_{me} from achieving its claims in the next steps. The set of arguments $Args^{opp}$ against A_{me} is identified as any argument in $JArgs^{T'_{INT}}$ attacking A_{me}'s arguments. A_{me} also establishes P^{opp} to select its counter-argument. It is noticed that A_{me} is self-consistent.

Example 21. Suppose the background knowledge T_{bg} and the private knowledge T_{me} of A_{me} are:

$$\begin{aligned} T_{bg} = \{R_d = \{\ & r_1 : e \Rightarrow c; \\ & r_2 : g, f \Rightarrow \sim c\}; \\ > = \{r_2 > r_1\}\} & \\ T_{me} = \{R_d = \{\ & r_1 : \Rightarrow e; \\ & r_2 : c \Rightarrow d; \\ & r_3 : \Rightarrow g\}\} \end{aligned}$$

Agent A_{me} currently plays $\{\Rightarrow e \Rightarrow c \Rightarrow d\}$ and knows about other agents:

$$\begin{aligned} \mathscr{T}_{other} &= \{T_1, T_2\} \text{ where } T_1 = \{\Rightarrow h \Rightarrow f \Rightarrow b \Rightarrow a\} \\ T_2 &= \{\Rightarrow e \Rightarrow c \Rightarrow a\} \end{aligned}$$

The claim of A_3 is acceptable w.r.t. arguments played by the other agents. However, the combination $T'_{INT} = T_{bg} \ni T_{me} + \mathscr{T}_{other}$ shows the difference. This combination generates $\{\Rightarrow$

$g;\ \Rightarrow e;\ \Rightarrow e \Rightarrow f \Rightarrow b;\ \Rightarrow g, f \Rightarrow \sim c\}$. The term $\{\Rightarrow g, f \Rightarrow \sim c\}$ is due to the superiority relation in T_{bg} which rebuts the claim of A_3. Therefore, the set of opposing arguments $Args^{opp} = \{\Rightarrow g, f \Rightarrow \sim c\}$ and $P^{opp} = \{f^1\}$. Given this information, A_3 should provide a counter-evidence to f to pursue c. Moreover, A_3 should not expose g to the other agents. Otherwise, A_3 has to drop its initial claim d.

6.5 Related Work

Substantial works have been carried out on the argumentation games in the artificial intelligence and law fields. Prakken and Sartor (1996) introduce a dialectical model of the legal argument, in the sense that the arguments can be attacked with appropriate counterarguments. In the model, the factual premises are not arguable, they are treated as strict rules. Bench-Capon (1998) presents an early specification and implementation of an argumentation game based on the Toulmin argument-schema without a specified underlying logic. Lodder (2000) presented The Pleadings Game as a normative formalisation and fully implemented computational model, using conditional entailments. The goal of the model was to identify the issues in the argumentation rather than, as in our case, elaborating the status of the main claim. Verheij (1996) provides a formal study on the role of rules and reasoning in argumentation, and the status of the arguments in the argumentation process.

Using the defeasible logic to capture the concepts of the argumentation game is supported by Hamfelt et al. (2005); Letia and Vartic (2006) and recently, Lundström et al. (2008); Thakur et al. (2007). Letia and Vartic (2006) focus on the persuasive dialogues for cooperative interactions among the agents. In the process, it includes the cognitive states of the agents, such as knowledge and beliefs, and presents some protocols for some types of dialogues (information seeking, explanation persuasion). Hamfelt et al. (2005) provide an extension of the defeasible logic to include the step of the adversarial dialogue by defining a meta-program for an alternative computational algorithm for ambiguity propagating defeasible logic. The logic presented here is ambiguity blocking.

We deal with the problem of the evolving knowledge of an agent during the iterations, where the argument construction is an extension of Lundström et al. (2008); Thakur et al. (2007). In

our work, we define the notion of the majority acceptance and a method to weight arguments. In (Thakur et al., 2007), the strength of the unchallenged rules is upgraded over the iterations. That is, the conclusions supported by these rules are not rebutted by the current iteration; these conclusions are unarguable in the following iterations. The upgrade is applied to all the participants during the iterations of the argumentation game. Lundström et al. (2008) distinguish the participants of the argumentation game. That is, one participant must provide a strong argument (a definite proof) to defeat the arguments from the other participants. Both the works do not directly handle the challenges coming from multiple participants.

We extend the protocol of a argumentation game to settle on a common goal. The termination condition of our framework is either there is no more argument to rebut or an agent can pass its proposal in one iteration.

Settling on a common goal among the agents can be seen as a negotiation process where agents exchange information to resolve the conflicts or to obtain missing information. Amgoud et al. (2007) provide a unified and general formal framework for the argumentation-based negotiation dialogue between two agents for a set of offers. The work provides a formal connection between the status of a argument including accepted, rejected and undecided with the possible actions of an agent (accept, reject and negotiate respectively). One important feature of the framework is that this representation is independent of the logical languages modelling the knowledge of an agent. Moreover, an agent's knowledge evolves by accumulating the arguments during the interactions.

We have advantages in using the defeasible logic, because it provides us with an elegant tool to capture the above statuses of the arguments naturally. Accepted, rejected, undecided conditions can be simulated by the proof conditions of defeasible reasoning w.r.t the ambiguity of the premises. If the preference of the knowledge sources is known, the accepted and rejected arguments are corresponding to $(+\partial, -\partial)$ using the feature of ambiguity blocking. Otherwise, the three conditions of the arguments are derived from ($+\partial, -\partial$ and $+\Sigma$). These notions correspond to the existence of a positive proof, a negative proof, and a positive support of a premise. In addition, defeasible logic provides a compact representation to accommodate the new information from the other agents.

From the perspective of coordination among agents, Parsons and McBurney (2003) present

an argumentation-based communication, where the agents can exchange arguments for their goals and plans to achieve those goals. The acceptance of the argument of an agent depends on the attitudes of this agent, namely credulous, cautious or sceptical. Also, Rueda et al. (2002) propose a communication mechanism based on the argumentation for collaborative BDI agents, in which the agents exchange their proposals and counter-proposals to reach a mutual agreement. During the course of conversations, an agent can retrieve missing literals (regarded as sub-goals) or fulfil its goals by requesting the collaboration of the other agents. However, these works did not clearly show how an agent can deal with the conflicts from multiple agents, especially when the preference over the exchanged arguments is unknown.

The main difference in our framework is the external model where more than two agents can argue to settle on a common goal. Because there is no preference over the proposal of the individual agents, the majority rule enables the group to identify the majority preference over the individual claims. On the one hand, we present the notion of the acceptance by the majority of the agents. On the other hand, this notion relaxes the complexity of the n-persons argumentation game by partitioning the agents into two sub-groups, one supports the major claim and the other opposes it. Moreover, the majority rule allows an agent to probe the attitudes of the group to dynamically create a preference over its defensive arguments if its main claim is not accepted by the majority of the agents. The strategy to defend against the topic of the dialogue is to attack the most common premise among the arguments supporting the topic.

In our framework, an individual agent efficiently tackles the conflicts from multiple sources of knowledge owing to the use of the defeasible logic as the underlying logic. The construction of the arguments requires an individual agent to integrate the background knowledge commonly shared among the agents, the knowledge from the other agents and its own private knowledge. The background knowledge has the priority over the other sources; therefore, when integrating any conflict, this knowledge is blocked. Because all agents are equally trustful, the knowledge from the other agents has the same weight. To achieve a consensus from the knowledge of the other agents and to discover the ‘collective wisdom’, the ambiguity propagation is applied over all knowledge sources of an individual agent.

6.6 Summary

This chapter has presented an n-person argumentation framework based on the defeasible logic. In the framework, we proposed an external model based on the argumentation/dialogue game that enables the agents in a group to settle on a common goal. An agent proposes its goal, including the explanation, and argues with the other agents about the goal. At the termination, the group identifies a common goal accepted by the majority of the group and the supportive argument for the goal.

We also propose an internal model of an agent where an individual agent can efficiently construct arguments from multiple sources of knowledge, including the background knowledge presenting the common constraints and expectations of the group, the knowledge from the others that is evolved during the iterations, and its own private knowledge. The background knowledge is preferred over the other sources of knowledge. Owing to the flexibility of defeasible logic in tackling the ambiguous information, these types of knowledge can be efficiently integrated with the private knowledge of an agent (with or without a preference over the knowledge sources) to generate and justify its arguments.

The majority rule relaxes the complexity of the n-persons argumentation dialogue game. This rule is used to identify the topic of the dialogue among the claims of agents. That is, the majority acceptance of an argument. Also, we propose a simple weighting mechanism, based on the frequency of the premises in the arguments attacking an agent's claim, to cope with the problem of the conflicts from multiple agents.

In future work, we will extend this mechanism to incorporate the notion of trustful arguments from trusted agents to better select a rebuttal argument and resolve the conflicts among agents.

7 Conclusions

In this section we summarise our work and provide some thoughts for further research issues.

