Interest-based Negotiation in Multi-Agent Systems

Iyad Rahwan

Submitted in total fulfilment of the requirements
of the degree of Doctor of Philosophy
August 2004

Department of Information Systems
The University of Melbourne

Abstract

Software systems involving autonomous interacting software entities (or agents) present new challenges in computer science and software engineering. A particularly challenging problem is the engineering of various forms of interaction among agents. Interaction may be aimed at enabling agents to coordinate their activities, cooperate to reach common objectives, or exchange resources to better achieve their individual objectives. This thesis is concerned with negotiation: a process through which multiple self-interested agents can reach agreement over the exchange of scarce resources. In particular, I focus on settings where agents have limited or uncertain information, precluding them from making optimal individual decisions. I demonstrate that this form of bounded-rationality may lead agents to sub-optimal negotiation agreements. I argue that rational dialogue based on the exchange of arguments can enable agents to overcome this problem. Since agents make decisions based on particular underlying reasons, namely their interests, beliefs and planning knowledge, then rational dialogue over these reasons can enable agents to refine their individual decisions and consequently reach better agreements. I refer to this form of interaction as “interested-based negotiation.”

The contributions of the thesis begin with a conceptual and formal framework for interest-based negotiation among computational agents. Then, an exploration of the differences between this approach and the more traditional bargaining-based approaches is presented. Strategic issues are then explored and a methodology for designing negotiation strategies is developed. Finally, the applicability of the framework is explored through a pilot application that makes use of interest-based negotiation in order to support cooperative activity among mobile users.

Declaration

This is to certify that

1. the thesis comprises only my original work towards the PhD except where indicated in the Preface
2. due acknowledgement has been made in the text to all other material used,
3. the thesis is less than 100,000 words in length, exclusive of tables, bibliographies and appendices.

Iyad Rahwan

To my parents . . .

with love and gratitude

Preface

> *"... all this sliding to one side and then the other, as if we were crazy or, worse, lost, is actually our way of going straight, scientifically exact and, so to speak, predetermined, although it undoubtedly resembles a random series of errors, or rethinkings, but only in appearance, because really it's just our way of going where we have to go, the way that is specifically ours, our nature, so to speak ... I find it very reassuring, that there is an objective principle behind all the stupid things we do."*
>
> Alessandro Baricco, 'City'

The question I ask myself most often now is: why did I do it? Why would anyone spend three to four years of their life doing the same thing? In fact, I asked myself this question repeatedly before and during my candidature. And every time, I find a different answer. When I applied for admission to the PhD degree at the University of Melbourne, I wanted to achieve two main objectives: to demonstrate to myself that I am capable of acquiring a higher degree; and to give myself the creative edge and credentials to establish an Internet business.[1] The former objective became insignificant as I was greatly humbled by the vast technical challenges of research and the many great minds that pursue it. The latter objective also faded away as I found research to be very rewarding.

I also had a 'hidden' objective, which later became my main motivation: to do something that would make me think about conceptual problems, that would enable me to learn about the human mind, about human relationships, about correct reasoning, and enable me to understand the world better. The *Intelligent Software Agents* field was an ideal candidate. It was (and still is) a field that is technologically appealing, funky, sufficiently challenging, and had a lot of interdisciplinary work going on. The latter characteristic meant that one could read about logic, philosophy, social sciences, economics, cognitive sciences, or even critical theory, and still be regarded as doing research in computer science (perhaps not by all though!). The field is relatively 'immature' that almost 'anything goes.' For example, some researchers adapt argumentation schemes from philosophy and

---

[1] I submitted my application for admission in 1999, right before the Information Technology market collapse.

informal logic to engineer flexible dialogues among software agents. Others use concepts from economic theory to study properties of equilibrium and rationality. Yet other people use social theory as an inspiration for specifying notions such as norms, roles, and obligations in distributed systems. This ‘luxury’ is what makes the field very exciting, rewarding, and liberating. This is perhaps why I was drawn to research in this area.

During my PhD, I got drawn into many areas of scientific, social and philosophical inquiry. I got distracted uncountable times. But in hind sight, I think this was what I needed to do. This was my own way of going where I had to go –as the quote at the beginning of this preface very elegantly states. I am grateful that the PhD experience taught me many essential lessons, about what it means to be a researcher, and about the power and joy of interdisciplinary research.

The PhD experience has provided me with the opportunity to meet and work with amazing people –people who have made lasting impressions on me: attending a talk in Liverpool by Professor Johan van Benthem on logic and games; discussing the value of research with Ronald Ashri over coffee in Southampton; discussing negotiation among humans versus software agents with Maqsoud Kruse in Melbourne; listening to Andrew Clausen talk about the power of mathematics on the train home; exchanging thoughts on economic theory with Peter McBurney. These experiences have changed me forever.

Among the people who inspired me are those with whom I co-authored papers during my PhD candidature. I am very grateful to their contributions to the ideas presented in this thesis, and their genuine guidance and support. Below, I list the publications upon which the thesis is based. I note that every chapter is based on published work of which I am the first author, and for which I did the majority of the work.

- The literature review presented in Chapter 2 is based on a joint survey paper with Sarvapali Dyanand (Gopal) Ramchurn, Nick Jennings, Peter McBurney, Simon Parsons, and Liz Sonenberg. The paper was published in the Knowledge Engineering Review [Rahwan et al., 2003b]. Moreover, the characterisation of classical negotiating agents and argument-based negotiating agents appeared in a more detailed form in a joint paper with Ronald Ashri and Michael Luck, published in the proceedings of the Central and Eastern European Conference on Multi-Agent Systems (CEEMAS 2003), which was held in Prague [Ashri et al., 2003].

- Chapter 3 is based on work with Frank Dignum and Liz Sonenberg published in the 2nd International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2003), which was held in Melbourne [Rahwan et al., 2003c]. An updated version of this paper was published in the post-proceedings of the AAMAS 2003 workshop on Agent Communiction Languages (ACL) [Rahwan et al., 2004c]. The chapter also benefited from useful feedback from Rogier van Eijk and Leila Amgoud.

- Chapter 4 is based on work with Peter McBurney and Liz Sonenberg that was published in the proceedings of the AAMAS 2004 workshop on Argumentation in Multi-Agent Systems, which was held in New York [Rahwan et al., 2004b]. I also benefited from discussions with Andrew Clausin.

- Chapter 5 has its roots in a joint attempt with Peter McBurney to characterise the factors that enter into consideration when designing negotiation strategies [Rahwan et al., 2003a]. This work was published in an AAMAS workshop on Game Theoretic and Decision Theoretic Agents (GTDT 2003). The chapter was developed mainly by myself as an extension of that work. Towards the end of my candidature, I received advice from Peter McBurney and Nick Jennings regarding a planned joint publication based on the chapter.

- Chapter 6 has its roots in a joint paper with Talal Rahwan, Tarek Rahwan, and Ronald Ashri [Rahwan et al., 2004d], which was published in the AAMAS 2003 International Workshop on Agent-Oriented Information Systems (AOIS). The work started as an attempt to use agent-based reasoning to provide intelligent support to a single user using a mobile device. The idea then developed towards using automated negotiation to support interaction among multiple users. This was joint work with Connor Graham and Liz Sonenberg, and was published in the Proceedings of the 7th Pacific Rim International Workshop on Multi-Agents (PRIMA 2004) held in Auckland [Rahwan et al., 2004a]. Additional work that influenced the chapter indirectly was published jointly with Fernando Koch both in and the AAMAS 2004 workshop on Agents for Ubiquitous Computing (UbiAgents) and in PRIMA 2004 [Koch and Rahwan, 2004a,b]. Moreover, Anton Kattan and Fernando Koch did a significant part of the implementation discussed in this chapter.

I am very grateful to my co-authors and everybody who contributed to this thesis, and I hope that our collaboration continues and grows. It was an honour.

Iyad Rahwan
Melbourne, August 2004

Acknowledgements

> *"And, when you want something,*
> *all the universe conspires in helping you to achieve it."*
>
> Paolo Coelho, 'The Alchemist'

Perhaps unsurprisingly, doing a Ph.D. is like going through hardship; it bonds you with those people who, in the midst of all your random wandering, help you, support you, listen to you, care about you, pat on your back, provide you with directions, or simply, love you. These people deserve a great deal of credit, because the thesis would not have come to existence without them.

My greatest gratitude goes to my family. I thank my mother, Alaa Baaj, who taught me principle, determination, morality and the importance of reason. I thank her for patiently answering my childhood questions, sometimes giving up her resting time in order to do so. I thank my father, Ahmad Rahwan, who always ignited my intellectual curiosity, and who supported me, despite the enormous financial difficulty, on my travel adventure to Australia five years ago. And I thank him for teaching me the value of being open to new people, cultures, and ideas, which I consider to be an essential precondition of doing research. I thank my brothers, Talal and Tarek, for being my best friends, and for making me always know I am part of a team. I also thank the members of my extended family in Syria for their emotional support.

The true silent warrior behind this PhD is undoubtedly my beloved wife Zoe. With her love, dedication, and sacrifice, she made the journey most enjoyable. Her discipline and dedication to her work were an inspiration. Moreover, she worked relentlessly over the final few months before submission to make sure I did not have to do any cooking or house work, and concentrate on thesis writing instead (not to mention her delicious cookies, which made a big impact on the quality of the thesis). If she ever decides to do a PhD in economics, I hope I can live up to her standards. I also thank Zoe's family and friends, my own new family and friends, for their support. I particularly thank Graeme Jones for his interest in my work, and for always reminding me to take things easy. I am particularly indebted to Robyn Ward and Anna Uebergang for doing a wonderful job at proof-reading chapters of the thesis. Thanks to Diane Whitcombe for offering help.

My close friends deserve a huge ‘thank you’ for providing me with support, and for sharing with me the simple joys along the way. I thank my dear friend Maqsoud Kruse for his wonderful friendship, from high-school years in Abu Dhabi, to university years in Al-Ain, through to postgraduate studies in Melbourne. Sharing a dorm room with a psychologist was a major inspiration of my interest in Artificial Intelligence. Together, we had our first encounters with the wonders of research and critical thinking and it was a pleasure to share this fascination and have someone who would listen and give insightful feedback. Maqsoud, I thank you for your support, for your feedback, for your friendship, for your pleasant idiosyncrasy, and for the many things we share. I have no doubt that our spontaneous and interesting discussions will continue for many years to come. Very big thanks to Julius Anton Kattan, for being a great colleage and friend, for his company during late-nights at the office, and for supporting me and putting up with me over all university years. Thanks also to Connor Graham for the interesting corridor chats. I hope you get through your PhD quicker than I did. Thanks to Connor also for his valuable input into the Human-Computer Interaction issues discussed in chapter 6. I thank my ex-flatmate, Anders Rundqvist, for the fun times we had thinking about business ideas before I started the PhD. Perhaps one day we might start up a business! Thanks also to Yousseff El-Khatib and Jason Battye for their constant encouragement and support.

Now moving academia, I must start by thanking my primary school mathematics teacher, Ms Hamida, for being a great teacher, and for being the first person to make me, as a child, desire to be a researcher; although at the time, I envisaged myself in a white lab coat with messy big hair. Sincere thanks to all my teachers during my school and university years. At the United Arab Emirates University, I thank Jawad Berri for igniting my interest in the field of Artificial Intelligence, and Hassan Selim for providing appeal to the business perspective of applying technology.

I am forever indebted to my PhD supervisor, Prof. Liz Sonenberg, for her incredible support. Liz supported me from my very early steps into research, when I had no idea what I was doing, until the handing of my thesis, and beyond. She was and remains to be a truly inspiring example of a person who could balance extreme professionalism with care and humanity, a balance I will always aspire to. I thank Liz for recruiting me, for following my steps closely, for being critical when I needed direction, for encouraging me when I was feeling depressed, and for defending my rights and self-confidence when I was unjustly criticised. I thank Liz for encouraging me to apply for funding, which allowed me to visit the University of Southampton and the University of Liverpool, for supporting part of the trip from her own funds, for guiding me through various grant and job application writing procedures, and for providing supportive reference letters. A lot can happen during PhD years, and I thank Liz for accommodating my personal and family-related happenings, and for giving me the space I needed. I also thank her for doing a superb job at speeding up the administrative work in order to help me finish early to take a position at the British University in Dubai. Liz has been a truly inspiring figure in my life, and has changed me forever. My gratitude to her is limitless. In this context, I owe David Kinny, from Agentis International Inc., a great deal of gratitude for introducing me to Liz. This took place when I was a student of his in the subject "Software Agents." David was an inspiring teacher, and it was due to his engaging style and enthusiasm that I became determined toursue my PhD in this field.

A number of people have actively contributed to the research presented in this thesis, either through co-authorship of papers, or by simply providing their insightful advice. I thank Frank Dignum, Rogier van Eijk and Fernando Koch of Utrecht University; Nick Jennings, Michael Luck, Ronald Ashri, and Sarvapali Ramchurn of the University of Southampton; Peter McBurney of the University of Liverpool; Simon Parsons of Brooklyn College of the City University of New York; Connor Graham of the University of Melbourne; and Ryszard Kowalczyk and Ha Pham from CSIRO; I particularly thank Ryszard for his supervision at CSIRO, for agreeing to sit on the committee for my second year report, and for the many hours he put into guiding me at the early stages of my candidature. I thank Wayne Wobcke of the University of New South Wales for helping me formulate my research question when he was a senior fellow at Melbourne. Thanks to Mikhail Prokopenko for his reality-checks and feedback when he was my supervisor at CSIRO in Sydney. I also thank Peter Wang of CSIRO for sharing his room with me, and for being a great office mate. I thank Michael Rovatsos of the University of Edinburgh, and Christian Guttmann of Monash University, for the many interesting chats over coffee and food, and for our exchange of PhD-related experiences.

I thank the following people at Swinburne University, where I started my PhD before transferring to the University of Melbourne. I thank Alex Buick and Vicki Formosa for their help in making the transition easy, and for making me feel genuinely cared for. I also thank my Swinburne supervisor, Yun Yang, for his understanding of my transfer decision. Finally, my sincere apology goes to Ahmet Sekercioglu, my initial designated supervisor, for not being able to follow through with my commitment.

Visiting the University of Southampton for two months was an incredible experience, both academically and personally. I am grateful to Nick Jennings for inviting me to visit his research group, and for making the time to fit me in his tight schedule. Thanks to members of the IAM lab who helped make my stay productive and enjoyable; I thank: Viet Dung Dang, Raj Dash, Nadim Haque, Nishan Karunatillake, Michael Luck, Steve Munroe, Thuc Dong (John) Nguyen, Cora Excelente Toledo, Fabiola Lopez, Shamima Paurobally, Sarvapali Dyanand (Gopal) Ramchurn, and Philip Turner. Particular thanks to Ronald Ashri for being a wonderful companion during my visit, and to his wife Katia for their generous hospitality.

During my PhD, I met many inspiring and remarkable people. Peter McBurney of the University of Liverpool has made a profound impact on my academic experience. I am grateful to Peter for responding to my emails when I first inquired about one of his papers, and for the very interesting discussions that followed (even before we met in person). These email ‘chats’ made a world of difference; they helped me clarify my ideas, justify my choices, and give me the confirmation and confidence I needed. I also thank Peter for inviting me to visit him at the University of Liverpool, for making the time to chat, show me around, introduce me to people. Words cannot adequately describe my indebtedness to you, Peter, for your friendship, mentorship, guidance and for teaching me (by simply being yourself) how a truly open-minded researchers should be. I also thank other people who made me feel welcome at the University of Liverpool. In particular, I thank Michael Wooldridge, Katie Atkinson, Alison Chorley, Rafael Bordini, Shaheen Fatima, Wiebe van der Hoek, Sieuwert van Otterloo, Marc Pauly, and Steve Phelps.

I thank Frank Dignum of Utrecht University for his enormous support and encouragement and for his valuable advice, particularly during his visits to the University of Melbourne.