7.1 Summary

From the literature of the multi-agents systems, there are two main challenges that the agent community is currently investigating. One is the development of formalisms for representing the knowledge the agents have about their actions, goals, plans for achieving their goals and other agents. The second challenge is the development of the reasoning mechanisms agents use to achieve autonomy during the course of their interactions. In the thesis, we have succeeded to construct a multi-agent framework with simple representation and efficient implementation in tackling the incomplete and conflicting information in agents' knowledge. The framework

allows not only us to internally model agents but also the dialogue between n agents using existing techniques namely defeasible logic and the social choice with minimal cost. The complexity of the reasoning mechanism equals to that of defeasible logic and the social choice.

The logical approach to multi-agent systems provides a declarative method for specifying what an agent knows and does not know about its working environment and the other agents. Furthermore, this approach can benefit from a 'free semantics' and a rich set of reasoning methods, such as deductive or abductive. A well-known and successful approach to modelling rational agents, the Beliefs, Desires and Intentions (BDI) architecture is inspired by human attitudes towards actions. Beliefs, Desires and Intentions are then the mental components in the architecture (Kinny et al., 1996; Rao and Georgeff, 1991). This architecture is strongly founded in the philosophical investigation by Bratman on human practical reasoning. The BDI model provides an insight on the decision-making process of an agent. Furthermore, the model facilitates building the agent systems, because of its clear definition of agent functionality.

The complexity of the logical model, including the BDI model, depends on the types of logics being used to model the agents' actions and the states of the environment. To obtain the tractability of the agents' rationality, there are trade-offs between the expressive capability and the computational complexity (Dantsin et al., 2001). Regarding this issue, the cost of the computing equilibrium among the agents' actions amounts to the space of the possible states. Essentially, the problem can be seen as a search to find an optimal path through all the possible combinations of the agents' states. That is a NP-hard problem.

Our research goals aim for a declarative method of representing the knowledge of the agents and the executable model for the agents' reasoning. Furthermore, we consider that an agent operates in a dynamic environment (an agent may be influenced by actions of other agents) and has only a partial image of the environment.

We have proposed a defeasible logic framework (DL-MAS) for multi-agent systems that explicitly captures the background knowledge of the group and the knowledge about the other agents. The joint knowledge from the other agents is a special source, because this knowledge is largely shared by the group. Considering the conflicts between the internal knowledge and this special knowledge, an individual agent can balance its mental attitudes with the 'desires' shared among the group.

To have a fine-grained model of the agent's behaviour, in particular, its social behaviour, we use the layer approach (MDL-MAS) to introduce modal notions into the knowledge structure of an agent. An agent can represent what it knows about the modal notions of the other agents, but not what it knows about what the others know. Being aware of the beliefs and goals of the other agents, an individual agent can discover the prevalent attitudes of the group and can balance those attitudes with its own. Therefore, agents are categorised into two types, majority and obedience. That is, an agent can either adapt to the majority behaviours or dominate the group, but is constrained to what are commonly recognised by the group. The agent in the second category can introduce a distinct behaviour that would lead the other agents, while committing to the obligations commonly recognised by the group.

We have investigated an n-person argumentation framework by using the technique from the DL-MAS model. In the framework, we studied a protocol for a group of agents settling on common goals. Also, we consider the efficiency of the underlying inference mechanism. We proposed an external model based on the argumentation/dialogue game that enables agents to communicate on a common goal. An agent proposes its goal including the explanation and argues with the other agents about the goal. At termination, the group identifies the common goals accepted by the majority of the group and the supportive argument for those goals.

We have investigated an n-person argumentation framework by using the technique from the DL-MAS model. In the framework, we study a protocol for a group of agents to settle on common goals. Also, we consider the efficiency of the underlying inference mechanism. We propose an external model based on the argumentation/dialogue game which enables agents to communicate on a common goal. An agent proposes its goal including the explanation and argues with other agents about the goal. At the termination, the group identifies common goals accepted by the majority of the group and the supportive argument for the goals.

We have also presented an internal model of an agent, where an individual agent can efficiently construct arguments from multiple sources of knowledge, including the background knowledge presenting the common constraints and expectations of the group, the knowledge from the others evolved during the iterations, and its own private knowledge. The background knowledge is preferred over the other sources of knowledge. Owing to the flexibility of the

defeasible logic in tackling the ambiguous information, these types of knowledge can be efficiently integrated with the private knowledge of an agent (with or without a preference over the knowledge sources) to generate and justify its arguments.

In addition to the efficiency of defeasible reasoning, the majority rule relaxes the complexity of the n-persons argumentation dialogue game. This rule is used to identify the topic of the dialogue among the claims of the agents. That is the majority acceptance of an argument. Also, the majority rule plays a key role in the weighting mechanism. The weight of an argument is based on the frequency of the premises in the arguments attacking an agent's claim, to cope with the problem of the conflicts from the multiple agents.

Another outcome of our research is the defeasible rule mark-up package. This package defines a standard for exchanging knowledge represented by defeasible logic. In addition to the standard defeasible reasoning, the package supports the modal notions as parameters of the reasoning process. In the next development, we aim for sophisticated interactions between the modal operators for capturing the more complex behaviour of the agents.

7.2 Discussion and Future work

Our multi-agent framework uses the majority rule to merge conclusions from individual knowledge bases of agents in the group. Interestingly, our reasoning mechanism does not always suffer the doctrinal paradox. The paradox is presented in the example below.

Example 22. Given a rule $r_1 : \ p \wedge q \rightarrow r$, and three belief bases A_1, A_2 and A_3 containing the truth values of p, q and the derivation of r_1 as follows:

$$\begin{array}{rccc} & p & q & r \\ A_1 = & \{true, & true, & true\} \\ A_2 = & \{true, & false, & false\} \\ A_3 = & \{false, & true, & false\} \end{array}$$

If these bases have equal weight (importance), the majority rule produces {true,true,false} for p, q and r respectively. The result contains a inconsistence with regards to r_1. If the majority supports p, q and knows r_1, r should be supported. That is the paradox.

The key to maintain the consistence of the merged belief is the rule r_1 known as integrity constraints or logical relations among premises (Pigozzi, 2006). With regards to our framework, r_1 is considered as background knowledge because all the agents recognise this rule in their belief bases. Consequently, the agents adapting to majority produce $\{p = true, q = true, r = true\}$. There is no paradox in the outcome. This surprising result is due to the extended defeasible reasoning where the majority knowledge is considered as inferior to the background knowledge. When combining the majority knowledge and the background knowledge, the agent retrieves the supports for $\neg r$ from the majority and r from r_1 in the background. The conclusion of r is retained thanks to the superiority of the background knowledge. Again, the key is the background knowledge that is largely shared among the agents. Clearly, without this information solving the paradox is not a trivial task. Neither voting on premises nor on the conclusions can solve the paradox (Pigozzi, 2006). In our framework, voting on the premises is worse and harder to obtain the majority conclusions because of the non-monotonicity of the logic representing the knowledge bases. Despite the high cost of computation, it is unlikely to reach the majority choice by changing the conflict-solving mechanism of every agent joining the pool. Voting on the conclusions and applying the extended defeasible reasoning produce an outcome consistent with the knowledge base largely shared by the group of agents. That can make sense because all the agents accept 'the logical relations' dictated by the background knowledge. In other words, the social choice is acceptable by the group if and only if the choice successfully passes the judgement of the commonly accepted 'logical relations' within the group. In fact, 'the logical relations' are not only the rules but also the conflict-solving mechanism of the knowledge base.

As can be seen from Example 22, if the agents in the group do not commonly accept r_1, they do not even know about the existence of the paradox. Therefore, in order to recognise and escape from the paradox it is critical for the group to identify the commonly accepted 'logical relations'. Chapter 6 presents a possible extension of the DL-MAS reasoning mechanism, where n agents in the group can argue to largely accept a conclusion and its explanation. In the extension, the majority rule is used to weight the popularity of premises constructing the explanation. We believe that this extension can better tackle with the beliefs which are 'deeply embedded' (Pettit, 2006) in agents' knowledge bases. In other words, our extension provides a method to construct the majority knowledge on the 'logical relations', which are embedded in

the knowledge bases of individual agents of the group. We have sketched out the intuition on tackling the doctrinal paradox. In the future work, we would investigate a formal representation and solution to this problem.

Our DL-MAS framework favours the internal view approach to agents' behaviour. However, we believe that our framework can be used as a tool for external modelling in the case that all agents expose their knowledge. This shows the flexibility of our framework and it is worth for a further investigation.

In our DL-MAS framework, we have developed a light-weight reasoning package based on defeasible logic. We plan to incorporate modal defeasible logic instead of the layer approach to capture the notions of belief, intention, and obligation. The interaction among belief, intention, and obligation is described as parameters of the reasoning process. In one hand, that improves the flexibility in describing behaviour of an agent and the interaction among agents. On the other hand, that increases the inter-operation among reasoning mechanisms. If a reasoning mechanism does not support the transformations between modal notions, those operations can be ignored. Consequently, the agents can illustrate the basic behaviour.

Our framework assumes an existing mechanism that allows an agent to obtain information from other agents. Boella et al. (2007a) propose a mechanism for exchanging rule-based information using FIPA communicative acts. The semantics of the communicative acts depend not depend only on the private knowledge of an agent, but also on the mental attitudes publicly known by the agents. We intend to extend this mechanism in such a way that the agents can recognise the mental attitudes largely shared by the agents. These attitudes should be considered as an important source for the assertion of upcoming information.

In our model for multi-agent systems, the information about the weight (reputation) of an agent is initiated by the designers. It is more useful to update this during the interactions among the agents. The idea is that the behaviour of our agents are driven by three types of knowledge, the internal knowledge, the knowledge shared by the group and the knowledge from other agents. Knowledge commonly shared or largely recognised by individuals enables the agents to 'discover' the common values in the group. Hence, the agents can justify the behaviour of the others. Behaviour for/against the values of the group can increase/decrease the reputation of the owners. Because of the knowledge from the others, an agent can reason about the intended

actions of the other agents. The agent can balance between the actual and the intended actions to update the reputation of the other agents. That is, by tracking the commitment of the individuals to the group, we can build up a social reputation model for our agents. The approach provides more quantitative evidence than the interaction rating model proposed by Sabater and Sierra (2001) or on-line auction systems such as eBay.

We are investigating a computer-based tool to simulate emergency situations where rescue teams are well equipped with comprehensive emergency procedures, but the information is incomplete and conflicting. The simulation tool facilitates studying the behaviour of the individual members and the whole team, and the effectiveness of the rescue procedures.

References

JACK Intelligent Agents User Guide. Agent Oriented Software (AOS), Carlton, Victoria, version 3 edition, 2001.

Emerson E. Allen and Srinivasan Jai. Branching time temporal logic. In J. W. de Bakker, Willem P. de Roever, and Grzegorz Rozenberg, editors, *REX Workshop*, volume 354 of *Lecture Notes in Computer Science*, pages 123–172. Springer, 1988. ISBN 3-540-51080-X.

Eduardo Alonso. Rights and argumentation in open multi-agent systems. *Artificial Intelligent Review*, 21(1):3–24, 2004. ISSN 0269-2821.

Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 100–109, Washington, DC,USA, 1997. IEEE Computer Society.

Rajeev Alur, Thomas A. Henzinger, and Orna Kupferman. Alternating-time temporal logic. *J. ACM*, 49(5):672–713, 2002. ISSN 0004-5411.

Leila Amgoud and Souhila Kaci. An argumentation framework for merging conflicting knowledge bases. *Int. J. Approx. Reasoning*, 45(2):321–340, 2007. ISSN 0888-613X.

Leila Amgoud, Yannis Dimopoulos, and Pavlos Moraitis. A unified and general framework for argumentation-based negotiation. In *Proceedings of the 6th international joint conference on AAMAS*, pages 1–8, 2007. ISBN 978-81-904262-7-5.

Grigoris Antoniou. *Nonmonotonic reasoning*. MIT Press, Cambridge, Mass., 1997. ISBN 0262011573 0262011573 0262011573.

Grigoris Antoniou. Defeasible reasoning: A discussion of some intuitions. *International Journal of Intelligent System*, 21(6):545–558, 2006. ISSN 0884-8173.

Grigoris Antoniou and Antonis Bikakis. DR-Prolog: A system for defeasible reasoning with rules and ontologies on the semantic web. *IEEE Trans. on Knowl. and Data Eng.*, 19(2): 233–245, 2007. ISSN 1041-4347.

Grigoris Antoniou, David Billington, and Michael J. Maher. On the analysis of regulations using defeasible rules. In *HICSS '99: Proceedings of the Thirty-second Annual Hawaii International Conference on System Sciences-Volume 6*, page 6033, Washington, DC, USA, 1999a. IEEE Computer Society. ISBN 0-7695-0001-3.

Grigoris Antoniou, Fumihiro Maruyama, Ryusuke Masuoka, and Hironobu Kitajima. Issues in intelligent information integration. In Borko Furht, editor, *Internet, Multimedia Systems and Applications*, pages 345–349, Nassau, The Bahamas, 1999b. IASTED/ACTA Press. ISBN 0-88986-269-9.

Grigoris Antoniou, David Billington, Guido Governatori, and Michael J. Maher. A flexible framework for defeasible logics. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 405–410. AAAI Press / The MIT Press, 2000a. ISBN 0-262-51112-6.

Grigoris Antoniou, Michael J. Maher, and David Billington. Defeasible logic versus logic programming without negation as failure. *Journal of Logic Programming*, 42(1):47–57, 2000b.

Grigoris Antoniou, David Billington, Guido Governatori, and Michael J. Maher. Representation results for defeasible logic. *ACM Transactions on Computational Logic*, 2(2):255–287, 2001. ISSN 1529-3785.

Grigoris Antoniou, David Billington, Guido Governatori, and Michael J. Maher. Embedding defeasible logic into logic programming. *Theory and Practice of Logic Programming*, 6(6): 703–735, 2006. doi: 10.1017/S1471068406002778.

Guillaume Aucher. A combined system for update logic and belief revision. In Mike Barley and Nikola K. Kasabov, editors, *PRIMA*, volume 3371 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2004. ISBN 3-540-25340-8.

Robert J. Aumann. Agreeing to disagree. *The Annals of Statistics*, 4(6):1236–1239, 1976. ISSN 00905364.

Alexandru Baltag, Lawrence S. Moss, and Slawomir Solecki. The logic of public announcements, common knowledge, and private suspicions. In *TARK '98: Proceedings of the 7th conference on Theoretical aspects of rationality and knowledge*, pages 43–56, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1-55860-563-0.

Chitta Baral. *Knowledge Representation, Reasoning and Declarative Problem Solving*. Cambridge Press, 2003.

Chitta Baral and Michael Gelfond. Reasoning about intended actions. In Manuela M. Veloso and Subbarao Kambhampati, editors, *AAAI*, pages 689–694. AAAI Press / The MIT Press, 2005. ISBN 1-57735-236-X.

Nick Bassiliades and Ioannis P. Vlahavas. R-DEVICE: A deductive RDF rule language. In Grigoris Antoniou and Harold Boley, editors, *RuleML*, volume 3323 of *Lecture Notes in Computer Science*, pages 65–80. Springer, 2004. ISBN 3-540-23842-5.

Nick Bassiliades, Grigoris Antoniou, and Ioannis P. Vlahavas. DR-DEVICE: A defeasible logic system for the semantic web. In Hans Jürgen Ohlbach and Sebastian Schaffert, editors, *Principles and Practice of Semantic Web Reasoning, Second International Workshop, PPSWR 2004, St. Malo, France, September*, volume 3208 of *Lecture Notes in Computer Science*, pages 134–148. Springer, 2004. ISBN 3-540-22961-2.

Nick Bassiliades, Grigoris Antoniou, and Ioannis P. Vlahavas. A defeasible logic reasoner for the semantic web. *International Journal on Semantic Web and Information Systems*, 2(1): 1–41, 2006.

Trevor J.M. Bench-Capon. Specification and implementation of Toulmin dialogue game. In J.C. Hage, T.J.M. Bench-Capon, A.W. Koers, C.N.J. de Vey Mestdagh, and C.A.F.M. Grutters, editors, *Legal Knowledge Based Systems: JURIX: The Eleventh Conference*, pages 5–20. Nijmegen: Gerard Noodt Instituut, 1998. ISBN 90-71478-58-0.

David Billington. Defeasible logic is stable. *Journal of Logic and Computation*, 3(4):379–400, 1993.

G. Boella, J. Hulstijn, G. Governatori, R. Riveret, A. Rotolo, and L. van der Torre. FIPA communicative acts in defeasible logic. In *Procs. of the 7th International Workshop on Nonmontonic Reasoning, Action and Change*, 2007a.

Guido Boella and Leendert W. N. van der Torre. A game theoretic approach to contracts in multiagent systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 36(1): 68–79, 2006.

Guido Boella and Leendert W. N. van der Torre. A game-theoretic approach to normative multi-agent systems. In Boella et al. (2007b).

Guido Boella, Leendert W. N. van der Torre, and Harko Verhagen, editors. *Normative Multi-agent Systems, 18.03. - 23.03.2007*, volume 07122 of *Dagstuhl Seminar Proceedings*, 2007b. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany.

Alan H. Bond and Les Gasser. An analysis of problems and research in DAI. In A. H. Bond and L. Gasser, editors, *Proc. Readings in Distributed Artificial Intelligence*, pages 3–36, San Mateo, CA, 1988. Morgan Kaufmann.

Rafael Bordini and Jomi Hübner. BDI agent programming in agentspeak using jason (tutorial paper). In *Computational Logic in Multi-Agent Systems*, pages 143–164. Springer-Verlag, 2006.