Locally, I thank the staff at the Department of Information Systems, University of Melbourne, for supporting me in various ways, academically, administratively, and socially. In particular I thank: Janine Campbell, Jennie Carroll, Vicki Jackson, Robert Johnston, Tanya Linden, Sean Maynard, James Morrison, Jon Pearce, Marg Ross, Graeme Shanks, Peter Seddon, and Csaba Veres. Thanks also to my student colleagues, particularly: Atif Ahmad, Shawn Alborz, Yu Ni Ham, Jeni Paay, Sonja Pedell, Krishna Venkitachalam, and Leslie Young. Thanks to Steven Bird, at the Department of Computer Science and Software Engineering, for agreeing to be on my second year report committee. I also owe a great deal of gratitude to Paul Gruba, who volunteered to give his popular thesis writing course at our department. His course was a key factor in helping me shape my thesis structure. Finally, I thank Nasser Spear for his encouragement over coffee chats.

Still locally, thanks to members of the University of Melbourne Agent Laboratory for showing interest in my work, and for creating a stimulating research environment: Muthukkaruppan Annamalai, Kevin Chan, Steve Goschnick, Clinton Heinze, Thomas Juan, Samin Karim, Kendall Lister, Emma Norling, Ayodele Oluyomi, Matt Pattison, Michael Papasimeon, Adrian Pearce, Kylie Pegg, Don Perugini, Budhitama Subagdja, Raymond So, Susannah Soon, Leon Sterling, and Peter Wallis. Particular thanks to Andrew Clausen for re-igniting my interest in mathematics during my write-up phase, an inspiration I hope to pursue after submission. I also thank the following members of 'Agents in Victoria': Shonali Krishnawamy, Lin Padgham, John Thangarajah, and Michael Winikoff.

I am grateful to the following academics for their help, either through providing me with their papers, or answering my queries about their research: Leila Amgoud, John Bell, Gerhard Brewka, Joris Hulstijn, Henry Prakken, Heine Rasmussen, Daniel Reeves, Katia Sycara, Fernando Tohmé, and Gerard Vreeswijk.

Attending conferences is a very crucial and highly educational part of the PhD candidature. I thank the following people for showing interest in my work and for discussions during conferences and workshops I attended: Juan-Antonio Rodriguez-Aguilar, Brahim Chaib-Draa, Mehdi Dastani, Virginia Dignum, Ya'akov (Kobi) Gal, Andrew Gilpin, Andrea Giovannucci, Sergio Guadarrama Cotado, David Hales, Sarit Kraus, Eleftheria Kyriazidou, Sanjay Modgil, Omer Rana, Antonio Ribeiro, Jeff Rosenschein, Onn Shehory, Carles Sierra, Guillermo Simari, and Paolo Torroni. I also thank the anonymous reviewers who reviewed my submissions and provided useful feedback. Finally, thanks to my co-initiators of the First International Workshop on Argumentation in Multi-Agent Systems: Antonis Kakas, Peter McBurney, Nicolas Maudet, Pavlos Moraitis, Chris Reed, and Simon Parsons.

Many thanks to Fernando Koch and Julius Anton Kattan for their invaluable effort in the implementation of 3APL-M and the negotiation application presented in chapter 6. Without them, that chapter would have taken me weeks longer to finish.

I thank Habib Talhami of the British University in Dubai (BUiD) for his support during the preparation of the final version of the thesis based on examiners' comments.

Finally, I would like to acknowledge financial support I received during my candidature. Thanks to the Commonwealth Department of Education, Training and Youth Affairs (DETYA) for their financial support through the International Postgraduate Research Scholarship (IPRS), which supported me prior to gaining permanent residency in Australia. Thanks to Swinburne University of Technology, where I initially started my PhD for the financial support I received through the stipend scheme. Thanks to the University of Melbourne for providing me with a stipend through the Melbourne Research Scholarship (MRS) scheme, and for providing me with financial support through the Postgraduate Overseas Research Experience Scholarship (PORES), which allowed me to spend three months as an academic visitor in the UK. Thanks to the Commonwealth Scientific and Industrial Research Organisation's (CSIRO) Mathematical Information Sciences (CMIS) division for their financial support through the Ph.D. top-up scholarship scheme.

Contents

---

[2] The narrative development and prototype implementation presented in this chapter are a result of work done jointly with Connor Graham, Anton Kattan, and Fernando Koch.

List of Figures

List of Tables

Chapter 1

Introduction

> *"My darling, I have much to say*
> *Where o precious one shall I begin?"*
>
> Nizar Qabbani, Syrian Poet (1923 – 1998)

In this chapter, I motivate and define my research question.

1.1 Brief Background

The increasing complexity of distributed computer systems has led researchers to utilise various tools of abstractions in order to improve the software engineering process. However, the requirements of an increasing number of computing scenarios go beyond the capabilities of traditional computer science and software engineering abstractions, such as object-orientation. According to Zambonelli and Parunak [2004], four main characteristics distinguish future software systems from traditional ones:

1. *Situatedness*: Software components execute in the context of an environment, which they can influence and be influenced by;
2. *Openness*: Software systems are subject to decentralised management and can dynamically change their structure;
3. *Locality in control*: Software systems components represent autonomous and proactive loci of control;

4. *Locality in interactions*: Despite living in a fully connected world, software components interact according to local (geographical or logical) patterns.

These characteristics have led to the emergence of a new paradigm in computing: the agent paradigm. An increasing number of computer systems are being viewed in terms of multiple, interacting *autonomous agents*. This is because the multi-agent paradigm offers a powerful set of metaphors, concepts and techniques for conceptualising, designing, implementing and verifying complex distributed systems [Jennings, 2001]. An agent is viewed as an encapsulated computer system that is situated in an environment and is capable of flexible, autonomous action in order to meet its design objectives [Wooldridge, 2002]. Applications of agent technology have ranged from electronic trading and distributed business process management, to air-traffic and industrial control, to health care and patient monitoring, to gaming and interactive entertainment [Jennings and Wooldridge, 1998, Parunak, 1999].

In multi-agent systems, agents need to interact in order to fulfil their objectives or improve their performance. Generally speaking, different types of interaction mechanisms suit different types of environments and applications. Agents might need mechanisms that facilitate information exchange, coordination (in which agents arrange their individual activities in a coherent manner), collaboration (in which agents work together to achieve a common objective), and so on. One such type of interaction that is gaining increasing prominence in the agent community is *negotiation*. We offer the following definition of negotiation, adapted from work on the philosophy of argumentation [Walton and Krabbe, 1995]:

> *Negotiation is a form of interaction in which a group of agents, with conflicting interests and a desire to cooperate, try to come to a mutually acceptable agreement on the division of scarce resources.*

The use of the word "resources" here is to be taken in the broadest possible sense. Thus, resources can be commodities, services, time, money etc. In short, anything that is needed to achieve something. Resources are "scarce" in the sense that competing claims over them cannot be fully simultaneously satisfied. In a multi-agent system context, the challenge of automated negotiation is to design mechanisms for allocating resources among software processes representing self-interested parties, be these parties human individuals, organisations, or other software processes.

For the sake of illustration, we list two example automated negotiation scenarios. With each scenario, I attempt to give the reader an idea of the non-trivial challenges it poses.

Telecommunications Markets: In this scenario, a software agent, representing the user and running on a smart telephone, negotiates with other software agents representing telephone companies. This is depicted in Figure 1.1(a). The user agent attempts to find a good quality phone connection for a reasonable price. Each telephone company agent aims at maximising its company's profit. The resources in need of (re-)allocation are money and communication bandwidth. Conflict of interests exists because the user competes on money and bandwidth with the various service providers, while service providers themselves compete over the user's money. Typical challenges that arise in this context include: What trading mechanism should agents use? Should there be a moderator to monitor the transactions? If an auction mechanism is used to allocate bandwidth, what bidding strategies are optimal? Should a provider's bidding strategy be dependent on how much bandwidth is available? And so on.

Dynamic Supply Chains: In this scenario, a software agent acting on behalf of a computer manufacturer negotiates with various supplier agents, in order to secure the deliver of various components. Each supplier agent might itself negotiate with subcontractors to get the components it needs. This is depicted in Figure 1.1(b). In this scenario, the negotiation mechanism involves allocating money and commodities (computer components). Each party aims at making more money, and hence the different monitor and printer cartridge suppliers compete over contracts with the computer and printer manufacturers respectively. Typical issues that arise in this situation include: What negotiation protocol to use? What happens if a contractor fails to delivery on time, or commits more than its capacity? Do we need some measure of the reliability of different contractors, and how do we use such measure in making decisions about allocation? And so on.

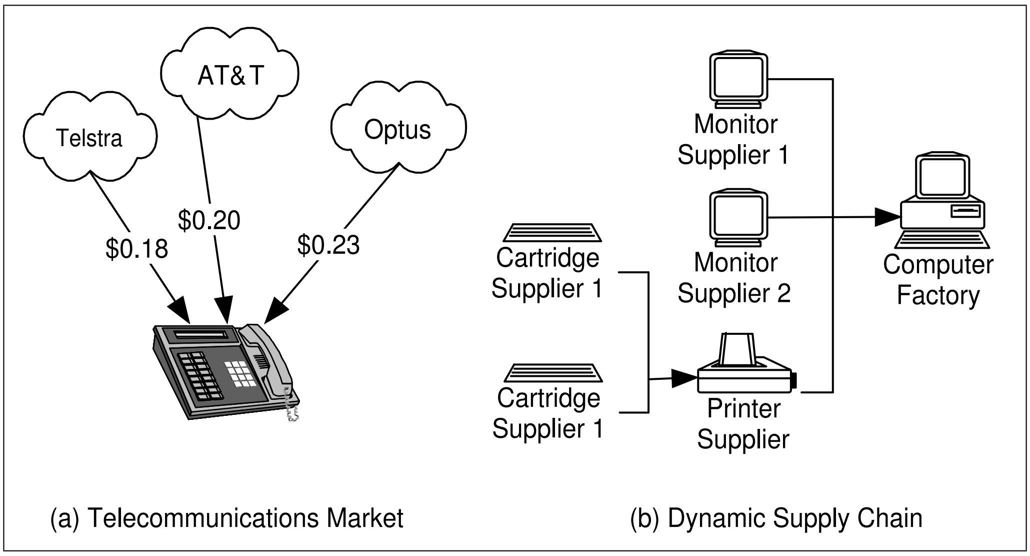


Figure 1.1: Automated Negotiation Scenarios

One might ask, what sorts of properties do designers of negotiation mechanisms aim at? Following is a list adapted from Rosenschein and Zlotkin [1994]:

1. *Simplicity:* A mechanism that requires less computational processing and communication overhead is preferable to one that does not.
2. *Efficiency:* A mechanism is efficient if it produces a good outcome. What is meant by 'good,' however, may differ from one domain to another.
3. *Distribution:* It is preferable to have a negotiation mechanism that does not involve a central decision-maker.
4. *Symmetry:* The mechanism should not be biased for or against some agent based on inappropriate criteria. Again, what constitutes an 'inappropriate' criterion depends on the domain.
5. *Stability:* A mechanism is stable if no agent has incentive to deviate from some agreed-upon strategy or set of strategies (e.g. to bid truthfully in an auction).
6. *Flexibility:* By this property, I mean that the mechanism should lead to agreement even if agents did not have complete and correct private information in relation to their own decisions and preferences. This property, which is the focus of this thesis,

requires a mechanism for enabling agents to refine their decisions, in light of new information, during the negotiation process.

Various interaction and decision mechanisms for automated negotiation have been proposed and studied. Frameworks for automated negotiation have been studied analytically using game-theoretic[1] techniques [Rosenschein and Zlotkin, 1994, Sandholm, 2002a] and logic-based techniques [Wooldridge and Parsons, 2000], as well as experimentally by programming and testing actual systems [Faratin, 2000, Kraus, 2001, Fatima et al., 2002]. These negotiation frameworks are mainly based on the exchange of offers, where an offer is a suggestion by a participant of a potential allocation of the resources in question, according to some interaction protocol. A bid in an English Auction is an example of an offer.

Analytical and empirical techniques have helped produce mechanisms that satisfy the properties discussed above to varying degrees. However, the *flexibility* property has only begun to receive attention in the multi-agent community. I will discuss this further in the following section.

1.2 Problem Statement

Controversially, most frameworks for automated negotiation often assume that agents have complete, pre-set and fixed preferences (or evaluation criteria) over negotiation outcomes (or agreements, or deals), as well as complete awareness of the space of possible outcomes. In other words, agents have already made decisions about what they need from one another, and how different deals satisfy these needs. All that is left is to jointly *search* for a deal that is satisfactory enough to all participants. Negotiation, in this sense, involves the exchange of suggested potential deals, which are then evaluated against the predetermined preferences until a deal is found that is acceptable to all parties.

However, in many application domains, the above strong conditions are not satisfied. Agents may begin negotiating without having complete and accurate preferences over alternative deals. This may be because agents have limited, uncertain, or false in-

---

[1] Game-theory is a collection of analytical tools designed to help understand and analyse strategic interaction among economically rational agents [Osborne and Rubinstein, 1994].

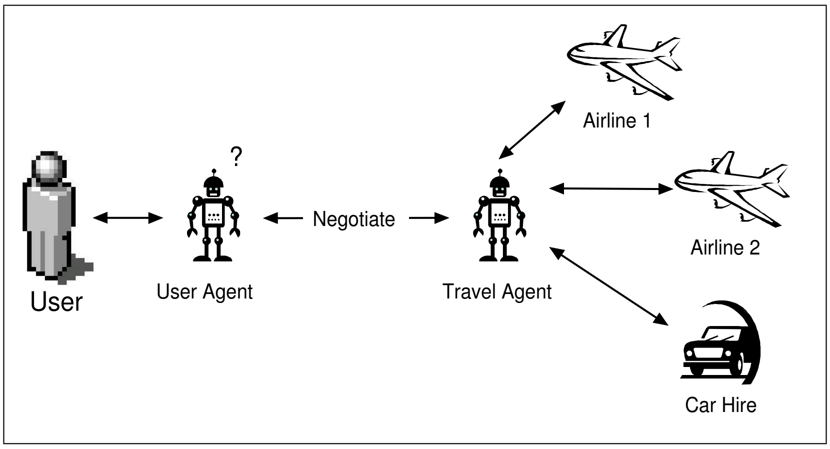


Figure 1.2: Travel Agent Scenario

formation, due to imperfect sensing of the environment or because they lack the time or computational resources needed to fully process information. As a result, agents may have incomplete or incorrect preferences over different deals. Negotiating on the basis of such preferences can lead to sub-optimal deals. For example, an agent organising a trip from Melbourne to Sydney on behalf of a user (depicted in Figure 1.2) might not know about all possible ways of travelling to Sydney, or might believe (wrongly) that flying to Sydney is more expensive than driving there. Further, the user might have forgotten to request a hotel booking. Due to such incomplete and inaccurate information, the agent can make incorrect decisions as it negotiates with different travel agents.

A central reason for this oversight in the existing literature on automated negotiation is that most these frameworks start form the *game theoretic* view of rationality. Game theory adopts the classical economic notion of rationality, represented by the Economic Person (EP), who is a *substantively* rational maximiser of subjective expected utility. The EP is substantive because he is assumed to have sufficient information and processing power in order to compute the optimal outcome. Consequently, in a negotiation encounter, where the outcome is determined by the actions of more than one self-interested participant, agents can make optimal strategic decisions. Young [2001] argues that the traditional notion of economic rationality fails to capture the process by which agents form and change their preferences during negotiation. In Young's words:

> "...since, under the assumptions of the received concept of economic ratio-

> nality, each player's objectives or ends are not justifiable rationally, and are set by the agent's preferences in advance of reasoning, game theory has no way to capture or evaluate those ends. It can never be sure it understands the complete motivational set which is driving the individual, and in the end can only reduce the rich mix of factors which motivate and guide human beings to simple economic self-interest. Unfortunately, . . . much is lost along the way." [Young, 2001, p. 97]

Young then argues that in order to solve this problem, agents must adopt a notion of *communicative rationality* rather than merely strategic, instrumental rationality. This communicative view of rationality takes into account the internal processes 'inside the heads' of the agents making decisions. I take the position that argumentation, which allows agents to conduct a rational discussion over their and each others' underlying motivations and resulting preferences, can enable agents to form and change their preferences during negotiation, leading to an increased likelihood and quality of agreement.