Rafael Bordini, Jomi Hübner, and Renata Vieira. Jason and the golden fleece of agent-oriented programming. In *Multi-Agent Programming*, pages 3–37. Springer-Verlag, 2005.

Rafael Bordini, Lars Braubach, Mehdi Dastani, Amal El Fallah Seghrouchni, Jorge Gomez-Sanz, Joao Leite, Gregory O'Hare, Alexander Pokahr, and Alessandro Ricci. A survey of programming languages and platforms for multi-agent systems. *Informatica*, 30(1):33–44, 2006.

Luc Bovens and Wlodek Rabinowicz. Democratic answers to complex questions – an epistemic perspective. *Synthese*, 150(1):131–153, 2006.

Michael E. Bratman. *Intention, Plans, and Practical Reason*. Harvard University Press, Cambridge (MA), 1987.

Gerhard Brewka. On the relationship between defeasible logic and well-founded semantics. In *LPNMR '01: Proceedings of the 6th International Conference on Logic Programming and Nonmonotonic Reasoning*, pages 121–132, London, UK, 2001. Springer-Verlag. ISBN 3-540-42593-4.

Jan Broersen, Mehdi Dastani, Joris Hulstijn, Zhisheng Huang, and Leendert W. N. van der Torre. The BOID architecture: conflicts between beliefs, obligations, intentions and desires. In *Agents*, pages 9–16, 2001.

Antonio Brogi. On the semantics of logic program composition. In *Program Development in Computational Logic*, volume 3049 of *Lecture Notes in Computer Science*, pages 115–151. Springer, 2004. ISBN 3-540-22152-2.

Birgit Burmeister, Afsaneh Haddadi, and Guido Matylis. Application of multi-agent systems in traffic and transportation. *IEE Proceedings - Software*, 144(1):51–60, 1997.

Cristiano Castelfranchi. Commitments: From individual intentions to groups and organizations. In Lesser and Gasser (1995), pages 41–48. ISBN 0-262-62102-9.

Cristiano Castelfranchi. Limits of strategic rationality for agents and M-A systems. In *Proceedings of the 8th European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, page 3, London, UK, 1997. Springer-Verlag. ISBN 3-540-63077-5.

Cristiano Castelfranchi. Modelling social action for ai agents. *Artificial Intelligence*, 103(1-2): 157–182, 1998. ISSN 0004-3702. doi: http://dx.doi.org/10.1016/S0004-3702(98)00056-3.

Cristiano Castelfranchi, Frank Dignum, Catholijn M. Jonker, and Jan Treur. Deliberative normative agents: Principles and architecture. In *ATAL '99: 6th International Workshop on Intelligent Agents VI, Agent Theories, Architectures, and Languages (ATAL),*, pages 364–378, London, UK, 2000. Springer-Verlag. ISBN 3-540-67200-1.

Lawrence Cavedon and Liz Sonenberg. On social commitment, roles and preferred goals. In Yves Demazeau, editor, *ICMAS*, pages 80–87. IEEE Computer Society, 1998. ISBN 0-8186-8500-X.

Laura A. Cecchi, Pablo R. Fillottrani, and Guillermo R. Simari. On the complexity of DeLP through game semantics. In Juergen Dix and Anthony Hunter, editors, *Proc. 11th Intl. Workshop on Nonmonotonic Reasoning (NMR 2006), Windermere, UK,*, IfI Technical Report Series,Clausthal University, pages 386–394. IfI Technical Report Series,Clausthal University, 2006.

Bruce Chapman. Rational aggregation. *Politics, Philosophy and Economics*, 1(3), 2002.

Carlos Iván Chesñevar, Jürgen Dix, Frieder Stolzenburg, and Guillermo Ricardo Simari. Relating defeasible and normal logic programming through transformation properties. *Theoretical Computer Science*, 290(1):499–529, 2003.

Roderick M. Chisholm. Contrary-to-duty imperatives and deontic logic. *Analysis*, 24:33–36, 1963.

Laurence Cholvy. A logical approach to multi-sources reasoning. In Michael Masuch and László Pólos, editors, *Knowledge Representation and Reasoning Under Uncertainty*, volume 808 of *Lecture Notes in Computer Science*, pages 183–196. Springer, 1994. ISBN 3-540-58095-6.

CLIPS. *CLIPS Reference Manual*. NASA, January 1992.

Philip R. Cohen and Hector J. Levesque. Intention is choice with commitment. *Artif. Intell.*, 42(2-3):213–261, 1990. ISSN 0004-3702.

Rosaria Conte and Cristiano Castelfranchi. *Cognitive and social action*. UCL Press London, 1995.

Michael A. Covington, Donald Nute, and André Vellino. *Prolog programming in depth*. Scott, Foresman & Co., Glenview, IL, USA, 1987. ISBN 0-673-18659-8.

Evgeny Dantsin, Thomas Eiter, Georg Gottlob, and Andrei Voronkov. Complexity and expressive power of logic programming. *ACM Comput. Surv.*, 33(3):374–425, 2001. ISSN 0360-0300.

Mehdi Dastani, Frank de Boer, Frank Dignum, and John-Jules Meyer. Programming agent deliberation: an approach illustrated using the 3APL language. In *AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 97–104, New York, NY, USA, 2003a. ACM. ISBN 1-58113-683-8.

Mehdi Dastani, Jeroen van der Ham, and Frank Dignum. Communication for goal directed agents. In *Communications in Multiagent Systems*, pages 239–252. 2003b.

Mehdi Dastani, Guido Governatori, Antonino Rotolo, and Leendert W. N. van der Torre. Preferences of agents in defeasible logic. In Shichao Zhang and Ray Jarvis, editors, *Australian Conference on Artificial Intelligence*, volume 3809 of *Lecture Notes in Computer Science*, pages 695–704. Springer, 2005a. ISBN 3-540-30462-2.

Mehdi Dastani, Guido Governatori, Antonino Rotolo, and Leendert W. N. van der Torre. Programming cognitive agents in defeasible logic. In Geoff Sutcliffe and Andrei Voronkov, editors, *LPAR*, volume 3835 of *Lecture Notes in Computer Science*, pages 621–636. Springer, 2005b. ISBN 3-540-30553-X.

Mehdi Dastani, M. Birna Riemsdijk, and John-Jules Meyer. Programming multi-agent systems in 3APL. In *Multi-Agent Programming*, pages 39–67. Springer-Verlag, 2005c.

Mehdi Dastani, Guido Governatori, Antonio Rotolo, Insu Song, and Leon van der Torre. Contextual deliberation of cognitive agents in defeasible logic. In *AAMAS '07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1–3, New York, NY, USA, 2007. ACM. ISBN 978-81-904262-7-5.

Keith Decker, Anandeep Pannu, Katia Sycara, and Mike Williamson. Designing behaviors for information agents. In *AGENTS '97: Proceedings of the first international conference on Autonomous agents*, pages 404–412, New York, NY, USA, 1997. ACM. ISBN 0-89791-877-0.

James P. Delgrande and Torsten Schaub, editors. *10th International Workshop on Non-Monotonic Reasoning (NMR 2004), Whistler, Canada, June 6–8, 2004, Proceedings*, 2004. ISBN 92-990021-0-X.

Frank Dignum. Autonomous agents with norms. *Artificial Intelligent Law*, 7(1):69–79, 1999.

Frank Dignum, David Kinny, and Liz Sonenberg. Motivational attitudes of agents: On desires, obligations, and norms. In *CEEMAS '01: Revised Papers from the Second International Workshop of Central and Eastern Europe on Multi-Agent Systems*, pages 83–92, London, UK, 2002. Springer-Verlag. ISBN 3-540-43370-8.

Yannis Dimopoulos and Antonis C. Kakas. Logic programming without negation as failure. In John W. Lloyd, editor, *International Logic Programming Symposium*, pages 369–383, Portland, Oregon, 1995. MIT Press. ISBN 0-262-62099-5.

Mark D'Inverno, David Kinny, Michael Luck, and Michael Wooldridge. A formal specification of dMARS. In *ATAL '97: Proceedings of the 4th International Workshop on Intelligent Agents IV, Agent Theories, Architectures, and Languages*, pages 155–176, London, UK, 1998. Springer-Verlag. ISBN 3-540-64162-9.

Mark D'Inverno, Michael Luck, Michael Georgeff, David Kinny, and Michael Wooldridge. The dMARS architecture: A specification of the distributed multi-agent reasoning system. *Autonomous Agents and Multi-Agent Systems*, 9(1-2):5–53, 2004. ISSN 1387-2532.

Jürgen Dix, Frieder Stolzenburg, Guillermo Ricardo Simari, and Pablo R. Fillottrani. Automating defeasible reasoning with logic programming. In *German-Argentinian Workshop on Information Technology*, pages 39–46, 1999.