In an attempt to find an inspiration for such argumentation-augmented negotiations among software agents, I turned to the literature on negotiation among humans. One of the most popular and frequently recommended alternative methods for conflict resolution is so-called *interest-based negotiation* (IBN) [Fisher et al., 1991]. Negotiation counterparts share information about their underlying interests, which enables them to identify opportunities for agreements that were not initially apparent. Moreover, interest-based negotiation enables negotiators to resolve misconceptions, leading to the refinement of their choices and evaluation criteria. As a result, acquiring and modifying preferences takes place during the process of negotiation itself. This raises the opportunity to realise similar benefits by equipping computational agents with the ability to conduct dialogues over interests during negotiation.

1.3 Aims and Contribution Scope

In light of the challenge discussed above, the research reported in this thesis began as an attempt to answer the following question:

> *How can we specify intelligent agents capable of negotiating effectively, de-*

> *spite having incomplete and/or inaccurate individual preferences over negotiation outcomes?*

This thesis reports on my explorations of this question. In particular, I contend that rational communication, through the process of argumentation, can enable agents to exchange the information necessary to refine their individual preferences. To this end, I attempt to identify the essential *primitives* and *mechanisms* needed for flexible automated negotiation. In particular, I explore the following concrete questions:

1. What are the appropriate abstractions and decision-making primitives that need to be specified in order to enable automated agents to conduct rational discussion over their preferences?
2. Can a suitable communication protocol be defined to enable rational discussion over these primitives, leading to flexible automated negotiation?
3. How would negotiation using such types of protocols compare to more basic "bargaining" where agents simply exchange offers?
4. How may software agent designers reason about the process of designing negotiation strategies for agents participating under flexible negotiation protocols?

To address the first two questions, I present a framework for interest-based negotiation among computational agents. In particular, I present an interaction protocol, expressed in the form of a formal dialogue game, to enable agents to conduct IBN dialogues. I introduce an argumentation-theoretic approach to enable agents to modify their preferences as a result of receiving arguments relating to their beliefs and planning knowledge. To answer the third question, I initiate an enquiry into the difference between bargaining and argumentation-based negotiation and I show that bargaining can fail to reach a desirable outcome due to the inaccurate and/or incomplete information agents use in deriving their preferences. Building on this understanding, I then explore various issues relating to the design of negotiation strategies for IBN and other negotiation settings such as the Trading Agent Competition. In particular, I address the fourth question by presenting a methodology for designing negotiation strategies in relatively complex domains. Finally, to provide a proof-of-concept verification of the framework, I present a pilot application for context-aware services on mobile devices. This application makes use of concepts and techniques developed in this thesis to support mobile users in coordinating their activities.

My work is confined to a problem in Artificial Intelligence (AI), specifically: the engineering of computational multi-agent systems capable of resolving conflicts over resources. Even though I draw on concepts from human negotiation, game theory, and the philosophy of argumentation, I do not –and do not claim to– make a contribution to these areas. Moreover, it must be stressed that I am not concerned with modelling the way humans negotiate, nor with building computer systems that can negotiate with humans. Thus, my work is not intended to contribute to computational linguistics or natural language processing.

1.4 Overview of the Thesis

This thesis is structured into seven chapters as follows.

Chapter 1 is where I present the research question.

Chapter 2 provides a brief critical survey of relevant existing research. In particular, I discuss and critique existing frameworks for automated negotiation in multi-agent systems.

In **Chapter 3**, I present a conceptual framework for automated interest-based negotiation in multi-agent systems. I formalise parts of this conceptual framework using argumentation-theoretic concepts. Then, I present an interaction protocol for facilitating IBN among software agents.

In **Chapter 4**, I draw some comparisons with bargaining-based frameworks and demonstrate that IBN is advantageous in certain circumstances.

In **Chapter 5**, I propose an informal methodology for designing negotiation strategies in multi-agent systems. The methodology provides guidance for strategy design in frameworks where game-theoretic techniques are not applicable. I demonstrate how the methodology can be used to construct an exemplar strategy for IBN. I also verify the methodology by dissecting an existing strategy for bidding in heterogeneous simultaneous auctions.

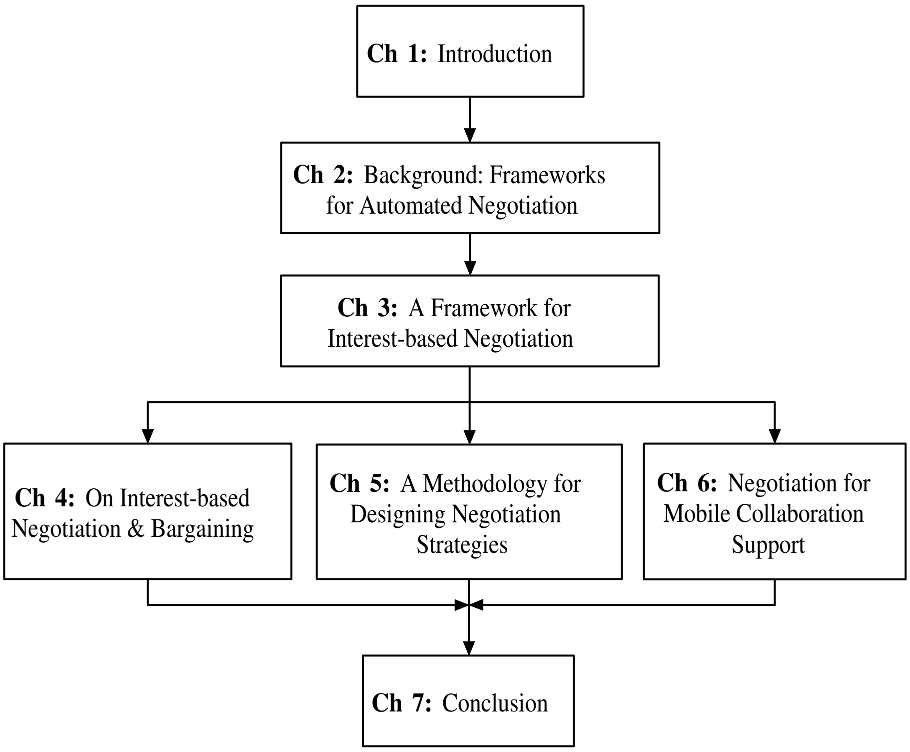


Figure 1.3: Structure of the thesis

In **Chapter 6**, I demonstrate a pilot application that uses IBN in the domain of cooperation support of mobile users.

Finally, in **Chapter 7**, I return to the four aspects identified in section 1.3 and summarise the progress achieved in the body of the thesis.

The structure of the thesis is summarised in Figure 1.3. Chapters 4, 5 and 6 are all elaborations on the conceptual framework presented in Chapter 3.

Chapter 2

Background: Frameworks for Automated Negotiation

"Knowledge is the conformity of the object and the intellect."

Ibn Rushd (Averroes), Arab Andalusian Philosopher (520 – 595)

In this chapter, I discuss and critique existing frameworks for automated negotiation in multi-agent systems.

2.1 Introduction

Having defined my research question in the previous chapter, I now present a brief critical survey of relevant prior research. In particular, I discuss and critique existing frameworks for automated negotiation in multi-agent systems.

Recall the definition of negotiation from the previous chapter:[1]

> *Negotiation is a form of interaction in which a group of agents with conflicting interests and a desire to cooperate, try to come to a mutually acceptable agreement on the division of scarce resources.*

Automated negotiation among autonomous agents is needed when agents have conflicting objectives and a desire to cooperate. This typically occurs when agents have competing

---

[1] The precise definition of negotiation is not always explicitly stated in the literature. However, I believe that this definition is a reasonable generalisation of both the explicit and implicit definitions that can be found.

claims on scarce resources, not all of which can be simultaneously satisfied. The use of the word “resources” here is to be taken in the broadest possible sense. Thus, resources can be commodities, services, time, money etc. In short, anything that is needed to achieve some objective.

In the multi-agent literature, various interaction and decision mechanisms for automated negotiation have been proposed and studied. These include: game-theoretic analysis [Kraus, 2001, Rosenschein and Zlotkin, 1994, Sandholm, 2002a]; heuristic-based approaches [Faratin, 2000, Fatima et al., 2002, Kowalczyk and Bui, 2001]; and argumentation-based approaches [Kraus et al., 1998, Parsons et al., 1998, Sierra et al., 1998]. Even though there may be many ways to classify existing approaches to automated negotiation, the classification adopted in this thesis, due to Jennings et al. [2001], suits the purpose of my discussion.

This thesis falls in the category of argumentation-based approaches. The main distinguishing feature of such approaches is that they allow for richer forms of interaction than their game-theoretic and heuristic counterparts. In the course of discussion, I identify some major unaddressed challenges and opportunities.

This chapter is organised as follows. In the next section I present various concepts essential to understanding the automated negotiation problem. I then discuss and critique game-theoretic approaches to studying automated negotiation in Section 2.3, followed by a discussion of heuristic-based approaches in Section 2.4. In Section 2.5, I give a conceptual outline of the idea of argumentation-based negotiation. I then present a detailed survey of argumentation-based approaches to negotiation in Section 2.6. The chapter concludes with a summary and discussion in Section 2.7.

2.2 The Negotiation Problem

Before I discuss different frameworks for studying automated negotiation, it is essential to define the negotiation problem more precisely. The goal of this section is to specify some fundamental concepts used in the automated negotiation literature.

2.2.1 The Negotiation Agreement Space

Recall that negotiation aims at reaching some *allocation* of resources that is acceptable to all parties. Since there are usually many different possible allocations (or possible *agreements*, or *deals*, or *outcomes*), negotiation can be seen as a "distributed search through a space of potential agreements" [Jennings et al., 2001]. Hence, we first require some way of characterising such a space. Abstractly, the space can be seen as a set of deals $\Psi = \{\Omega_1, \ldots, \Omega_n\}$ where n is the size of the search space.

Another way of characterising the search space is in terms of vectors in a multi-dimensional Euclidean space. In this characterisation, a deal is defined in terms of a set of attributes $A_1, \ldots, A_n$ (dimensions in an n dimensional vector space), where each attribute A_i can take a set of values $a_1^i, \ldots, a_m^i$. This is illustrated by the following example:

Example 1. *Suppose a buyer and seller are negotiating over the purchase of a car, where attributes are 'make' and 'price'. Suppose that make can be either Volv or Toyot, and price can be any integer value between one and $20,000. Hence, every possible combination of make/price values is a deal, such as* $(\textit{Toyot}, \$10,000)$ *or* $(\textit{Volv}, \$15,000)$. *In this case, the size of the space of possible deals is* $2 \times 20,000 = 40,000$*, since for each make, we can assign any number between* 1 *and* $20,000$.

More generally, if we have a set of attributes $A_1, \ldots, A_n$, each ranging over a domain of size $s(A_i)$, then the size of the space of possible deals is $\prod_{i=1}^{n} s(A_i)$. The space grows significantly as the number of attributes and the number of possible attribute values increases. In complex domains, where the space of possible deals is very large, it is infeasible to consider every possible allocation. This becomes of particular importance when there are time constraints. In such cases, it may be useful to use *heuristics* that enable agents to find deals more efficiently. I will discuss heuristic approaches in more detail in section 2.4.

Note that in example 1 above, the deal $(\textit{Toyot}, \$10,000)$ is to be implicitly understood as: the Toyot car is to be given (or allocated) to the buyer by the seller, while the $10,000 is to be given to the seller by the buyer. This was possible because our domain knowledge rules out the ambiguity. Sometimes, however, a deal may require explicit specification of the agent for which a resource is allocated, particularly when such agents are themselves part of the choice of allocation.

Example 2. *A port authority allocating different shipment jobs to privately owned ships needs to specify what ship each job goes to, as well as the price agreed upon. Suppose we have three jobs: delivery1, delivery2, and delivery3 and two ships: ship1 and ship2. Then a deal can be specified in terms of pairs of ships and sets of delivery jobs. For example, the following may represent a deal where ship1 is allocated the first and second delivery jobs, while ship2 is allocated the third:*

$$((\textit{ship1} : \{\textit{delivery1}, \textit{delivery2}\}), (\textit{ship2} : \{\textit{delivery3}\}))$$

Note that when we specify the agents as well as resources, the number of possible deals increases again, since we now need to consider each combination of agent/resource pairs explicitly.

In summary, to specify a negotiation framework, we need to somehow characterise the set of possible allocations. The precise way in which this is carried out is domain-dependent.

2.2.2 Criteria for Evaluating Deals

In this section, I discuss criteria for evaluating different deals. I begin by discussing how a single agent evaluates deals based on its own preferences. Then I discuss evaluation from the global point of view.

Individual Evaluation

The very nature of competition over resources means that different agents prefer different allocations of the resources in question. Hence, we need to capture the *individual agent preference* over the set Ψ of possible deals. Preferences of agent i can be captured using a binary *preference relation* $\succeq_i$ over Ψ, and we denote by $\Omega_1 \succeq_i \Omega_2$ that for agent i, allocation Ω_1 is at least as good as allocation Ω_2. It is also common to use $\Omega_1 \succ_i \Omega_2$ to denote that $\Omega_1 \succeq_i \Omega_2$ and it is not the case that $\Omega_2 \succeq_i \Omega_1$ (this is called *strict* preference). Economists and decision-theorists consider a preference relation to be *rational* if it is both transitive and complete [Mas-Colell et al., 1995].[2]

For convenience, the preference relation of agent i is often described in terms of a *utility function* $U_i : \Psi \rightarrow \mathbb{R}+$, which assigns a real number to each possible deal. The utility function $U_i(.)$ represents the relation $\succeq_i$ if we have $U_i(\Omega_1) \geq U_i(\Omega_2)$ if and only if $\Omega_1 \succeq_i \Omega_2$.

In a sense, the utility function (and corresponding preference relation) captures the *level of satisfaction* of an agent with a particular deal. A *rational*[3] agent attempts to reach a deal that *maximises the utility* it receives. If an allocation that maximises the utilities of all agents is possible, then by finding this allocation, all agents are maximally satisfied. If, however, such a deal is not possible, then each agent will attempt to reach an allocation which gives it as much utility as possible.

It is worth noting that agents may already have a particular allocation of resources before they begin negotiation. Negotiation, in this case, is an attempt to *reallocate* the resources in order to reach a new allocation that is more preferable to both. This is clarified in the following example.

Example 3. *A customer attempting to purchase a car may already have \$15,000 in his pocket. Similarly, a car dealer already has a number of cars in her showroom. Suppose the initial allocation is represented by* $\Omega_1 = ((\textit{customer} : \{\textit{\$15K}\}), (\textit{dealer} : \{\textit{Volv}\}))$*, while one potential reallocation is* $\Omega_2 = ((\textit{customer} : \{\textit{Volv}\}), (\textit{dealer} : \{\textit{\$15K}\}))$*. If the customer prefers owning a Volv car to having \$15,000 in his pocket (i.e. if* $\Omega_1 \succeq_{\textit{customer}} \Omega_2$*), then it is rational for the customer to accept the new allocation* Ω_2*. If the car dealer also prefers to have \$15,000 in her safe rather than have the Volv in the showroom (i.e. if* $\Omega_1 \succeq_{\textit{dealer}} \Omega_2$*), then it is also rational for the dealer to accept* Ω_2*.*

A *conflict deal*, call it $\Omega_{\textit{conflict}}$, refers to the situation where agents do not reach an agreement in negotiation. In the above example, $\Omega_{\textit{conflict}} = \Omega_1$ because Ω_1 was the initial (default) allocation of resources.

---

[2] A binary $\succeq$ relation is *transitive* if whenever $x \succeq y$ and $y \succeq z$, we also have $x \succeq z$; and it is *complete* if for all x and y, either $x \succeq y$ or $y \succeq x$ or both hold.

[3] Here, I refer to *rationality* in the economic-theoretic sense: acting to maximise (expected) subjective utility.

Global Evaluation

So far, I discussed how a particular agent evaluates and compares deals. I now consider the *global* evaluation of deals. The first concept is that of *dominance*. A deal Ω_1 *weakly dominates* another deal Ω_2 if for all agents involved, we have $U_i(\Omega_1) \geq U_i(\Omega_2)$. A deal Ω_1 *strictly dominates* (or simply *dominates*) Ω_2 if Ω_1 weakly dominates Ω_2 and for at least one agent k, we have $U_i(\Omega_1) > U_i(\Omega_2)$.

Definition 1. (Individual Rational Allocation) *A deal (or allocation) is* individual rational *if it weakly dominates the conflict deal* $\Omega_{conflict}$.

In other words, a deal is individual rational if all agents are at least as well off with the deal as they were with the initial allocation.

Another useful concept is that of Pareto optimality, due to Italian economist Vilfredo Pareto [Pareto, 1906].

Definition 2. (Pareto Optimal Allocation) *A deal (or allocation) is* Pareto optimal *(or* Pareto efficient*), if it is not dominated by any other deal.*

In other words, a deal is Pareto optimal if there is no other deal that makes one agent better off without making another agent worse of. This concept is important because if agents reach a non-Pareto optimal outcome, then they are able to improve that outcome without incurring any trade-offs.

Finally, the interested reader is referred to *welfare economics* [Mas-Colell et al., 1995], a field of enquiry concerned with evaluating social choice from the point of view of the *policy maker* (i.e. the mechanism designer).

2.2.3 Negotiation Mechanism

Given a set of agents, a set of resources, an existing resource allocation and a set of other possible allocations, the main goal of negotiation is to find an allocation that is *better* in some sense, if such allocation exists. In order to achieve this goal, agents need some *mechanism*. Abstractly, a mechanism specifies the *rules of encounter*: what agents are allowed to say and when, how allocations are calculated, whether calculations are done using a centralised algorithm or in a distributed fashion, and so on.

A typical example of a mechanism is an *English auction*. In this resource allocation mechanism, the players include a seller and a number of potential buyers. The seller has a single resource (e.g. a house), and each potential buyer is endowed with money and a particular *valuation* of the seller's resource. Potential buyers make successive *bids*, each proposing to purchase the resource for a specific price. The mechanism requires that no bid could offer an amount of money lower than a previous bid, hence it is an "ascending bid protocol." The highest bidder wins.

Another example mechanism is *bargaining*, where two agents exchange offers (i.e. suggestions about how to exchange resources) until one agent makes an offer that is acceptable by the other.

Different mechanisms may have different properties. Following is a list of desirable features for negotiation mechanisms, which I presented briefly in Chapter 1.[4]

1. *Simplicity:* A mechanism that requires less computational processing and communication overhead is preferable to one that does not.

2. *Efficiency:* A mechanism is efficient if it produces a good outcome. What is meant by "good," however, may differ from one domain to another. One common criteria is that of Pareto Optimality, where no agent could be better off in a different allocation without some other agent being worse of. Another criteria might be Global Optimality, where the sum of agents benefits is maximised.

3. *Distribution:* It is preferable to have a negotiation mechanism that does not involve a central decision-maker. Centralisation may lead to communication bottlenecks or decreased reliability due to the single-point-of-failure.

4. *Symmetry:* This property implies that the mechanism should not be biased for or against some agent based on inappropriate criteria. Again, what constitutes an 'inappropriate' criteria depends on the domain in question.

5. *Stability:* A mechanism is stable if no agent has incentive to deviate from some agreed-upon strategy or set of strategies. In an auction, for example, we may require that no agent lie by making a false bid, or that no group of agents can form strategic coalitions to create a disadvantage for other agents.

---

[4] This list is adapted from Rosenschein and Zlotkin [1994].

6. *Flexibility:* By this property, I mean that the mechanism should lead to agreement even if agents did not have complete and correct private information in relation to their own decisions and preferences.[5] This property, which is the focus of this thesis, requires a complementary mechanism for rational investigation and possible refinements of internal decisions during negotiation itself.

A variety of automated negotiation mechanisms have been studied in the literature. Different mechanisms have different advantages and disadvantages. Next, I discuss the major families of such mechanisms.

2.3 Game-theoretic Approaches to Negotiation

In this section, I discuss approaches to automated negotiation based on (non-cooperative) game-theory. I first introduce some concepts from game-theory; then I use these concepts to describe and critique existing automated negotiation approaches that make use of these concepts.

2.3.1 An Overview of Non-Cooperative Game Theory

Game theory [Osborne and Rubinstein, 1994] is a branch of economics in which strategic interactions between self-interested economic[6] agents are studied. It has its roots in the work of von Neuman and Morgenstern [1944]. Recently, it has been used extensively to study and engineer interaction between self-interested computational agents [Rosenschein and Zlotkin, 1994, Sandholm, 2002a].

Game theory provides tools for conducting two types of analysis:

– analysing optimal behaviour (or strategies) of individuals or organisations *given* some underlying resource allocation mechanism;

– analysing how one may design optimal mechanisms, *given* that agents behave strategically.

---

[5] This should not be confused with the case of *incomplete information* in game-theory, where an agent is uncertain about its counterpart's preferences.

[6] I refer to these as "economic" agents because economics is concerned with the interaction among people, organisations etc., rather than solely among computational agents.

I will discuss the main ideas behind these two types of analysis in the next two subsections.

Concepts for Analysing Strategic Behaviour

A *game* is a form of "strategic encounter" between a number of *players* (or agents). Each player has a set of alternative *actions* (or *choices*, or *strategies*) available to it. The rules of the game (or the *mechanism*) specify (i) the set of actions available; and (ii) the *outcome* of the encounter, based on the choices made by different players.

The most popular introductory example to game theory is the so-called *prisoner's dilemma* [Poundstone, 1993], where the players are two criminals held by the police and interrogated in separate rooms. The players cannot communicate, and each must make an individual choice that influences both of them. In particular, each player must choose whether to *confess* or *not confess*. If neither suspect confesses, they go free, and split the proceeds of their crime, receiving utility of, say, 5 each. However, if one prisoner confesses and the other does not, the prisoner who confesses testifies against the other in exchange for going free and gets the entire loot, hence getting utility 10, while the prisoner who did not confess goes to prison and gets nothing (utility zero). If both prisoners confess, then both are given a reduced term, but both are convicted, which we represent by giving each 3 units of utility: better than having the other prisoner confess, but not as good as going free.

Table 2.1 shows a matrix representation of the game, its actions and outcomes. The first row shows actions available to player 1, while the first column shows actions available to player 2. The numbers in the upper right hand of each cell represent the utility (or *payoff*) received by player 1 from that action combination,[7] while the bottom left number represents the utility of player 2. Note that higher numbers are better (more utility).

The following analysis helps the reader become familiar with the game-theoretic techniques of analysing strategies. Assume that player 1 knows the set of actions available to him and to player 2, and that player 1 also has complete information about the payoffs in the matrix. Player 1 reasons strategically as follows: suppose player 2 does not confess! In this case, I would rather confess, because I would get a utility of 10 (compared to 5

---

[7] In game-theoretic terms, this action combination is called a *strategy profile*.

	not confess	**confess**
not confess	5 5	10 0
confess	0 10	3 3

Table 2.1: The Prisoner's Dilemma

for not confessing). Suppose, instead, that player 2 confesses. In this case, I would also rather confess, because I get a utility of 3 (compared to 0 for not confessing). Hence, for every possible action of player 2, player 1 is better off confessing. For player 1, confession is the *dominant strategy*.[8] The exact same analysis can be taken from the point of view of player 2, leading to a dominant strategy to confess. As a result of both agents confessing, they will get a payoff of 3 each. Note that in this case, both agents are worse off than they would be if they both did not confess (in which case they would receive a utility of 5 each). In other words, even though the outcome resulting from mutual non-confession strictly dominates the outcome resulting from mutual confession, "rational" agent behaviour will lead to the latter.

A central concept in game theory is that of *equilibrium*. A notable type of equilibrium is the so-called *Nash equilibrium*, where no player has an incentive to deviate from a particular strategy, given that other players stick to their strategies. In the prisoner's dilemma, the mutual confession strategies are in Nash equilibrium.

Concepts for Mechanism Design

Mechanism design is concerned with the design of the resource allocation mechanism,[9] given that each agent will behave in such a way as to maximise its (expected) utility. Mechanism designers analyse strategic behaviour, using notions of dominant strategies, equilibria etc., in order to produce a mechanism that has certain properties.

An example of desired mechanism properties is "incentive compatibility." A mechanism is said to be *incentive compatible* if, under that mechanism, the dominant strategy for all agents is to tell the truth about their preferences (often referred to as their *types*).

---

[8] Note that this notion of dominance among strategies is different from the dominance among deals discussed earlier.

[9] In fact, mechanism design is concerned with the more general problem of producing collective (social) choice based on revealed preferences of individuals [Mas-Colell et al., 1995].

This is a powerful concept, since by guaranteeing incentive compatibility, mechanism designers make sure agents cannot strategically manipulate the outcome by lying about their types. This property is an example of the *stability* requirement mentioned earlier in section 2.2.3. In economics, mechanism design principles are used to design various negotiation mechanisms, ranging from auctions, to voting, to bilateral bargaining [Mas-Colell et al., 1995].

2.3.2 Game Theory for Automated Negotiation

Game theory offers a very powerful tool for studying and engineering strategic interaction among self-interested computational agents in general, and to automated negotiation in particular [Binmore and Vulkan, 1997]. Recall that game theory can be applied to study and engineer both the *strategies* as well as the *mechanism*.[10] The field of *computational mechanism design* [Dash et al., 2003] uses mechanism design techniques in order to construct mechanisms for allocating resources in multi-agent systems. In what follows, I survey some of the most influential uses of game theory in studying automated negotiation.

A Domain Theory for Automated Negotiation

Perhaps the most seminal work on the use of mechanism design to study automated negotiation is by Rosenschein and Zlotkin [1994]. The authors present a *domain theory* for automated negotiation, distinguishing between three different domains:

1. *Task-oriented domains*: domains involving the division of tasks to execute; the utility function measured in terms of the costs associated with different task allocations; each agent tries to minimise the cost of the tasks it has to execute.

2. *State-oriented domains*: domains involving a joint decision about what state agents will achieve; the utility function is measured in terms of preference over states that result from different deals; each agent tries to get to a more preferable state to itself.

3. *Worth-oriented domains*: domains involving a joint decision about what goals to

---

[10] Note that a 'game' in game theory does not necessarily have to be a negotiation game. Game theory is concerned with strategic multi-party decision-making in general. Hence, game theory can be used to study interaction that do not involve resource allocation, such as strategic decisions in war time and in law cases.

achieve; the utility function is measured in terms of the number of goals each deal achieves; each agent tries to achieve as many goals as possible.

The authors use concepts from game theory and mechanism design theory in order to study agent strategies in different domains, and under different mechanisms. The authors are particularly interested in designing (centralised) mechanisms that produce an outcome (e.g. an allocation of tasks, or a state to reach) based on the information agents reveal about themselves (e.g. the set of tasks they each have to execute, or their preferences over states). The authors show that, in certain situations, an agent can benefit from *strategic manipulation*, for example by lying about the tasks it has to perform, or about its preferences over states. This analysis was then used in order to design incentive compatible mechanisms, i.e. mechanisms that *force* agents to be truthful. However, such mechanisms are restricted by certain conditions. For example, the authors were able to construct incentive compatible mechanisms when agents have incomplete information about each other's preferences over states, but not when they have incomplete information about each other's goals.

Mechanisms for Combinatorial Auctions

Sandholm [2002a] used game-theoretic techniques in order to construct *eMediator*, an electronic commerce server that uses algorithmic and game-theoretic techniques to allocate resources among multiple agents. *eMediator* includes the *eAuctionHouse*, a configurable auction server that can handle a number of combinatorial auctions and exchanges; and the *eCommitter*, a contract optimiser that determines the optimal contract price and decommitting penalties for the different parties, taking into account that agents may decommit strategically. The author is concerned with achieving optimal outcomes using a mechanism that ensures agents do not deviate from the desired strategies. In related work, Sandholm [2002b] presents an algorithm for optimal winner determination in combinatorial auctions (auctions where bidders can bid on combinations of items). Conitzer and Sandholm [2002] explore viewing the mechanism design problem itself as a computational problem, and present algorithms that produce preference aggregation mechanisms at run-time, given a particular setting.

2.3.3 Limitations of Game Theory

An adequate evaluation of game theory is beyond the scope of this thesis. Therefore, I focus my discussion on issues relevant to automated negotiation, and particularly to the topic of this thesis.

In game-theoretic analysis, researchers usually attempt to determine the optimal strategy by analyzing the interaction as a game between identical participants, and seeking its equilibrium [Harsanyi, 1956, Rosenschein and Zlotkin, 1994, von Stengel, 2002]. The strategy determined by these methods can sometimes be made to be optimal for a participant, given the game rules, the assumed payoffs, and the goals of the participants, and assuming that the participants have no knowledge of one another not provided by introspection. Assuming further that participants behave according to the assumptions of rational-choice theory [Coleman, 1990], then this approach can guide the design of the interaction mechanism itself, and thus force such agents to behave in certain ways [Varian, 1995].

Classical game theory assumes, among other things, that agents:

1. have unbounded computational resources,
2. have complete knowledge of the outcome space, and
3. are optimisers of utility in the sense of rational-choice theory,

From a computational perspective, these assumptions imply unrealistic assumptions about the negotiating software agents.

The first assumption implies that no computation or communication cost is incurred in order to reach a deal. In most realistic computational environments, however, this assumption fails due to the limited processing and communication capabilities of information systems. The size of the set of possible deals grows exponentially with the number of attributes and attribute values. Calculating and evaluating all of these may require more time and computation than can be afforded. Similarly, in a bargaining encounter, exchanging every possible offer may be impractical, given time and communication bandwidth limitations. Classical game-theoretic models do not provide a way to account for these costs and study their impact on strategic decisions.

The second assumption implies that not only does the software agent have unbounded computational resources to *evaluate* every possible resource allocation, but it also has all preference information needed to perform such evaluation. In many domains, however, it may be impractical for the user to specify its complete preference information to the agent.[11]

The third assumption implies that agents always make decisions that optimise their utility. Game theory requires this because an agent must first reason about the "optimal" strategy of the opponent before deducing the best response to that strategy. However, software agents may be resource-constrained (as discussed above), altruistic, malicious, or simply badly-coded, so that participant behaviour may not conform to the assumptions of rational choice theory. Hence, if game theory's *predictions* are inaccurate, its *prescriptive* advice becomes unreliable.

Game theory can also be critiqued from a "software-engineering" point of view. Game theory is *normative* since it is concerned with *what* constitutes an optimal decision given a game description. Hence, classical game theory has nothing to say about *how* to implement agents that reason optimally.

It is worth pointing out that an emerging sub-area of game theory, termed *evolutionary game theory* [Samuelson, 1998], is concerned with some of the limitations discussed above. Evolutionary game theory relaxes the assumption of unbounded rationality. Instead of calculating optimal strategies, games are played repeatedly and strategies are tested through a trial-and-error learning process in which players gradually discover that some strategies work better than others. However, other assumptions, such as the availability of a preference valuation function, still hold. Another attempt is the modelling of "bounded rationality" by explicitly capturing elements of the process of choice, such as limited memory, limited knowledge, approximate preferences (that ignore minor difference between options) etc. [Rubinstein, 1997]. These frameworks are primarily aimed at producing models that better explain and predict human behaviour in real economic and social scenarios. Their insight into the building of multi-agent systems requires further exploration and is relevant to heuristic approaches discussed in the next section.

---

[11] Elicitation of user preferences for automated negotiation has been addressed in recent work by Luo et al. [2003a].