Kurt Dresner and Peter Stone. Multiagent traffic management: A reservation-based intersection control mechanism. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 530–537, Washington, DC, USA, 2004. IEEE Computer Society. ISBN 1-58113-864-4.

Phan Minh Dung. An argumentation semantics for logic programming with explicit negation. In *ICLP'93: Proceedings of the tenth international conference on logic programming on Logic programming*, pages 616–630, Cambridge, MA, USA, 1993. MIT Press. ISBN 0-262-73105-3.

Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, 1995.

Edmund H. Durfee. Planning in distributed artificial intelligence. pages 231–245, 1996.

Edmund H. Durfee. Practically coordinating. *AI Magazine*, 20(1):99–116, 1999.

Edmund H. Durfee. *An Application Science for Multi-Agent Systems*, chapter Challenges to scaling-up agent coordination strategies, pages 113–132. Kluwer Academic, Dordrecht, 2004.

Thomas Eiter, Nicola Leone, Cristinel Mateis, Gerald Pfeifer, and Francesco Scarcello. The KR system dlv: Progress report, comparisons and benchmarks. In *KR*, pages 406–417, 1998.

Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning about knowledge*. The MIT Press,, 2003. ISBN 0-262-56200-6.

Marcelo A. Falappa, Alejandro Javier García, and Guillermo Ricardo Simari. Belief dynamics and defeasible argumentation in rational agents. In Delgrande and Schaub (2004), pages 164–170. ISBN 92-990021-0-X.

Jacques Ferber. *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*, chapter Interactions and Cooperation, pages 59–85. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

Edgardo Ferretti, Marcelo Errecalde, Alejandro Javier García, and Guillermo Ricardo Simari. An application of defeasible logic programming to decision making in a robotic environment. In Chitta Baral, Gerhard Brewka, and John S. Schlipf, editors, *LPNMR*, volume 4483 of *Lecture Notes in Computer Science*, pages 297–302. Springer, 2007. ISBN 978-3-540-72199-4.

Klaus Fischer. Cooperative transportation scheduling: An application domain for DAI. *Applied Artificial Intelligence*, 10(1):1–34, 1996. ISSN 0883-9514.

Michael Fisher, H. Bordini Rafael, Benjamin Hirsch, and Paolo Torroni. Computational logics and agents: A road map of current technologies and future trends. *Computational Intelligence*, 23(1):61–91, February 2007. ISSN 0824-7935.

André Fuhrmann. *An Essay on Contraction*. CSLI Publications, Stanford University, 1997.

Alejandro Javier García and Guillermo R. Simari. Defeasible logic programming: an argumentative approach. *Theory and Practice of Logic Programming*, 4(2):95–138, 2004. ISSN 1471-0684.

Alejandro Javier García, Nicolás D. Rotstein, Mariano Tucat, and Guillermo Ricardo Simari. An argumentative reasoning service for deliberative agents. In Zili Zhang and Jörg H. Siekmann, editors, *KSEM*, volume 4798 of *Lecture Notes in Computer Science*, pages 128–139. Springer, 2007. ISBN 978-3-540-76718-3.

Michael Gelfond. Logic programming and reasoning with incomplete information. *Annals of Mathematics and Artificial Intelligence*, 12(1-2):89–116, 1994.

Michael Gelfond and Vladimir Lifschitz. Representing actions in extended logic programming. In *Proceedings of the Joint International Conference and Symposium on Logic Programming*, pages 559–573. MIT Press, 1992. ISBN 0-262-51064-2.

Michael R. Genesereth, Matthew L. Ginsberg, and Jeffrey S. Rosenschein. Cooperation without communication. In *Proceedings of the 5th National Conference on Artificial Intelligence*, pages 51–57, Philadelphia, PA, 1986. Morgan Kaufmann.

Michael P. Georgeff and Amy L. Lansky. Reactive reasoning and planning. In *Proceedings of the 6th National Conference on Artificial Intelligence (AAAI)*, pages 677–682, Menlo Park, CA, USA, 1987.

Jelle Gerbrandy and Willem Groeneveld. Reasoning about information change. *Journal of Logic, Language and Information*, 6(2):147–169, 1997. ISSN 0925-8531.

Guido Governatori. Representing business contracts in RuleML. *International Journal of Cooperative Information Systems*, 14(2-3):181–216, 2005.

Guido Governatori and Zoran Milosevic. A formal analysis of a business contract language. *International Journal of Cooperative Information Systems*, 15(4):659–685, 2006. doi: 10.1142/S0218843006001529.

Guido Governatori and Duy Hoang Pham. DR-CONTRACT: An architecture for e-contracts in defeasible logic. In *2nd EDOC Workshop on Contract Architectures and Languages (CoALA 2005)*. IEEE Digital Library, 2005a. Published on CD.

Guido Governatori and Duy Hoang Pham. A semantic web based architecture for e-contracts in defeasible logic. In *Rules and Rule Markup Languages for the Semantic Web, First International Conference, RuleML 2005, Galway, Ireland, November 10-12, 2005, Proceedings*, volume 3791 of *Lecture Notes in Computer Science*. Springer, 2005b. ISBN 3-540-29922-X.

Guido Governatori and Antonino Rotolo. Defeasible logic: Agency, intention and obligation. In Alessio Lomuscio and Donald Nute, editors, *Deontic Logic in Computer Science*, volume 3065 of *Lecture Notes in Computer Science*, pages 114–128. Springer, 2004. ISBN 3-540-22111-5.

Guido Governatori and Antonino Rotolo. BIO logical agents: Norms, beliefs, intentions in defeasible logic. *Autonomous Agents and Multi-Agent Systems*, 17(1):36–69, 2008. ISSN 1387-2532.

Guido Governatori, Michael J. Maher, Grigoris Antoniou, and David Billington. Argumentation semantics for defeasible logic. *Journal of Logic and Computation*, 14(5):675–702, 2004. ISSN 0955-792X.

Guido Governatori, Zoran Milosevic, and Shazia Sadiq. Compliance checking between business processes and business contracts. In Patrick C. K. Hung, editor, *10th International Enterprise Distributed Object Computing Conference (EDOC 2006)*, pages 221–232. IEEE Computing Society, 16–20 October 2006a. ISBN 978-0-7695-2558-7. doi: 10.1109/EDOC.2006.22.

Guido Governatori, Antonino Rotolo, and Vineet Padmanabhan. The cost of social agents. In Peter Stone and Gerhard Weiss, editors, *5th International Conference on Autonomous Agents and Multi-Agent Systems*, pages 513–520, New York, 2006b. ACM Press. ISBN 1-59593-303-4.

Guido Governatori, Duy Hoang Pham, Simon Raboczi, Andrew Newman, and Subhasis Thakur. On extending ruleml for modal defeasible logic. In *Rule Representation, Interchange and Reasoning on the Web, International Symposium, RuleML 2008, Orlando, FL, USA, October 30-31, 2008. Proceedings*, volume 5321 of *Lecture Notes in Computer Science*, pages 89–103. Springer, 2008. ISBN 978-3-540-88807-9.

Eric Grégoire and Sébastien Konieczny. Logic-based approaches to information fusion. *Information Fusion*, 7(1):4–18, 2006. ISSN 1566-2535.

Benjamin N. Grosof. Prioritized conflict handling for logic programs. In Jan Maluszynski, editor, *International Logic Programming Symposium*, pages 197–211, Port Jefferson, Long Island, N.Y., 1997. MIT Press. ISBN 0-262-63180-6.

Benjamin N. Grosof, Yannis Labrou, and Hoi Y. Chan. A declarative approach to business rules in contracts: courteous logic programs in xml. In *EC '99: Proceedings of the 1st ACM conference on Electronic commerce*, pages 68–77, New York, NY, USA, 1999. ACM. ISBN 1-58113-176-3.

Joseph Y. Halpern. A computer scientist looks at game theory. *Games and Economic Behavior*, 45(1):114–131, 2003.

Joseph Y. Halpern. Beyond nash equilibrium: solution concepts for the 21st century. In *PODC '08: Proceedings of the twenty-seventh ACM symposium on Principles of distributed computing*, pages 1–10, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-989-0.

Joseph Y. Halpern and Ronald Fagin. Modelling knowledge and action in distributed systems. *Distributed Computing*, 3(4):159–177, 1989.

Andreas Hamfelt, Jenny Eriksson, and Jorgen Fischer Nilsson. A metalogic formalization of legal argumentation as game trees with defeasible reasoning. In *ICAIL '05: Proceedings of the 10th international conference on Artificial intelligence and law*, pages 250–251, New York, NY, USA, 2005. ACM. ISBN 1-59593-081-7.

Jörg Hansen, Gabriella Pigozzi, and Leendert W. N. van der Torre. Ten philosophical problems in deontic logic. In Boella et al. (2007b).

Sven Ove Hansson. Preference-based deontic logic (PDL). *Journal of Philosophical Logic*, 19:75–93, 1990.

Risto Hilpinen. *Deontic logic: introductory and systematic readings*. Dordrecht: Reidel, 1971.

Koen V. Hindriks, Frank S. de Boer, Wiebe van der Hoek, and John-Jules Ch. Meyer. Formal semantics for an abstract agent programming language. In *ATAL '97: Proceedings of the 4th International Workshop on Intelligent Agents IV, Agent Theories, Architectures, and Languages*, pages 215–229, London, UK, 1998. Springer-Verlag. ISBN 3-540-64162-9.

Koen V. Hindriks, Frank S. De Boer, Wiebe Van Der Hoek, and John-Jules Ch. Meyer. Agent programming in 3APL. *Autonomous Agents and Multi-Agent Systems*, 2(4):357–401, 1999. ISSN 1387-2532.

Koen V. Hindriks, Frank S. de Boer, Wiebe van der Hoek, and John-Jules Ch. Meyer. Agent programming with declarative goals. In *ATAL '00: Proceedings of the 7th International Workshop on Intelligent Agents VII. Agent Theories Architectures and Languages*, pages 228–243, London, UK, 2001. Springer-Verlag. ISBN 3-540-42422-9.

Jaakko Hintikka. *Knowledge and belief : an introduction to the logic of the two notions*. Ithaca, N.Y. : Cornell University Press, 1962.

John F. Horty. Argument construction and reinstatement in logics for defeasible reasoning. *Artificial Intelligence and Law*, 9(1):1–28, 2001.

John F. Horty. Skepticism and floating conclusions. *Artif. Intell.*, 135(1-2):55–72, 2002. ISSN 0004-3702. doi: http://dx.doi.org/10.1016/S0004-3702(01)00160-6.

Michael N. Huhns and Munindar P. Singh. Managing heterogeneous transaction workflows with co-operating agents. pages 219–239, 1998.

Michael N. Huhns and Larry M. Stephens. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, chapter Multiagent Systems and Societies of Agents, pages 79–121. MIT Press, 1999.

Nicholas R. Jennings. Commitments and conventions: the foundation of coordination in multiagent systems. *The Knowledge Engineering Review*, 8(3):223–250, 1993.

Nicholas R. Jennings and E. H. Mamdani. Using joint responsibility to coordinate collaborative problem solving in dynamic environments. In *AAAI*, pages 269–275, 1992.

Nicholas R. Jennings, Simon Parsons, Pablo Noriega, and Carles Sierra. On argumentation-based negotiation. MIT, MIT, 1998.

Nick R. Jennings. Coordination techniques for distributed artificial intelligence. pages 187–210, 1996.

David Kinny and Michael P. Georgeff. Commitment and effectiveness of situated agents. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, volume 1, pages 82–88, Sydney, Australia, 1991. Morgan Kaufmann.

David Kinny, Michael Georgeff, and Anand Rao. A methodology and modelling technique for systems of BDI agents. In *MAAMAW '96*, pages 56–71. Springer-Verlag New York, Inc., 1996. ISBN 3-540-60852-4.

Efstratios Kontopoulos, Nick Bassiliades, and Grigoris Antoniou. Deploying defeasible logic rule bases for the semantic web. *Data & Knowledge Engineering*, 66(1):116–146, 2008.

Lewis A. Kornhauser and Lawrence G. Sager. Unpacking the court. *The Yale Law Journal*, 96 (1):82–117, Nov 1986.

Robert Kowalski. Logic programming and the real world. *Logic Programming Newsletter*, 14 (1), 2001.

Saul Kripke. A completeness theorem in modal logic. *Journal of Symbolic Logic*, 24(1):1–14, 1959.

Saul Kripke. Semantical considerations on modal logic. *Acta Philosophica Fennica*, 16:83–94, 1963.

Kenneth Kunen. Negation in logic programming. *Journal of Logic Programming*, 4(4):289–308, 1987. ISSN 0743-1066.

Yannis Labrou. *Semantics for an Agent Communication Language*. PhD thesis, University of Maryland, USA, 1997.

Nicholas Lacey and Henry Hexmoor. Norm adaptation and revision in a multi-agent system. In Ingrid Russell and Susan M. Haller, editors, *FLAIRS Conference*, pages 27–31. AAAI Press, 2003. ISBN 1-57735-177-0.

Adam J. Lee, Jodie P. Boyer, Lars E. Olson, and Carl A. Gunter. Defeasible security policy composition for web services. In *FMSE '06: Proceedings of the fourth ACM workshop on Formal methods in security*, pages 45–54, New York, NY, USA, 2006. ACM. ISBN 1-59593-550-9.

Victor R. Lesser and Les Gasser, editors. *Proceedings of the First International Conference on Multiagent Systems, June 12–14, 1995, San Francisco, California, USA*, 1995. The MIT Press. ISBN 0-262-62102-9.

Ioan Alfred Letia and Raluca Vartic. Defeasible protocols in persuasion dialogues. In *WI-IATW '06: Proceedings of the 2006 IEEE/WIC/ACM International conference on Web Intelligence and Intelligent Agent Technology*, pages 359–362, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2749-3.

Clarence Lewis. *A Survey of Symbolic Logic*. University of California Press, 1918. Republished by Dover, 1960.

David K. Lewis. *Convention: a philosophical study.* Harvard University Press, Cambridge, 1969.

Churn-Jung Liau. A modal logic framework for multi-agent belief fusion. *ACM Transactions on Computational Logic*, 6(1):124–174, 2005. ISSN 1529-3785.

Jinxin Lin. Integration of weighted knowledge bases. *Artificial Intelligence*, 83(2):363–378, 1996. ISSN 0004-3702.

Magnus Ljungberg and Andrew Lucas. The oasis air-traffic management system. In *Proceedings of the Second Pacific Rim International Conference on Artificial Intelligence PRICAI '92*, pages 236–243, Seoul, Korean, September 1992. ISBN 89-85368-00-093560.

Arno R. Lodder. Thomas F. Gordon, The Pleadings Game – an artificial intelligence model of procedural justice. *Artifficial Intelligence and Law*, 8(2/3):255–264, 2000.

Jenny Eriksson Lundström, Guido Governatori, Subhasis Thakur, and Vineet Padmanabhan. An asymmetric protocol for argumentation games in defeasible logic. In *10 Pacific Rim International Workshop on Multi-Agents*, volume 5044 of *LNCS*. Springer, 2008.

Michael J. Maher. Propositional defeasible logic has linear complexity. *Theory and Practice of Logic Programming*, 1(6):691–711, 2001. ISSN 1471-0684.

Michael J. Maher and Guido Governatori. A semantic decomposition of defeasible logics. In *AAAI '99/IAAI '99: Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence*, pages 299–305, Menlo Park, CA, USA, 1999. American Association for Artificial Intelligence. ISBN 0-262-51106-1.

Michael J. Maher, Andrew Rock, Grigoris Antoniou, David Billignton, and Tristan Miller. Efficient defeasible reasoning systems. *International Journal of Artificial Intelligence Tools*, 10(4):483–501, 2001.

David Makinson and Karl Schlechta. Floating conclusions and zombie paths: two deep difficulties in the "directly skeptical" approach to defeasible inheritance nets. *Artif. Intell.*, 48(2): 199–209, 1991. ISSN 0004-3702.

Victor W. Marek Marek and Miroslaw Truszczynski. *Nonmonotonic Reasoning*. Springer, Berlin, 1993.

John McCarthy. Circumscription - a form of non-monotonic reasoning. *Artificial Intelligence*, 13(1-2):27–39, 1980.

L. Thorne McCarty. Defeasible deontic reasoning. *Fundamenta Informaticae*, 21(1/2):125–148, 1994.

Michael Merz, Boris Liberman, and Winfried Lamersdorf. Using mobile agents to support interorganizational workflow management. *Applied Artificial Intelligence*, 11(6):551–572, 1997.

John-Jules Ch. Meyer and Roel J. Wieringa. Deontic logic: a concise overview. pages 3–16, 1994.

Robert C. Moore. Semantical considerations on nonmonotonic logic. *Artificial Intelligence*, 25(1):75–94, 1985. ISSN 0004-3702.

Leora Morgenstern. *The MIT Encyclopedia of the Cognitive Sciences*, chapter Nonmonotonic logics. The MIT Press, Cambridge, MA, USA, 1999.

John F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36(1):48–49, 1950.

John Von Neumann and Oskar Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 3rd edition, 1953.

Ilkka Niemelä and Patrik Simons. Smodels - an implementation of the stable model and well-founded semantics for normal LP. In Jürgen Dix, Ulrich Furbach, and Anil Nerode, editors, *LPNMR*, volume 1265 of *Lecture Notes in Computer Science*, pages 421–430. Springer, 1997. ISBN 3-540-63255-7.

Donald Nute. Defeasible reasoning. In *Proceedings of 20th Hawaii International Conference on System Science*. IEEE press, 1987.

Donald Nute. Defeasible logic. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3. 1994.