2.4 Heuristic-based Approaches to Negotiation

When agent designers relax some of the assumptions of game theory, particularly regarding unbounded rationality, they immediately fall outside the region of predictability of classical game-theory. This implies that analytical results (e.g. about optimal strategies) become hard to achieve. Instead, approximate strategies (or heuristics) must be devised. Heuristics are rules of thumb that produce good enough (rather than optimal) outcomes. For heuristic approaches, experimentation through simulation becomes a more viable option for studying the properties of different strategies. The support for a particular heuristic is usually based on empirical testing and evaluation in comparison with other heuristics [e.g., Faratin, 2000, Kraus, 2001]. In general, these methods offer approximations to the decisions made according to game-theoretic studies. The heuristic approach has been applied both to bargaining mechanisms as well as auction-based mechanisms. In the next section, I survey some major frameworks in each category.

2.4.1 Heuristics for Automated Negotiation

Heuristics for Bargaining

A number of heuristic methods have been employed in a "service-oriented" negotiation framework presented by Faratin, Sierra and Jennings in a number of papers [see Faratin, 2000, Sierra et al., 1997]. In this framework, different heuristic decision functions are used for evaluating and generating offers in multi-attribute negotiation [Faratin et al., 1998]. Instead of exploring all possible deals, agents exchange offers based on heuristic functions that depend on time deadlines and resource availability. Moreover, in order to improve the convergence to a deal, the authors present a method that enables an agent to generate offers that are "similar" to previous offers made by its negotiation counterpart [Faratin et al., 2002] (where, "similarity" representation is based on fuzzy-logic techniques [Zadeh, 1971]). The intuition is that such offers are more likely to be accepted by the counterpart.

Kowalczyk and Bui [2001] present a negotiation model with decision procedures based on distributed constraint satisfaction [Yokoo, 1998]. This enables agents to use heuristics used in the constraint satisfaction literature in order to improve the process of generating and evaluating offers. This framework was later extended to allow for multiple concurrent negotiations [Rahwan et al., 2002] and to accommodate fuzzy (as opposed to "crisp") constraints [Kowalczyk, 2000]. The idea of using fuzzy constraint satisfaction is further investigated by Luo et al. [2003b].

Kraus [2001] presents a negotiation framework based on Rubinstein's model for alternating offers [Rubinstein, 1982]. The framework has been used to solve data-allocation, resource-allocation and task-distribution problems, and was verified via empirical simulation and (to a certain extent) related analytically to game-theoretic concepts. In related work, Fatima et al. [2004a, 2002, 2001] studied the influence of information and time constraints on the negotiation equilibrium in a particular heuristic model.

The Trading Agent Competition

Another example of the use of heuristics in negotiation is the Trading Agent Competition[12] (TAC): an annual competition, which involves multiple competing agents bidding in simultaneous auctions. I discuss TAC–02 as an example.

Eight agents participated in each TAC–02 game. Each agent performed the role of a travel agent attempting to provide booking for eight clients travelling from TACtown to Tampa and back during a five-day period. Each client was characterised by a random set of preferences for arrival and departure dates, hotels and entertainment tickets. Utility was gained by purchasing a complete package and was calculated based on comparison with the corresponding client's preferences. Package constituents were sold in separate simultaneous auctions, each with certain price dynamics. Airline tickets were sold in single round continuous auctions with biased random pricing that was more likely to increase. Hotel bookings were sold in ascending English auctions clearing every minute, while entertainment tickets were traded in continuous double auctions. The score of an agent was the difference between the total utility gained for its clients and the agent's expenditure.

TAC represents a real challenge for automated negotiation, where game-theoretic techniques fail. This is mainly due to the complexity of the problem and the time limitations. Agents participate in 28 different auctions over a period of 12 minutes. Each agent has to solve a combinatorial assignment problem, where goods must be packaged into

---

[12] An overview of TAC has been presented by Greenwald [2003].

bundles. Moreover, agents' bidding behaviour must be strategic, taking into account the strategies of other agents in order to decide when to buy and how much to bid. A consequence of these complications is that 'there is no known way to compute the best course of action' [Eriksson and Janson, 2002]. TAC–02 participants used techniques ranging from Linear Programming for finding optimal bundles, to Machine Learning for modelling other agents' behaviours, to Genetic Algorithms for evolving adaptive strategies.

2.4.2 Limitations of Heuristic Approaches

Heuristic methods do indeed overcome many of the shortcomings of game-theoretic approaches. However, they also have a number of disadvantages [Jennings et al., 2001]. Firstly, the models often lead to outcomes that are sub-optimal because they adopt an approximate notion of rationality and because they do not examine the full space of possible outcomes. And secondly, it is very difficult to predict precisely how the system and the constituent agents will behave. Consequently, the models need extensive evaluation through simulations and empirical analysis.

Another limitation of heuristic approaches is that, like game-theoretic approaches, they assume that agents *know what they want*. In other words, agents have a precise and correct way of calculating the quality of the negotiation outcome (usually using numerical utility functions). As I shall argue in depth in the following section, this requirement cannot always be satisfied, in which case alternative techniques would be needed.

2.5 Argumentation-based Approaches to Negotiation

Recall from section 2.2.3 that one of the desirable properties of a negotiation mechanism is *flexibility*: the ability to lead to agreement even if agents did not have complete and correct private information in relation to their own decisions and preferences. In the next subsection, I will argue that existing game-theoretic and heuristic approaches do not satisfy this property. Then, I will show how an emerging family of negotiation frameworks, based on the notion of *argumentation*, has the potential to overcome this limitation. Such frameworks have been termed *argumentation-based negotiation* (ABN) frameworks.[13]

---

[13] The review presented in this section is a summarised version of a more comprehensive survey [Rahwan et al., 2003b].

2.5.1 A Closer Look at Game-Theoretic and Heuristic Models

Game-theoretic and heuristic approaches to automated negotiation mostly assume that agents' utilities or preferences are completely characterised prior to the interaction. Thus an agent is assumed to have a mechanism by which it can assess and compare any two proposals. This may be easy, for example, when the utility of the negotiation object is defined in terms of a monetary value, such as the charging rate of a telephone call. An agent can compare proposals of two telephone service providers by simply comparing how much they charge per minute. However, in more complex negotiation situations, such as trade union negotiations, agents may well have incomplete information which limits this capability. Thus, agents might:

- lack some of the information relevant to making a comparison between two potential outcomes,
- have limited resources preventing them from acquiring such information,
- have the information, but lack the time needed to process it in order to make the comparison,
- have inconsistent or uncertain beliefs about the environment,
- have unformed or undetermined preferences (e.g. about products new to them), or
- have incoherent preferences.

A wealth of the situations described above exist in the human negotiation world. For example, consumers form their preferences based on information available to them. They acquire and modify their preferences as a result of interaction with the environment and other consumers [Lilien et al., 1992]. Advertising capitalises on this idea, and can be seen a process of 'argumentation' in which marketers attempt to persuade consumers to change their preferences among different products [Slade, 2002].

Information relevant to making preferential decisions in negotiation may also be lacking in a multi-agent software system. For example, the user may not have specified a complete set of preferences to the agent acting on his/her behalf, or may have specified incorrect or incoherent preferences to the agent.[14] Another example is that of a trading agent that has uncertain beliefs about the market in which it is trading.

Among humans, the processes of acquiring information, resolving uncertainties and revising preferences often take place as part of the negotiation process itself. A reasonable question to ask is: *can computational agents benefit from acquiring and revising their preferences during negotiation?* For game-theoretic and heuristic frameworks, the answer is mostly “No” for the following reasons:

– In most game-theoretic and heuristic models, agents exchange *proposals* (i.e. potential agreements or potential deals). This, for example, can be a promise to purchase a good at a specified price in an English auction, a value assignment to multiple attributes in a multi-dimensional auction [Wurman, 1999], or an alternate offer in a bargaining encounter [Larson and Sandholm, 2002]. Agents are not allowed to exchange any additional information other than what is expressed in the proposal itself.

– Agents’ preferences over proposals are assumed to be *proper* in the sense that they reflect the true benefit the agent receives from satisfying these preferences. For example, an agent attempting to purchase a car might assign a high value to a particular brand based on a false belief that this brand makes safer cars than other brands. In this case, the preferences do not properly reflect the agent’s actual gain if it was to purchase that car.

– Game-theoretic and heuristic approaches assume that agents’ utilities or preferences are *fixed*. A rational agent would only modify its preferences upon receipt of new information, and traditional automated negotiation mechanisms do not facilitate the exchange of such information.

Against this background, argumentation-based approaches to negotiation attempt to overcome the above limitations by allowing agents to exchange additional information, or to “argue” about their beliefs and other mental attitudes during the negotiation process.

---

[14] This is exemplified in the trip organisation scenario I presented in Chapter 1.

2.5.2 Arguments in Negotiation

In the context of negotiation, I view an *argument* as a piece of information that may allow an agent to [Jennings et al., 1998b]:

1. *justify* its negotiation stance; or
2. *influence* another agent's negotiation stance.[15]

Justifications allow agents to give each other useful "hints" during negotiation. Consider the following example dialogue:

BUYER: *I would like to rent a car for 4 days please.*
SELLER: *I can offer you a 2003 Volv model for $400.*
BUYER: *I've got a limited budget of $200. What's the best you can do?*

In this case, instead of providing a mere rejection of the seller's proposal, the buyer also provides a *justification for a rejection*, or a *critique* of the proposal. The seller should now avoid proposing any car that costs more than $200.

Arguments can also allow an agent to *influence* another agent's preferences by providing the latter with new information. This may make it possible to change the other agent's preferences, or its perception of the negotiation space itself. The following dialogue exemplifies a case of preference change:

BUYER: *I would like to rent a car for 4 days please.*
SELLER: *I can offer you a 2003 Volv model for $400.*
BUYER: *That is too much!*
SELLER: *But you will not find anything cheaper in the market, because it's a holiday season!*

2.5.3 Elements of ABN Mechanisms

While resource allocation mechanisms have a precise meaning in game-theory and economics [Varian, 1995], there is no agreed upon approach to characterising ABN mechanisms. However, it is instructive to develop such a framework so that the essential com-

---

[15] Though useful for my purpose here, this definition of an argument is contentious. A monotonic agent would never change its stance no matter what arguments were presented.

ponents that are needed to conduct automated negotiation, and consequently their associated challenges, can be clearly identified. In this section, I outline those elements that I consider essential in the design of an ABN framework. This helps give structure to my discussion of existing ABN frameworks, and enables me to identify the ABN landscape and some major open research questions in the field. As a side objective, I also hope to enable the reader to appreciate the nontrivial challenges involved in constructing ABN frameworks.

Communication Language and Domain Language

Negotiation is, by definition, a form of interaction between agents. Therefore, a negotiation framework requires a language that facilitates such communication [Labrou et al., 1999]. Elements of the *communication language* are usually referred to as *locutions* or *utterances* or *speech acts* [Searle, 1969, Traum, 1999]. Game-theoretic or heuristic-based negotiation mechanisms normally include locutions such as `propose` for making proposals, `accept` for accepting proposals, and `reject` for rejecting proposals.

In ABN frameworks, agents need richer communication and domain languages to be able to exchange meta-level information (i.e., information other than that describing deals, proposals, rejections etc.). Table 2.2 shows the main distinguishing features between ABN and non-ABN frameworks with respect to the communication and domain languages.

	Non-ABN Frameworks	**ABN Frameworks**
Domain Language	Expresses proposals only (e.g. by describing products available for sale).	Expresses proposals as well as meta-information about the world, agent's beliefs, preferences, goals etc.
Communication Language	Locutions allow agents to pass call for bids, proposals, acceptance and rejection etc.	In addition, locutions allow agents to pass meta-information either separately or in conjunction with other locutions.

Table 2.2: ABN vs. Non-ABN w.r.t Domain and Communication Languages

In multi-agent systems, two major proposals for agent communication languages have been advanced, namely the Knowledge Query and Manipulation Language (KQML)

[Mayfield et al., 1996] and the Foundation for Intelligent Physical Agents' Agent Communication Language (FIPA ACL) [FIPA, 2001]. FIPA ACL, for example, offers 22 locutions. The contents of the messages can be in any domain language. The locution $\texttt{inform}(a, b, \varphi, L)$, for example, allows agent a to inform another agent b of statement φ which is in language L. Other locutions exist allowing agents to express proposals for action, acceptance and rejection of proposals, make various queries about time and place, and so on. FIPA ACL has been given semantics in the form of pre- and post-conditions of each locution. This semantics is based on *speech act theory*, due to a philosopher of language, John Austin [Austin, 1962] and his student John Searle [Searle, 1969], in which a locution is seen as an action that affects the world in some way.

While FIPA ACL offers the benefits of being a more or less standard agent communication language, it fails to capture all utterances needed in a negotiation interaction. For example, FIPA ACL does not have locutions expressing the desire to enter or leave a negotiation interaction, to provide an explicit critique to a proposal or to request an argument for a claim. While such locutions may be constructed by injecting particular domain language statements within FIPA ACL locutions, the semantics of these statements fall outside the boundaries of the communication language. Consider the following locution from the framework presented by Kraus et al. [1998]:

$$\texttt{Request}(j, i, \textit{Do}(i, \alpha), \textit{Do}(i, \alpha) \rightarrow \textit{Do}(j, \beta))$$

In this locution, agent j requests that agent i performs action α and supports that request with an argument stating that if i accepts, j will perform action β in return. For the locution to properly express a promise, agent j must believe that action β is desired by agent i. If, on the contrary, j believes β is undesirable to i, the same locution might deter i from executing α, and can be seen as a threat. The locution `Request`, however, does not include information that conveys this distinction. If protocol designers wish to have expressive locutions that capture such meaning, they need to provide their own negotiation-specific locutions which hold, within them, the appropriate semantics of the messages. For example, Sierra et al. [1998] and Ramchurn et al. [2003] provide explicit locutions for expressing threats and rewards (e.g. $\texttt{threaten}(i, j, \alpha, \beta)$ and $\texttt{promise}(i, j, \alpha, \beta)$).

Negotiation Protocol

Given a communication and domain language, a negotiation framework should also specify a *negotiation protocol* in order to constrain the use of the language. Here I view a protocol as a formal set of conventions governing the interaction among participants [Jennings et al., 2001]. This includes the *interaction protocol* as well as other rules of the dialogue.

The interaction protocol specifies, at each stage of the negotiation process, who is allowed to say what. For example, after one agent makes a proposal, the other agent may be able to accept it, reject it or criticise it, but might not be allowed to ignore it by making a counterproposal. The protocol rules might be based solely on the last utterance made, or might depend on a more complex history of messages between agents.

Other rules that form part of the negotiation protocol may address the following issues [Jennings et al., 2001, Esteva et al., 2001]:

- **Rules for admission**: specify when an agent can participate in a negotiation dialogue and under what conditions.
- **Rules for participant withdrawal**: specify when a participant may withdraw from the negotiation.
- **Termination rules**: specify when an encounter must end (e.g. if one agent utters an acceptance locution).
- **Rules for proposal validity**: specify when a proposal is compliant with some conditions (e.g. an agent may not be allowed to make a proposal that has already been rejected).
- **Rules for outcome determination**: specify the outcome of the interaction. In an auction-based framework, this would involve determining the winning bid(s) [Sandholm, 2002b]. In argumentation-based frameworks, these rules might enforce some outcome based on the underlying theory of argumentation (e.g. if an agent cannot construct an argument against a request, it accepts it [Parsons et al., 1998]).
- **Commitment rules**: specify how agents' commitments should be managed, whether and when an agent can withdraw a commitment made previously in the dialogue,

how inconsistencies between an utterance and a previous commitment are accounted for, and so on.

With respect to the interaction protocol, a variety of trends can be found in the ABN literature. Interaction protocols can be either specified in an *explicit* accessible format, or be only *implicit* and "hardwired" into the agents' specification. Explicit specification of interaction protocols may be carried out using *finite state machines* [e.g. Sierra et al., 1998, Parsons et al., 1998], or *dialogue games*[16] [e.g. as in Amgoud et al., 2000, Amgoud and Parsons, 2001, McBurney et al., 2003].