Hyacinth S. Nwana, Lyndon C. Lee, and Nicholas R. Jennings. Co-ordination in multi-agent systems. In *Software Agents and Soft Computing: Towards Enhancing Machine Intelligence, Concepts and Applications*, pages 42–58, London, UK, 1997. Springer-Verlag. ISBN 3-540-62560-7.

Andrea Omicini and Sascha Ossowski. *Intelligent Information Agents*, volume 2586 of *Lecture Notes in Computer Science*, chapter Objective versus Subjective Coordination in the Engineering of Agent Systems, pages 179–202. Springer Veglas, 2003.

Martin J Osborne and Ariel Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

Simon Parsons and Peter McBurney. Argumentation-based dialogues for agent co-ordination. *Group Decision and Negotiation*, 12(5):415–439, 2003. ISSN 0926-2644 (Print) 1572-9907 (Online).

Philip Pettit. Deliberative democracy and the discursive dilemma. *Philosophical Issues*, 11: 268–299, 2001.

Philip Pettit. When to Defer to Majority Testimony – and When Not. *Analysis*, 66(291): 179–187, 2006.

Duy Hoang Pham. Efficient representation and effective reasoning for multi-agent systems. Doctoral Consortium in International Conference on Principles of Knowledge Representation and Reasoning, Sydney, Australia, 2008.

Duy Hoang Pham, Guido Governatori, and Simon Raboczi. Agents adapt to majority behaviours. In *IEEE 2008 International Conference on Research, Innovation and Vision for Future in Computing and Communication Technologies Ho Chi Minh, Vietnam*, pages 7–12, 2008a.

Duy Hoang Pham, Subhasis Thakur, and Guido Governatori. Defeasible logic to model n-person argumentation game. In Maurice Pagnucco and Michael Thielscher, editors, *Proceedings of the Twelfth International Workshop on Non-Monotonic Reasoning*, pages 215–222, Syney, Australia, 2008b.

Duy Hoang Pham, Subhasis Thakur, and Guido Governatori. Settling on the group's goals: An n-person argumentation game approach. In *Intelligent Agents and Multi-Agent Systems, 11th Pacific Rim International Conference on Multi-Agents, PRIMA 2008, Hanoi, Vietnam, December 15-16, 2008. Proceedings*, volume 5357 of *Lecture Notes in Computer Science*, pages 328–339. Springer, 2008c. ISBN 978-3-540-89673-9.

Gabriella Pigozzi. Belief merging and the discursive dilemma: an argument-based account to paradoxes of judgment aggregation. *Synthese*, 152(2):285–298, 2006.

Jan A. Plaza. Logics of public communications. In M. L. Emrich, M. S. Pfeifer, M. Hadzikadic, and Z.W. Ras, editors, *Proceedings of the fourth international symposium on methodologies for intelligent systems: Poster session program*, pages 201–216. Oak Ridge National Laboratory, ORNL/DSRD-24, 1989.

Henry Prakken. Intuitions and the modelling of defeasible reasoning: some case studies. In Salem Benferhat and Enrico Giunchiglia, editors, *Proceedings of 9th International Workshop on Non-Monotonic Reasoning*, pages 91–102, Toulouse, France, 2002.

Henry Prakken and Giovanni Sartor. A dialectical model of assessing conflicting arguments in legal reasoning. *Artificial Intellifgence and Law*, 4(3-4):331–368, 1996.

Anand S. Rao. Agentspeak(l): BDI agents speak out in a logical computable language. In *MAAMAW '96: Proceedings of the 7th European workshop on Modelling autonomous agents in a multi-agent world : agents breaking away*, pages 42–55, Secaucus, NJ, USA, 1996. Springer-Verlag New York, Inc. ISBN 3-540-60852-4.

Anand S. Rao and Michael P. Georgeff. Modeling rational agents within a BDI architecture. In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning (KR91)*, pages 473–484. Morgan Kaufmann, 1991.

Anand S. Rao and Michael P. Georgeff. BDI agents: From theory to practice. In Lesser and Gasser (1995), pages 312–319. ISBN 0-262-62102-9.

Daniel M. Reeves, Benjamin N. Grosof, Michael P. Wellman, and Hoi Y. Chan. Towards a declarative language for negotiating executable contracts. In *Proceedings of the AAAI-99 Workshop on Artificial Intelligence in Electronic Commerce (AIEC-99)*. AAAI Press / MIT Press, 1999.

Raymond Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1-2):81–132, 1980.

Jeffrey S. Rosenschein and Michael R. Genesereth. Deals among rational agents. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pages 91–99, Los Angeles, CA, 1985. Morgan Kaufmann.

Jeffrey S. Rosenschein and Gilad Zlotkin. *Rules of encounter: designing conventions for automated negotiation among computers*. MIT Press, Cambridge, MA, USA, 1994. ISBN 0-262-18159-2.

Bram Roth, Régis Riveret, Antonino Rotolo, and Guido Governatori. Strategic argumentation: A game theoretical investigation. In *Proceedings of 11th International Conference on Artificial Intelligence and Law*, pages 81–90. ACM Press, 2007.

Nicolás D. Rotstein, Alejandro Javier García, and Guillermo Ricardo Simari. Defeasible argumentation support for an extended BDI architecture. In Iyad Rahwan, Simon Parsons, and Chris Reed, editors, *ArgMAS*, volume 4946 of *Lecture Notes in Computer Science*, pages 145–163. Springer, 2007. ISBN 978-3-540-78914-7.

Sonia V. Rueda, Alejandro J. Garcia, and Guillermo R. Simari. Argument-based negotiation among BDI agents. *Journal of Computer Science and Technology*, 2(7), 2002.

Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*, chapter Logical agent, pages 194–239. Prentice Hall, second edition, December 2002. ISBN 0137903952.

Young U. Ryu and Ronald M. Lee. Defeasible deontic reasoning: a logic programming model. pages 225–241, 1993.

Young U. Ryu and Ronald M. Lee. *Defeasible Deontic Logic*, chapter Deontic logic viewed as defeasible reasoning, pages 123–137. Kluwer Academic Publisher, Dordrecht, The Netherlands, 1997.

Jordi Sabater and Carles Sierra. Regret: reputation in gregarious societies. In *AGENTS '01: Proceedings of the fifth international conference on Autonomous agents*, pages 194–195, New York, NY, USA, 2001. ACM. ISBN 1-58113-326-X.

David Sadek. *Attitudes mentales et interaction rationnelle: vers une thorie formelle de la communication*. PhD thesis, Universit de Rennes I, France, 1991.

Chiaki Sakama and Katsumi Inoue. Coordination between logical agents. In João Alexandre Leite and Paolo Torroni, editors, *Computational Logic in Multi-Agent Systems, 5th International Workshop, CLIMA V, September 29–30, 2004, Revised Selected and Invited Papers*, volume 3487 of *Lecture Notes in Computer Science*, pages 161–177, Lisbon, Portugal, 2005. Springer. ISBN 3-540-28060-X.

Chiaki Sakama, Katsumi Inoue, Koji Iwanuma, and Ken Satoh. A defeasible reasoning system in multi-agent environment. In Ken Satoh and Fariba Sadri, editors, *CL-2000 Workshop on Computational Logic in Multi-Agent Systems*, pages 1–6, London, UK, 2000.

Ronald Schrooten and Walter Van de Velde. Software agent foundation for dynamic interactive electronic catalogs. *Applied Artificial Intelligence*, 11(5):459–481, 1997.

Yoav Shoham. Agent-oriented programming. *Artificial Intelligence*, 60(1):51–92, 1993.

Yoav Shoham and Moshe Tennenholtz. On the synthesis of useful social laws for artificial agent societies. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, pages 276–281, 1992.

Yoav Shoham and Moshe Tennenholtz. On the emergence of social conventions: modeling, analysis, and simulations. *Artificial Intelligence*, 94(1-2):139–166, 1997.

Carles Sierra. Agent-mediated electronic commerce. *Autonomous Agents and Multi-Agent Systems*, 9(3):285–301, November 2004. ISSN 1387-2532.

Guillermo R. Simari and Ronald P. Loui. A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence*, 53(2-3):125–157, 1992. ISSN 0004-3702.

Guillermo Ricardo Simari, Alejandro Javier García, and Marcela Capobianco. Actions, planning and defeasible reasoning. In Delgrande and Schaub (2004), pages 377–384. ISBN 92-990021-0-X.

Munindar P. Singh and Michael N. Huhns. Multiagent systems for workflow. In *International Journal of Intelligent Systems in Accounting, Finance and Management*, volume 8, pages 105–117. John Wiley & Sons, Ltd., 1999.

Kaile Su, Abdul Sattar, Guido Governatori, and Qingliang Chen. A computationally grounded logic of knowledge, belief and certainty. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 149–156, New York, NY, USA, 2005. ACM. ISBN 1-59593-093-0.

Katia P. Sycara. Multiagent systems. *AI Magazine*, 19(2):79–92, 1998.

Katia P. Sycara, Anandeep Pannu, Mike Williamson, Dajun Zeng, and Keith Decker. Distributed intelligent agents. *IEEE Expert*, 11(6):36–46, 1996.

Subhasis Thakur, Guido Governatori, Vineet Padmanabhan, and Jenny Eriksson Lundström. Dialogue games in defeasible logic. In Mehmet A. Orgun and John Thornton, editors, *Australian Conference on Artificial Intelligence*, volume 4830 of *Lecture Notes in Computer Science*, pages 497–506. Springer, 2007. ISBN 978-3-540-76926-2.

S. Rebecca Thomas. The PLACA agent programming language. In *ECAI-94: Proceedings of the workshop on agent theories, architectures, and languages on Intelligent agents*, pages 355–370, New York, NY, USA, 1995. Springer-Verlag New York, Inc. ISBN 3-540-58855-8.

Paolo Torroni. Computational logic in multi-agent systems: Recent advances and future directions. *Annals of Mathematics and Artificial Intelligence*, 42(1-3):293–305, 2004. ISSN 1012-2443.

Maksim Tsvetovatyy, Maria L. Gini, Bamshad Mobasher, and Zbigniew Wieckowski. Magma: An agent based virtual market for electronic commerce. *Applied Artificial Intelligence*, 11(6): 501–523, 1997.

Johan van Benthem and Fenrong Liu. Dynamic logic of preference upgrade. Technical report Report PP-2005-29, University of Amsterdam, 2005.

Johan van Benthem, Jan van Eijck, and Barteld Kooi. Logics of communication and change. *Information and Computation*, 204(11):1620–1662, 2006. ISSN 0890-5401.

Wiebe van der Hoek and Michael Wooldridge. Towards a logic of rational agency. *Logic Journal of the IGPL*, 11(2):135–159, 2003.

Hans van Ditmarsch. Prolegomena to dynamic logic for belief revision. *Synthese (Knowledge, Rationality & Action)*, 147:229–275, 2005.

Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. *Dynamic Epistemic Logic*. (Synthese Library). Springer, 1st edition, November 2007. ISBN 1402069081.

Govert van Drimmelen. Satisfiability in alternating-time temporal logic. In *LICS '03: Proceedings of the 18th Annual IEEE Symposium on Logic in Computer Science*, pages 208–217, Washington, DC, USA, 2003. IEEE Computer Society. ISBN 0-7695-1884-2.

Birna van Riemsdijk, Wiebe van der Hoek, and John-Jules Ch. Meyer. Agent programming in dribble: from beliefs to goals using plans. In *AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 393–400, New York, NY, USA, 2003. ACM. ISBN 1-58113-683-8.

Javier Vázquez-Salceda, Huib Aldewereld, and Frank Dignum. Norms in multiagent systems: from theory to practice. *Computer Systems: Science & Engineering*, 20(4), 2005.

Bart Verheij. *Rules, Reasons, Arguments. Formal studies of argumentation and defeat.* Phd thesis, Universiteit Maastricht, Holland, 1996.

Georg H. von Wright. Deontic logic. *Mind*, 60:58–74, 1951.

Gerard A. W. Vreeswijk. Abstract argumentation systems. *Artificial Intelligence*, 90(1–2): 225–279, 1997. ISSN 0004-3702.

Dirk Walther, Carsten Lutz, Frank Wolter, and Michael Wooldridge. Atl satisfiability is indeed exptime-complete. *J. Log. Comput.*, 16(6):765–787, 2006.

Gerhard Weiss, editor. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press, 1999. ISBN 0262232030.

Cees Witteveen. Partial semantics for truth maintenance. In P. Doherty, editor, *Partiality, Modality and Nonmonotonicity*, pages 197–222, Stanford, California, 1996. CSLI publications.

Michael Wooldridge. Computationally grounded theories of agency. In *ICMAS '00: Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS-2000)*, page 13, Washington, DC, USA, 2000. IEEE Computer Society. ISBN 0-7695-0625-9.

Michael Wooldridge. *Introduction to MultiAgent Systems*. John Wiley & Sons, Chichester, England, June 2002. ISBN 047149691X.

Michael Wooldridge and Nicholas R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995a.

Michael Wooldridge and Nicholas R. Jennings. Agent theories, architectures, and languages: a survey. In *ECAI-94: Proceedings of the workshop on agent theories, architectures, and languages on Intelligent agents*, pages 1–39, New York, NY, USA, 1995b. Springer-Verlag New York, Inc. ISBN 3-540-58855-8.

Haizheng Zhang and Victor Lesser. Multi-agent based peer-to-peer information retrieval systems with concurrent search sessions. In *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 305–312, New York, NY, USA, 2006. ACM. ISBN 1-59593-303-4.

Defeasible reasoning algorithm

In this appendix, we present the algorithms for computing the definite and defeasible conclusions.

7.3 Basic defeasible theory

A basic defeasible theory does not have any defeater rule. Also, the superiority relation between the defeasible rules is removed. Therefore, the basic defeasible theory has only a set of strict and defeasible rules. A standard defeasible theory can be transformed to a basic theory according to Antoniou et al. (2001). Assume that we have a basic defeasible theory D, which contains a set of strict and defeasible rules R_s and R_d respectively, and a set of literals $\mathscr{L}_D$. Given a rule r, $A(r)$ represents the antecedence (or the body) whilst $C(r)$ represents the consequent (or the head). If a literal q is supported by the rule r, the we have the representation of $r[q]$. In what follows, we present the algorithms for computing the extension of a defeasible theory.

7.4 Defeasible reasoning algorithm

The two algorithms for deriving two types of conclusions are illustrated in the next two sections.

7.4.1 Algorithm for definite conclusions

The algorithm consists of two cycles. The first cycle verifies whether every literal occurs in the head of a rule in D. The second cycle performs the forward chaining on the set of strict rules

R_s and modifies the structure of the rules in the case that a literal in the body of rules has been proved. These cycle stops when all of the strict rules or all of the literals have been investigated.

Algorithm 1: Algorithm for definite conclusions

Input: a defeasile theory D

Output: sets of definite conclusions Δ^+ and Δ^-

```
1  while 𝓛_D ≠ ∅ and R_s ≠ ∅ do
2      foreach l ∈ 𝓛_D do
3          if R_s[l] = ∅ then
4              Δ⁻ = Δ⁻ + {l};
5              R_s = R_s − {r | l ∈ A(r)}
6          if R_s[l] = ∅ and R_s[∼l] = ∅ then 𝓛_D = 𝓛_D − {l, ∼l}
7      foreach r ∈ R_s do
8          if l ∈ A(r) and l ∈ Δ⁺ then
9              A(r) = A(r) − {l}
10         if A(r) = ∅ and C(r) = q then
11             Δ⁺ = Δ⁺ + {q};
12             R_s = R_s − {r}
```

7.4.2 Algorithm for defeasible conclusions

In this algorithm, R_{sd} is the set of strict and defeasible rules. This algorithm has two main steps. The first step checks if each literal in D is supported by a rule in R_{sd}. The second step investigates a rule in R_{sd}. If there is a rule with an empty body, the theory D should not provide the complement of the literal in the head of that rule. Also, the literal in the head is in the positive conclusions ∂^+ provided that there is no support for the opposite in D ($R_{sd}[q] = \emptyset$). These steps are repeated until one of following conditions is met, (1) all of the literals are verified, (2) all of the rules are investigated, (3) there is no new conclusion to be derived.

Input: a defeasile theory D

Output: sets of defeasible conclusions ∂^+ and ∂^-

while $\mathscr{L}_D \neq \emptyset$ *or* $R_{sd} \neq \emptyset$ *or* $(\partial^{\pm})' \neq \partial^{\pm}$ **do**
  foreach $l \in \mathscr{L}_D$ **do**
    if $R_{sd}[l] = \emptyset$ **then**
      $\partial^- = \partial^- + \{l\}$;
      $R_{sd} = R_{sd} - \{r : a \in A(r)\}$;
    if $R_{sd}[l] = \emptyset$ *and* $R_{sd}[\sim l] = \emptyset$ **then** $\mathscr{L}_D = \mathscr{L}_D - \{l, \sim l\}$;
  foreach $r \in R_{sd}$ **do**
    if $l \in A(r)$ *and* $l \in \partial^+$ **then** $A(r) = A(r) - \{l\}$;
    if $A(r) = \emptyset$ *and* $C(r) = q$ **then**
      $\partial^- = \partial^- + \{\sim q\}$;
      if $R_{sd}[\sim q] = \emptyset$ **then**
        $\partial^+ = \partial^+ + \{q\}$;
        foreach $s \in R_{sd}$ **do**
          if $q \in A(s)$ **then** $A(s) = A(s) - \{s\}$;
        $R_{sd} = R_{sd} - \{r\}$;
        $R_{sd} = R_{sd} - \{t | t \in R_{sd}$ and $\sim q \in A(t)\}$;

Algorithm 2: Algorithm for definite conclusions