Information Stores

In many ABN frameworks there is a need to keep externally accessible information during interaction. For example, we might need to store the history of utterances for future reference or to store information about the reputation of participants [Yu and Singh, 2002, Rubiera et al., 2001]. Having external information stores also makes it possible to *enforce* protocol-related behaviours. For example, we may be able to prevent an agent from denying a promise it has previously made.

One type of information store that is common in the argumentation literature is the *commitment store*. Commitment stores were initially conceived by Hamblin [1970] as a way of tracking the claims made by participants in dialogue games. Hamblin's notion of commitment store has been influential in later work on dialogue games, both in philosophy and in multi-agent systems, although the notions of commitment used sometimes differ. In the work on the philosophy of dialogue [e.g. Walton and Krabbe, 1995] the focus is on action commitments, i.e., promises to initiate, execute or maintain an action or course of actions. Commitments to defend a claim if questioned, called propositional commitments, are viewed as special cases of such action commitments by these authors. In the multi-agent systems literature the concern is usually with action commitments, where the actions concerned are assumed to take place outside the agent dialogue. For

---

[16] Dialogue games are interactions between two or more players, where each player makes a *move* by making some utterance in a common communication language, and according to some pre-defined rules. Dialogue games have their roots in the philosophy of argumentation [Aristotle, 1928, Hamblin, 1970]. They have been used in legal applications of artificial intelligence [Bench-Capon et al., 1991, Gordon, 1993, Prakken and Sartor, 2001] and more recently to specify protocols in multi-agent systems [Amgoud et al., 1998, Parsons and Jennings, 1996, Dignum et al., 2000].

example, one agent may commit to providing a specified product or service to another agent.

Commitment stores should not be confused with the *interaction history*, which only records the sequence of utterances during the whole interaction.[17] While the latter only form a passive storage of "unprocessed" utterances, commitments in commitment stores have more elaborate consequences. For example, when an agent asserts a proposition p, it may not only be committed to believe that p holds, but also to defending that p holds (if challenged), not denying that p holds, giving evidence that p holds, and so on [Walton and Krabbe, 1995]. Another difference of commitment stores in comparison with interaction histories is that commitment stores have specific *commitment rules* governing the addition and removal of statements to which the agent is committed. One rule may specify, for example, that if the agent retracted a previously asserted claim, it must also retract every claim based on the former via logical deduction. Another relevant concept is that of *pre-commitment* proposed by Colombetti [2000]. A request pre-commits the utterer in the sense that the utterer will be committed in case the hearer accepts the request. Commitment stores enable us to capture such pre-commitments. The tricky problem of retraction is discussed at length (though not solved) by Walton and Krabbe [1995].

Some of the key questions that need to be addressed in an ABN framework are: Under what conditions should an agent be allowed to retract its commitments and how would this affect the properties of dialogues? Under what conditions should an agent be forced to retract its commitments to maintain consistency? While many of these questions are being investigated in the multi-agent dialogue literature in general [Maudet and Chaib-draa, 2003], there are issues specific to negotiation dialogues. In particular, commitments to providing, requesting, and exchanging resources may require different treatments from commitments in other types of dialogues, such as persuasion or information seeking.

2.5.4 Elements of ABN Agents

I refer to an agent involved in negotiation interactions which largely depend on exchanging proposals, such as auction-based and bargaining agents, as a *classical negotiating*

---

[17] Sierra et al. [1998] use the term *negotiation thread*, while Sadri et al. [2001b] use the term *dialogue store*.

agent. This agent, sketched in Figure 2.1,[18] needs to have a *locution interpretation* component, which parses incoming messages. These locutions usually contain a proposal, or an acceptance or rejection message of a previous proposal. They might also contain other information about the interaction, such as the identity of the sender (especially in the case of multi-party encounters). Acceptance messages usually terminate the encounter with a deal. A proposal may be stored in a *proposal database* for future reference. Proposals (or rejections) feed into a *proposal evaluation and generation component*, which ultimately makes a decision about whether to accept, reject or generate a counterproposal, or even terminate the negotiation.[19] This finally feeds into the *locution generation* component which sends the response to the relevant party or parties. A more sophisticated classical agent may maintain a *knowledge base* of its mental attitudes (such as beliefs, desires, preferences, and so on [Wooldridge, 2002]), as well as models of the environment and the negotiation counterpart(s). This knowledge may be used in the evaluation and generation of proposals by judging the validity and worth of proposals made (for example, by verifying whether proposals are actually feasible and do not conflict with the current observations of the environment). Moreover, the knowledge base may be updated in the light of new information. However, the updates that can be made are somewhat limited because the only information usually available to the agent during the interaction is:

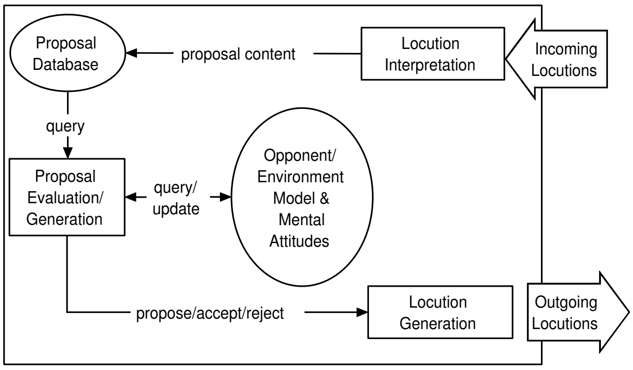


Figure 2.1: Conceptual Elements of a Classical Negotiating Agent

---

[18] This model is not meant to be an idealisation of all existing models in the literature, but rather a useful abstraction for understanding and comparison.

[19] Note that the way components operate is constrained by the negotiation protocol. For example, in English auctions, which are highly *asymmetrical*, only one agent usually receives bids while others only generate them. In bargaining models [e.g. Kowalczyk and Bui, 2001], however, the protocol tends to be *symmetrical*, allowing both agents to evaluate and generate proposals.

1. Proposals (or bids) from the counterpart or a competitor.
2. A message rejecting a proposal initially made by the agent.
3. Other observations from the environment (e.g. a manufacturing plant agent bidding for raw material may monitor customer demand changes and bid accordingly).

The agent may be able to infer certain things from this information. For example, by receiving a rejection the agent may infer that the counterpart does not rate certain attribute/value assignments highly. Similarly, by receiving a proposal (or by observing the proposal of another competing bidder) the agent might infer attribute values that appeal to the counterpart (or competitor), which can then guide his own bargaining or bidding strategy.[20]

In contrast with a classical negotiating agent, richer meta-level information can be explicitly exchanged between the ABN agents (see Figure 2.2).[21] This, in turn, can have a direct effect on the agent's knowledge base. Therefore, in addition to evaluating and generating proposals, an agent capable of participating in argument-based negotiation must be equipped with mechanisms for *evaluating* arguments (and updating the mental state accordingly) and for *generating* and *selecting* arguments. If the locution contains an argument, an *argument evaluation* or *interpretation* mechanism is invoked which updates the agent's mental state accordingly. This may involve updating the agent's mental attitudes about itself and/or about the environment and the counterparts. Now, the agent can enter the *proposal evaluation* stage in the light of this new information. Note that at this stage, not only does the agent evaluate the most recent proposal, but it can also re-evaluate previous proposals made by its counterparts; these proposals are stored in the proposal database. This is important since the agent might (intentionally or otherwise) be persuaded to accept a proposal it had previously rejected.

As a result of evaluating proposals, the agent may generate a counterproposal, a rejection, or an acceptance. In addition, however, a final *argument generation* mechanism is responsible for deciding what response to actually send to the counterpart, and what

---

[20] Similar issues have been investigated in the study of *signalling* in game-theory [Spence, 1974].

[21] The actual way in which ABN agents are designed or implemented may differ from the above. For example, the agent might perform certain operations in a different order, or might combine or further decompose certain functionalities. Therefore, Figure 2.2 is to be taken in the abstract sense and should not be seen as a prescriptive account of what ABN agents *must* precisely look like.

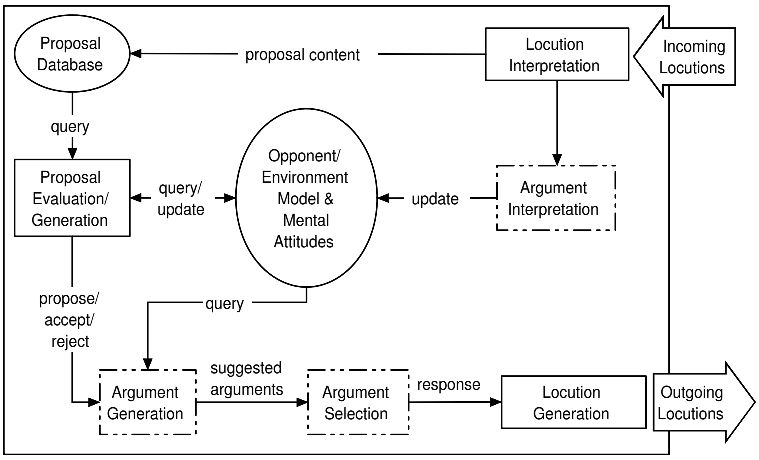


Figure 2.2: Conceptual Elements of an ABN Agent (the dashed lined boxes represent the components necessary for ABN agents in additional to those of a classical negotiating agent).

(if any) arguments should accompany the response. For example, the proposal evaluation and generation component might decide that a proposal is not acceptable, and the argument generation mechanism might accompany the rejection with a critique describing the reasons behind the rejection. Such arguments might also be explicitly requested by the other party or even enforced by the protocol. Note that an agent may also choose to send a stand-alone argument (that is, not necessarily in conjunction with a proposal, acceptance or rejection).

At times, there might be a number of potential arguments that the agent can send. For example, in order to exert pressure on a counterpart, an agent might be able to either make a threat[22] or present a logical argument supporting some action. Deciding on which argument to actually send is the responsibility of an *argument selection* mechanism. Finally, this information is given to the *locution generation* mechanism which places this information in the proper message format and utters it.

In summary, in addition to evaluating and generating proposals and locutions, an ABN agent must be able to:

1. evaluate incoming arguments and update its own mental state accordingly,

---

[22] Promises and threats have also been studied in evolutionary game theoretic models [Samuelson, 1998]. For example, by punishing noncooperative moves by its opponent, an agent sends an indirect threat for future iterations of the game. However, such threats and rewards span over multiple, complete iterations of the same encounter, rather than being part of a single encounter.

2. generate candidate outgoing arguments, and
3. select an argument from the set of candidate arguments.

In the next subsections, I consider each of these features in more detail.

Argument and Proposal Evaluation

Argument evaluation is a central topic in the study of argumentation and has been studied extensively by philosophers at least from the days of Aristotle [Aristotle, 1928, Hitchcock, 2002]. In Artificial Intelligence, argument evaluation and comparison has been applied, for example, in internal agent deliberation [Kakas and Moraitis, 2003], in legal argumentation [Prakken and Sartor, 2001] and in medical diagnosis [Krause et al., 1995, Fox and Parsons, 1998].

In negotiation, it is crucial to distinguish between two types of considerations in argument evaluation:

Epistemic Considerations: If two agents are reasoning about what is true in the world (i.e., if they are conducting *epistemic reasoning*[23]), then it makes sense for them to adopt a more-or-less "objective" convention that is not influenced by their individual desires. For example, whether it is sunny outside should not be influenced by whether participants want it to be sunny, but rather only by the material evidence available. An argument may be seen as a *tentative proof* for some conclusion. Hence, an agent, or a set of agents, may evaluate arguments based on some *objective convention* that defines how the quality of a proof is established. This may be done, for example, by investigating the correctness of its inference steps, or by examining the validity of its underlying assumptions. For example, Elvang-Gøransson et al. [1993b] propose a classification of arguments into *acceptability classes* based on the strength of their construction. Arguments may also be evaluated based on their relationships with other arguments. For Dung [1995], for instance, an argument is said to be *acceptable* with respect to a set S of arguments if every argument attacking it is itself attacked by an argument from that set.

---

[23] I.e. reasoning relating to, or involving knowledge, also sometimes referred to as *theoretical reasoning*.

Motivational Considerations: Agents participating in negotiation are not concerned with establishing the truth *per se*, but rather with the satisfaction of their needs (or desires). For example, an agent's desire to hang a picture may be motivated by the desire to enjoy its colours. If such motivations are purely internal to the agent and cannot be articulated, then they cannot be subject to critical discussion and hence cannot be argued about. I refer to such kinds of criteria as *motivational*. Negotiation dialogues require agents to evaluate deals using motivational criteria, while evaluating the context of their decisions based on objective epistemic criteria.

Note that by using the term "objective" to describe conventions governing epistemic reasoning, I do not mean that agents' epistemic reasoning corresponds to the actual truth. Instead, the term "objective" is meant to denote that beliefs are subject to critical scrutiny by other agents according to some common convention that defines "correct reasoning." After all, beliefs have a subjective element that may be influenced by agent's biases, by the reasoning mechanism used, or by the reliability of the information sources upon which reasoning is based.

Negotiation must take into account objective as well as subjective epistemic aspects. For example, a potential airline traveller may perceive a particular airline as unsafe, while the staff of the airline itself may consider it to be safe. Presumably such a difference in perceptions may be resolved with recourse to objective criteria (if any can be agreed) regarding relative crash statistics, deaths-per-mile-flown on different airlines etc. But if, for example, potential travellers perceive a particular airline as unsafe compared to other airlines, despite objective evidence showing the airline to be safer than others, this perception may inhibit them from flying the airline anyway. The marketing team of the airline concerned, in trying to persuade potential travellers to fly with it, will have to engage in dialogue with potential customers on the basis of those customers' subjective perceptions, even though such perception may be false. For the marketers to ignore such mis-perceptions risks the dialogue terminating without the potential customers flying the airline.

In summary, negotiating agents need to perform *objective* epistemic argument evaluation as part of their *subjective* motivational proposal evaluation. Moreover, they need to consider each others' *subjective* epistemic biases in order to effectively reach an agreement.

Argument Generation and Selection

Another central problem in the study of argumentation is that of argument generation and selection [Zukerman et al., 1998]. The first part of this problem is concerned with *generating candidate arguments* to present to a dialogue counterpart. In negotiation, arguments are usually sent in order to entice the counterpart to accept some proposed agreement. Hence, argument and proposal generation are closely related processes.

Related to the problem of argument generation is that of *argument selection*. The question of argument selection is as follows: given a number of candidate arguments an agent may utter to its counterpart, which one is the “best” argument from the point of view of the speaker?

Note that argument *selection* may take place in conjunction with argument *generation*. An agent need not generate all possible arguments before it makes a selection of the most suitable one. Instead, the agent may only concern itself with the generation of the most suitable argument itself. In other words, the agent might prune the set of candidate arguments during the process of argument generation. Whether or not this is possible, of course, depends on the nature of the argumentation framework underlying the agent’s reasoning.

2.6 Extant Frameworks for Argument-Based Negotiation

In the previous section, I presented a general view of the elements that form a comprehensive argumentation-based negotiation framework. I also discussed the essential components (or computational mechanisms) needed by agents capable of engaging in ABN. In this section, I review a number of the most prominent argument-based negotiation frameworks that have been proposed to date.[24] Specifically, I explain the general ideas behind each framework and then relate them to the conceptual sketch presented above.

---

[24] The order in which the frameworks are reviewed is based on convenience and is not significant in any way.

2.6.1 Kraus, Sycara and Evenchick

The work by Kraus et. al builds upon a number of papers on the PERSUADER system developed by Sycara [Sycara, 1985, 1990, 1992]. The PERSUADER is an agent that aims to settle labour negotiations between a union agent and a company agent using persuasive argumentation. The author presents persuasive argumentation as a means to change an opponent's perceptions, goals and constraints using a number of techniques such as case-based reasoning, preference analysis, and a number of heuristics that help in generating proposals and arguments during negotiation. The framework relies on the construction of hierarchical goal structures and attempts to reconcile them [Sycara, 1990]. However, from the descriptions given, it is not apparent how the framework, which mostly describes the reasoning of a single agent mediating negotiation among human opponents, could be extended to (artificial) multi-agent systems. To this end, a successor of the PERSUADER was introduced by Kraus et al. [1998]. This framework presents argumentation as a mechanism for achieving cooperation between BDI (belief, desire, intention) agents [Rao and Georgeff, 1995]. Hence, the domain language is a multi-modal language with modalities for beliefs, desires, goals, preferences, intentions and so on. The authors claim that their model can characterise negotiation in which an agent causes another to add a new intention, retract an existing intention, or change its preferences.

The authors present a number of argument types that can be used by the negotiating agents. These arguments are drawn from work in the psychology of persuasion [O'Keefe, 1990, Pruitt, 1981, Karlins and Abelson, 1970]. The common types of arguments taken into account are: threats, promises, appeal to past promises, appeal to precedents, appeal to prevailing practice, and appeal to self-interest (to convince the persuadee that taking this action will enable the achievement of an important goal).

There are no specific illocutions supplied to describe threats, rewards or appeals, but these are expressed using other primitives. For example, in the formula $\neg[t_1, Do(i, \alpha)] \rightarrow [t_2, Do(j, \beta)]$, agent j threatens to perform β at time t_2 if agent i did not perform α at t_1.

The interaction protocol rules are expressed implicitly in the agent programs and agent designers are assumed to conform to them. There is no external mechanism for enforcing such rules, and there is no notion of public commitment, or of the consequences of violating commitments. One problem with this approach is that it becomes difficult to verify properties of the protocol independently of the internal specification of agents in the system.

Argument generation and selection is based on ad-hoc rules. By means of an illustration, what follows is an informal description of the threat-generation rule for agent i:

IF
A request has been sent to agent j to perform action α &
j rejected this request & j has goals g_1 and g_2 & j prefers g_2 to g_1 &
doing α achieves $\neg g_1$ & doing β achieves $\neg g_2$ &
i believes doing β is credible and appropriate
THEN
i requests α again with the following threat:
if you don't do α, I will do β

If the rule body is satisfied, the corresponding threat will become a *candidate argument*. In case of multiple candidate arguments, the agent selects an argument according to the following argument strength order, with the first being the "weakest" argument: (1) appeal to prevailing practice; (2) counter example; (3) appeal to a past promise; (4) appeal to self-interest; (5) promise of future reward; (6) threat.

The intuition is that a negotiator would progress from weak arguments up to the strongest ones. For example, there is no need to threaten the counter-party if an appeal is sufficient to persuade him/her to accept a request.

Proposal and argument evaluation makes use of a notion of a *collision factor* (assigned 'true' if a requested action conflicts with one of the agent's own intentions), and a *convincing factor* (a value between 0 and 1 assigned through ad-hoc rules). The following rule describes how to assign a convincing value to an appeal to past promise.

IF The agent believes it has made a promise in the past &
The appeal mentions this promise
THEN The convincing factor is set to 1
ELSE it is set to 0.

Finally, argument evaluation is also based on the argument's *acceptability value*; an integer calculated by taking a ratio between the case in which the request is accepted and the case in which it is not.

It is not clear from the paper what happens next. The authors do not explain how they specify the procedure that takes the above numbers and produces a decision about whether to accept or reject the request.

Discussion:

In essence, this paper combines *heuristics*, expressed in terms of ad-hoc rules, with a simple notion of argument. These heuristics rely on the agent having complete knowledge of its counterpart's mental state. If the attitudes of the counterpart are not known, it is very hard to see how arguments can be generated. As a result, the rules concerning argument evaluation and generation are very rigid, leaving little room to reasoning under uncertainty. Moreover, rejection is caused by a contradiction between a request and one of the persuadee's goals. This is a very strong assertion since one can imagine situations where agents reject requests based on their preferences over goals, or simply because they are not willing to help.

The framework presents a ramping approach to selecting arguments; that is, from appeals to threats in increasing strength. The basis of this selection rule is that the negotiation would normally start with weak arguments and then proceed to stronger ones. The rationale is that appeals normally cost less to issue than threats or rewards. This seems reasonable only if agents were not able to reason about past experience with other agents or according to information gathered about a persuadee. Otherwise, it would make sense, for example, to send a threat to an agent known to be non-cooperative together with the first request, rather than going through all sorts of appeals or rewards.

2.6.2 Sierra, Jennings, Noriega and Parsons

Sierra et al. [1998] present an ABN framework motivated by and demonstrated upon a scenario in business process management. The communication language contains locutions of the form `locution`$(sender, receiver, content, time)$. In addition to locutions which express offers, requests, acceptance, rejection and withdrawal, the communication language includes locutions for expressing threats, rewards and appeals. Locutions can be nested within one another. For example, the locution

$$\begin{gathered}\texttt{threaten}(a, b, not\texttt{accept}(b, a, time = 24h, t_2),\\ \texttt{appeal}(a, boss\_of\_b, b = incompetent,\\ not\texttt{accept}(b, a, time = 24h, t_2), t_3), t_1)\end{gathered}$$

means that agent a threatens at time t_1 that if agent b does not accept at time t_2 to perform a task in 24 hours, then a will appeal to b's boss about b's incompetence by referring to the fact that b did not accept the 24 hour proposal.

The protocol is specified as a finite state machine, which terminates whenever an agent utters an `accept` or `withdraw` locution. The protocol does not prevent agents from repeating the same rejected proposal over and over again, resulting in infinite dialogues, and does not provide rules for checking consistency in the agents' commitments. For example, there is nothing in the protocol that prevents an agent from rejecting an offer that is identical to an offer it made one step earlier.

Argument evaluation and generation is based on *authority* deduced from a relation over agent roles. This graph can be used to specify, for each pair of agents, which agent has higher authority. The authors also propose a way of comparing authority levels of *sets of* agents. This can be used to compare arguments involving statements made by multiple agents. An argument H_1 is preferred to another H_2 if and only if the authority level of agents involved in H_1 is higher than those in H_2. As an example, the authors define a "conciliatory" agent, which accepts appeal-to-authority arguments regardless of the content of the justification of the appeal. This means that there would be no difference between a strong appeal and a weak (or even meaningless) one. While authority seems to be a useful factor in evaluating arguments in an organisation, it seems unreasonable to rely solely on it. Despite this limitation, it seems possible to embed particular frameworks for logic-based argumentation to overcome this problem.

The authors base their specification of conflict resolution on the assumption that in the business process management domain, authority is the only way of determining the power of an argument. This is not necessarily true. For example, one agent might simply be unable to perform the service required, due to resource constraints. This holds even if the general manager threatened to fire him. If authority was used, then this agent would make a commitment that is clearly unachievable. I suggest that a more objective way of resolving conflicts (namely, logic-based argumentation techniques) needs to be adopted in order to overcome this problem.

In terms of dialogue strategies (argument generation and selection), the authors define two simple strategies only. An *authoritarian* agent always exploits its social power by threatening whenever possible, while a *conciliatory* agent always resorts to any explanatory argument if one exists.

2.6.3 Ramchurn, Jennings & Sierra

Ramchurn et al. [2003] build upon the dialogic framework and negotiation protocol of the framework of Sierra et. al (see previous section). The authors focus on the key mechanisms needed to carry out argumentation-based negotiation; that is, argument generation, selection and evaluation. The approach taken is based on the psychology of persuasion, similar to Kraus et. al, whereby arguments can be threats, rewards or appeals. The framework relies on trust [Ramchurn et al., 2003, Marsh, 1994, Sabater and Sierra, 2002] and utility maximisation [von Neuman and Morgenstern, 1944] to guide the reasoning of agents.

The reasoning mechanisms of an agent are given in terms of preconditions and postconditions of actions and using two evaluation functions; the function $V^\alpha : S \rightarrow [0,1]$ is used by agent α to evaluate the current world state, while $EV^\alpha : A \times S \rightarrow [0,1]$ evaluates the *expected* value of the state resulting from executing an action in a particular state. Agents make proposals that bring them maximum utility. Hence, an agent will only propose p in state s if the following is true: $\{EV^\alpha(s,p) > V^\alpha(s)\}$. Arguments also have axiomatic pre-conditions. For example, utterance $threaten(\alpha, \beta, p, th)$ means that agent α threatens to carry out threat th if agent β did not accept proposal p. This threat has the following pre-condition:

$$\left\{ \begin{array}{c} B^\alpha(V^\beta(s) > EV^\beta(s,p)), \\ B^\alpha(V^\beta(s) > EV^\beta(s,th)), \\ B^\alpha(V^\beta(\delta(s,p)) > V^\beta(\delta(s,th))) \end{array} \right\}$$

Thus, for an agent α to threaten another agent β, α must believe that β prefers staying in the current state to enacting the proposal and that β can be threatened (meaning α

believes β will prefer to stay in the current state to having the threat effected). Also, α must believe that the state brought about by the threat is less preferred by β than the state brought about by the proposal (in the third condition). At the receiving end, the following axiom describes the post-condition of a threat, whereby β now has a new belief about α's preferences.

$$\left\{ \; B^{\beta}(B^{\alpha}(V^{\beta}(s) > V^{\beta}(\delta(s, th)))) \; \right\}$$

The *strength value* of an argument is calculated using a function $SV : A \rightarrow [0, 1]$, which depends on some evaluation over resulting states. A strong argument is more likely to force the opponent to accept a proposal. However, the stronger an argument is, the more likely it is to lessen the opponent's *trust* in the proponent, which then affects future cooperation. Thus strong arguments should only be sent, for example, when the negotiation needs to take place in the shortest time possible, when the proposal has high utility for the proponent or when it is known that the other partner cannot be trusted to efficiently reach agreements. Otherwise weaker arguments should be used. The following (fuzzy) rule exemplifies the encoding of these intuitions. The authors present simulation results that support a variety of such rules.

RULE 1: *if* ***trust*** *is* ***low*** *and* ***utility*** *of the proposal is* ***high***
(I need to do X and I don't trust you)
then send a ***strong argument***
RULE 2*: if* ***trust*** *is* ***high*** *and* ***utility*** *of the proposal is* ***low***
(I don't really need to do X and I trust you)
then send a ***weak argument***

Proposal and argument evaluation is based on the evaluation of possible states of the world given the proposal, the argument and the current state of the world. In case of a threat, for example, the agent compares the value of the state where the opponent's threat has been enacted against the value of the proposal requested by the opponent being enacted. The evaluation function then chooses the maximum of the two. The authors also use *trust* to evaluate whether the agent will actually follow through by enacting its threat or promise, or it is just "bluffing."

Discussion:

This paper is another attempt to reconcile heuristic and argument-based approaches, in a similar manner to Kraus et al. [1998]. It is an advancement over the earlier framework of Kraus et al. [1998] since it provides clearer reasoning techniques about arguments in terms of resulting states. However, the framework described here also shares its main limitation: argument evaluation is solely based on subjective motivational criteria (namely trust and utility valuation). There is no way for agents to influence each other's preferences by arguing about the state of the world. This is because the functions EV and V are fixed, and preferences are changed only as a result of carrying out actions that change the resulting state. Another strong assumption is that functions EV and V are assumed to be complete (that is, defined for all states and all actions).

2.6.4 Parsons, Jennings and Sierra

Parsons et al. [1998] present an ABN framework based on a particular formal theory of argumentation [Elvang-Gøransson et al., 1993a, Fox et al., 1992, Krause et al., 1995]. The framework was explained in the context of belief-desire-intention (BDI) negotiating agents and implemented using multi-context systems [Giunchiglia and Serafini, 1994].

Consider the following example from Parsons et al. [1998]. An agent a intending to hang a picture would produce, after executing its planning procedure, intentions to acquire a nail, a hammer and a picture. Interactions with other agents are only motivated in case the agent is not able to fulfil its intentions on its own. Suppose the agent does not have a nail. This leads the agent to adopt a new intention (we can refer to that as a sub-intention) to acquire a nail, which may be written $I_a(\mathit{Have}(a, \mathit{nail}))$. If a believes that another agent b has a nail, it would generate another sub-sub-intention that b gives the nail to it, written $I_a(\mathit{Give}(b, a, \mathit{nail}))$. This triggers a request to be made to agent b in the form $H_1 \vdash I_a(\mathit{Give}(b, a, \mathit{nail}))$, where the argument H_1 constitutes the sequence of deductive steps taken to reach the request.[25] In general, agent b accepts the request unless it has a conflict with it. There are two types of conflicts that would cause b to reject the request:

1. Agent b has a conflicting intention. In argumentation terms, the agent refuses the

---

[25] Note that the argument (or plan) may not contain intentions only, but also belief and desire formulae about the agent and its environment. For example, in the argument H_1, agent a may state the assumption that it believes b has a nail, and so on.

proposal if it can build an argument that *rebuts* it.

2. Agent b rejects one of the elements of the argument supporting the intention that denotes the request. In argumentation terms, the agent refuses the proposal because it can build an argument that *undercuts* it.

I shall explain the above two cases using the same picture-hanging example. An example of the first case is if agent b rejects the proposal because it also needs the nail, say to hang a mirror; that is, it can build an argument for the intention $I_b(\neg \mathit{Give}(b, a, \mathit{nail}))$. This argument is based on (among other things) the intention $I_b(\mathit{Can}(b, \mathit{hang}(\mathit{mirror})))$. An example of the second case is if, in the plan supporting the intention $I_a(\mathit{Give}(b, a, \mathit{nail}))$, agent a made the false assumption that b possesses a nail, written $B_a(\mathit{Have}(b, \mathit{nail}))$. If b actually does not have a nail, then it would adopt the intention of modifying that belief, i.e. $I_b(\neg B_a(\mathit{Have}(b, \mathit{nail})))$. Agents continue through a process of argument exchange, which may involve recursively undercutting each other's arguments until a resolution is reached.

In order for argumentation to work, agents must be able to *compare* arguments. This is needed, for example, in order to be able to reject "weak" arguments. Parsons et al. compare arguments by classifying them into *acceptability classes* based on the strength of their construction [Elvang-Gøransson et al., 1993b]. If two conflicting arguments belong to the same class, the authors assume the agent has some capability to perform comparisons based on utility analysis. However, they do not specify how this decision procedure is actually undertaken, nor do they specify the conditions it needs to satisfy.

The model assumes that a record of all inference steps exchanged between agents is kept. This allows for one agent to use information uttered by another agent in constructing its own arguments. For example, after one agent states that it intends something, the counter-party might use that intention in constructing a plan for achieving its own goals and putting it forward. There is no mechanism for retracting commitments from the information store.

Discussion:

This framework is relatively expressive, because it enables agents to distinguish their beliefs, desires, and intentions, and to treat these within a well-established argumentation theory. The framework was subsequently developed by Simon Parsons with Leila Amgoud and Nicolas Maudet [Amgoud et al., 2000]. The extension involves a dialogue-game protocol to enable dialogues that capture the proof theory of an argumentation logic based on preferences over beliefs [Amgoud and Cayrol, 1998, 2002]. The protocol involves commitment stores, and locutions for expressing requests, refusals, promises, and preferences. More recently, the authors started investigating strategic move selection in persuasive dialogues [Amgoud and Maudet, 2002], as well as in inquiry and information seeking dialogues [Parsons et al., 2002], but not as yet in negotiation dialogues.

One problem with the framework is that it requires agents to be fully cooperative and provide complete arguments (that is, complete plans) along with their proposals. As a result, this approach would not be appropriate in competitive environments in which agents would prefer to keep their resources to themselves (in case they need them later) or attempt to get something in exchange for anything they give.

The suitability of the argumentation theory used in this framework is questionable. In particular, arguments are evaluated based on their acceptability classes or preferences over beliefs. This means that both frameworks require shared agent preferences over arguments.[26] Hence, while arguments can be evaluated objectively, it is not clear how subjective motivational criteria, such as desires or preferences over outcomes, can be applied within the same framework. This is because the theory requires that proposal evaluation is embedded as part of argument evaluation.

Finally, the authors specify a rule stating that an agent cannot repeat the same move previously uttered by it or by another agent. The intuition is that this would prevent the agent from, say, repeating the same question over and over again. This turns out to be problematic. Suppose a request for p is made by agent A_1 to agent A_2 followed by a refusal from A_2. A_1 might utter a promise to do q in return: $\textit{promise}(p \Rightarrow q)$. Suppose A_2 accepts on the basis that q is useful to the achievement of one of its own intentions. Now A_2 cannot refuse this proposal later on, say based on new arguments that render q useless to it. More work needs to be done on the dynamics of this interaction. Moreover, we need a more comprehensive account of the conditions under which the dialogues may be successful or even terminating.

---

[26] An attempt to deal with conflicting agent preferences was made [Amgoud and Parsons, 2001], but these preferences had to be resolved based on agent authority before arguments can be compared.

2.6.5 Sadri, Torroni & Toni

Sadri, Torroni and Toni present another logic-based approach to argument-based negotiation. Their approach, inspired by Parsons et al. [1998],[27] was reported in a series of papers [Sadri et al., 2001a, Torroni and Toni, 2001, Sadri et al., 2002]. This framework extends an existing architecture for multi-agent systems [Kowalski and Sadri, 1999] based on a particular abductive proof procedure for logic programs [Fung and Kowalski, 1997, Sadri and Toni, 2000]. The idea is to split the proof procedure in such a way as to make it dialogic (i.e. involving two agents). This is, in essence, the same idea underlying the work of Amgoud et al. [2000] described above.

Intentions are denoted using the $\mathit{plan}(.)$ predicate. The following expression, for example, denotes a plan (or intention) to hit the nail and hang the picture. The resources this plan requires are a picture, a nail and a hammer:

$$plan(\langle hit(nail), hang(picture)\rangle, \{picture, nail, hammer\})$$

The basic dialogue performative is $\mathit{tell}(a, b, \mathit{Move}, t)$, where a and b are the sending and receiving agents, t is the time of the utterance, and "*Move*" is a locution defined recursively as follows:

1. $\mathit{request}(\mathit{give}(R))$ to request resource R.
2. $\mathit{promise}(\mathit{give}(R), \mathit{give}(R'))$ to propose and to commit to give resource R' in return for R.
3. $\mathit{accept}(\mathit{Move})$, $\mathit{refuse}(\mathit{Move})$ to accept or refuse the dialogue move "*Move*."
4. $\mathit{challenge}(\mathit{Move})$ to ask for a justification for the move.
5. $\mathit{justify}(\mathit{Move}, \mathit{Support})$ to justify a move by some support.

Since the framework is based on declarative logic programming, the protocol is presented as a set of *dialogue constraints* which take the form of if-then rules. Each protocol rule is of the form $P(t) \wedge C(t) \Rightarrow P'(t+1)$, which means that if the agent received performative P at time t, and condition C was satisfied at that time, then the agent *must*

---

[27] This claim is based on personal communication with Paolo Torroni.

use the performative P' at the next time point. The condition C describes the rationality precondition in the agent's mental state. The following rule, for example, states that if an agent received a challenge for its request, it would provide a *justification* that supports the request. The justification involves the plan the agent is trying to execute, the goal that this plan achieves, and the missing resources in the plan (which include the requested resource R).

$$\begin{gathered} \mathit{i\_am}(X) \wedge \mathit{tell}(\mathit{Agent}, X, \mathit{challenge}(\mathit{request}(\mathit{give}(R)))) \\ \wedge \mathit{miss}(R, \mathit{Plan}, \mathit{Goal}, \mathit{Missing}) \wedge \\ \mathit{Support} = \{\mathit{missing}(\mathit{Missing}), \mathit{plan}(\mathit{Plan}), \mathit{goal}(\mathit{Goal})\} \\ \Rightarrow \mathit{tell}(X, \mathit{Agent}, \mathit{justify}(\mathit{request}(\mathit{give}(R)), \mathit{Support})) \end{gathered}$$

Once uttered, all dialogue moves are recorded on a publicly accessible blackboard called the *dialogue store*. However, no commitment store processing is done.

Proposal evaluation follows a simple rule. The agent accepts a request unless it needs it to achieve one of its own goals (i.e. unless it is part of one of its intentions). This is technically enforced by the protocol through the following rule, which means that if agent X received a request for resource R from agent "*Agent*", and if it had resource R and did not need it, then it must accept the request:

$$\begin{gathered} \mathit{i\_am}(X) \wedge \mathit{tell}(\mathit{Agent}, X, request(\mathit{give}(R))) \wedge \mathit{have}(R) \wedge \neg \mathit{need}(R) \\ \Rightarrow \mathit{tell}(X, \mathit{Agent}, \mathit{accept}(\mathit{request}(\mathit{give}(R)))) \end{gathered}$$

Argument evaluation is also specified in the form of the protocol rules. The following rule, for example, states that an agent would refuse a request even after a justification if the counter-party cannot find an alternative plan for its own goal that could be proposed for a deal.

$$\begin{gathered} \mathit{i\_am}(X) \wedge \mathit{tell}(\mathit{Agent}, X, \mathit{justify}(\mathit{request}(\mathit{give}(R)), Support)) \wedge \\ goal(Goal) \in \mathit{Support} \wedge \mathit{miss}(R', \_, \_, \_) \wedge \\ \neg \mathit{exists\_alternative\_plan}(\mathit{Goal}, \mathit{without}(\{R', R\})) \\ Rightarrow\mathit{tell}(X, \mathit{Agent}, \mathit{refuse}(\mathit{give}(R))) \end{gathered}$$

Discussion:

This framework has a number of important features. It offers a comprehensive model of agent interaction in which both the interaction protocol as well as the agents' internal reasoning mechanism are integrated in a declarative manner. As a result, it was possible to implement the framework successfully [Ciampolini et al., 2003]. This detailed specification also allowed studies of the properties of the dialogue [Sadri et al., 2001a, Torroni, 2002] such as termination and success and the conditions under which these properties hold. For example, Torroni [2002] determined an upper limit to the maximum length of a dialogue, measured in the number of exchanged messages. However, since these results were produced by referring to the machinery of abductive logic programs, it is not clear whether they apply to more general ABN dialogues.

Another limitation of the framework is that the protocol is embedded in the agent specification. As I argued earlier, this makes dialogue monitoring and protocol enforcement difficult.

2.6.6 McBurney, van Eijk, Parsons & Amgoud

Recently, McBurney, van Eijk, Parsons and Amgoud [McBurney et al., 2003] introduced a framework for automating a particular type of argument-based negotiation scenario. This framework does not subscribe to a particular argumentation theory and is less concerned with the internal decision-making mechanisms of participating agents. Rather, the paper is concerned with the locutions and interaction protocols. Specifically, it presents a set of locutions and a dialogue-game protocol that facilitate a particular class of purchase negotiation scenarios. The paper shows that if the internal decision mechanisms within the agents satisfy certain conditions, these mechanisms would be sufficient to generate automated negotiation dialogues according to the proposed interaction formalism.

The framework has three types of agents: *consumers*, *sellers* and *advisors*. An offer exchanged between participants is referred to as a *sales option* or a *purchase option* depending on who makes the offer. Advisor agents provide information to consumers about sales options (that is, potential offers) by sellers, but do not themselves perform any sales. The dialogue takes the following stages:

1. **Open Dialogue**.

2. **Inform:** The consumer seeks information from sellers and/or advisors about purchase options available.
3. **Form Consideration Set:** Consumer applies its *inclusion criteria* to the purchase options in order to eliminate some of them.
4. **Select Option:** Consumer applies its *selection criteria* in order to pick the most preferred option from the consideration set.
5. **Negotiate:** This may involve a request for an option from the consideration set or a novel option not in the consideration set.
6. **Confirm:** Participants confirm purchase agreements.
7. **Close Dialogue**.

The framework specifies, for each agent, two types of stores: an *information store* and a *commitment store*. An agent's information store contains all purchase or sales options provided by that agent during the dialogue. Entries in the information stores are not binding yet, but rather exploratory. The commitment store contains purchase (or sales) options that the agent has committed to buying (or selling) during the **Confirm** stage of the dialogue. For example, the framework involves two locutions: `agree_to_buy`(.) and `agree_to_sell`(.), for committing to certain resource exchanges. Instead of providing explicit locutions for retracting these commitments, the authors provide additional locutions: `willing_to_buy`(.) and `willing_to_sell`(.), which are softened versions of the former locutions, however, with no commitments incurred (that is, they are free to refuse to sell or buy something they were previously willing to sell or buy). This way, agents may usefully provide information without necessarily committing to it or having to explicitly retract it.

For each communicative message, the paper specifies the *locution*, its *preconditions*, *meaning*, the *possible responses* to that locution, and finally the updates to the *information store* and *commitment store* of each of the participating agents. The following is the specification of the locution that allows a selling agent to announce that it is willing to sell a particular option:

Locution: `willing_to_sell`(P_1, T, P_2, V), where P_1 is either an advisor or a seller, T is the set of participants, P_2 is a seller and V is a set of sales options.
Preconditions: Some participant P_3 must have previously uttered a locution `seek_info`(P_3, S, p) where $P_1 \in S$ (the set of sellers), and the options in V satisfy constraint p.
Meaning: The speaker P_1 indicates to audience T that agent P_2 is willing to supply the finite set $V = \{\bar{a}, \bar{b}, \ldots\}$ of purchase-options to any buyer in set T. Each of these options satisfy constraint p uttered as part of the prior `seek_info`$(.)$ locution.
Response: None required.
Information Store Updates: For each $\bar{a} \in V$, the 3-tuple $(T, P_2, \bar{a})$ is inserted into $IS(P_1)$, the information store for agent P_1.
Commitment Store Updates: No effects.

In a similar manner, the paper specifies locutions that allow an agent to open a dialogue, seek information, announce a desire to buy certain options, declare preferences, refuse to buy, refuse to sell, agree to buy, agree to sell or withdraw from the dialogue.

The authors provide some additional rules to control the dialogue. For example: the *Open Dialogue* stage may only occur once, at least one instance of the *Inform* stage must precede the first instance of each of the *Form Consideration Set*, *Negotiate* and *Confirm* stages etc.

The consumer agent model is inspired by consumer models in marketing theory [Lilien et al., 1992]. However, the authors do not specify agents in detail. Instead, they present a number of *semantic decision mechanisms* which are needed to allow agents to participate in the dialogues specified above. For example, buyers are assumed to possess a mechanism **B6:Generate Novel Options**, where **B6** is the mechanism identifier for the buyer, which generates novel bundles of attributes not among those in its current consideration set. Mechanisms are only specified at this very abstract level.

After specifying the fundamental decision mechanisms needed by sellers, buyers and advisors in order to generate negotiation dialogues, the authors present an *operational semantics* for the dialogue games. They do so by presenting *transition rules* that show how the state of the system changes as a result of uttering locutions and invoking decision mechanisms. For example, the transition rule:

$$\langle P_{B_i}, B6, V(\theta)\rangle \overset{L5}{\rightarrow} \langle P_{S_k} S3, .\rangle$$

states that when a buyer agent P_{B_i} invokes the mechanism **B6** (which generates novel options) and producing as an output of this mechanism the set of options $V(\theta)$, then the agent would utter the locution **L5:desire_to_buy(.)** (which indicates a "desire to buy" to a number of sellers). This locution results in mechanism **S3:Assess Options** (in which an agent assesses newly received options) being invoked in all seller agents.

Discussion:

Even though the paper seems to situate itself in the argument-based negotiation, the process of argumentation seems, at this stage, preliminary and does not yet realise the full potential of using argumentation in negotiation. The only meta-information (that is, other offers) that can be exchanged in the dialogue-game are the preference statements in the locution **prefer**(P_i, S, V, W), which specify that agent P_i informs audience S that it prefers every option in V to every option in W. While this may be very useful, the paper does not specify how this may influence the mental states of the audience. Moreover, the framework does not have locutions for passing other meta-level information such as arguments, proofs, persuasive utterances and so on. Nevertheless, the paper provides a solid first step towards the characterisation of such complex dialogue-games.

2.7 Summary and Discussion

From the discussion it should be clear that there is no universal approach to automated negotiation that suits every problem domain. Rather, there is a set of approaches, each based on different assumptions about the environment and the agents involved in the interaction. Argumentation-based negotiation frameworks are gaining increasing popularity for their potential ability to enable more flexible dialogues than game-theoretic and heuristic-based approaches. However, such models are typically more complex than their game-theoretic and heuristic counterparts. The challenges involved range from defining a clear process for evaluating, generating and selecting arguments, to specifying rich communication languages and protocols, to understanding the dynamics of dialogues and the properties of agent strategies.

In ABN frameworks, argument evaluation depends largely on the object of negotiation and the way agents represent and update their internal mental states. There seems to be two main approaches to argument and proposal evaluation in the literature:

- **Motivation-based (or Value-based) Approach:** In this approach, exemplified by frameworks such as those of Ramchurn et al. [2003] and Kraus et al. [1998], argument evaluation is based solely on the direct comparison of expected utilities. Agents do not influence each other's beliefs, but rather exert pressure on each other by exercising their ability to influence the outcomes (for example, by making a promise or a threat). In other words, an agent would not *voluntarily modify* its position as a result of changes in its beliefs, but rather *forcedly concede* on its position as a result of pressure from its counterpart.

- **Belief-based Approach:** In this approach, exemplified by Parsons et al. [1998] and Amgoud et al. [2000], agents can argue over their beliefs, and potentially influence those beliefs, using a formal theory of argumentation. These beliefs could capture, for example, facts about the environment, the availability of resources, or the achievability of intentions. Preferences among arguments (e.g. derived from acceptability classes [Parsons et al., 1998] or a stratification over the knowledge base [Amgoud et al., 2000]) are then used to help agents evaluate different arguments and reach a consistent conclusion. Preferences among arguments can intuitively capture the relative epistemic strength of different arguments (e.g. the degree of confidence in different believed facts). However, when arguments include motivational notions such as desires or goals, the meaning of these preferences becomes unclear. In other words, there is no way of calculating preferences among arguments that contain both epistemic and motivational notions within the same construction.

Opportunities exist for combining the epistemic (belief-based) and motivational (value-based) approaches to argument evaluation and generation in a negotiation framework. This would result in a framework in which agents are able to influence each others' beliefs and preferences, while making a clear distinction between epistemic and motivational aspects of decision-making.[28] In Chapter 3 I present a framework that combines the logical (epistemic) form of an argument with a subjective evaluation of its consequences based on utilities, in a single ABN framework.

---

[28] In the context of *deliberation*, Fox and Parsons [1998] distinguish between *belief arguments* and *value arguments*, and provide a qualitative probabilistic model for combining the two in order to infer the *expected value* of the situation(s) resulting from executing an action.

Chapter 3

A Framework for Interest-based Negotiation

> *"Maybe we could take a step back*
> *and discover what leads us to attack."*
>
> Chuck Schuldiner, Musician (1968 – 2001)

In this chapter, I present a semantic framework for interest-based negotiation and a dialogue game protocol that enables agents to conduct interest-based negotiation dialogues.

3.1 Introduction

In the previous chapter, I argued for the need for *flexible* negotiation mechanisms whereby agents can exchange arguments in order to compensate for their bounded knowledge during negotiation. I showed that existing frameworks for argument-based negotiation have failed to provide a coherent way of combining the "epistemic" argumentation over beliefs with the "motivational" reasoning about one's own goals or desires. In this chapter, I present a semantic framework for interest-based negotiation and a dialogue game protocol that enables agents to conduct interest-based negotiation dialogues. In this framework, epistemic and motivational reasoning are catered for by introducing three separate argumentation theories supporting goals and by making an explicit distinction between (i) the way an agent *subjectively* evaluates its plans and desires and (ii) the way it *objectively* evaluates new arguments from its negotiation counterpart.

The central aim of this chapter is *to define basic data structures and computational modules needed in order to enable negotiation dialogues in which participants can influence each others' preferences by exchanging additional information.* To this end, I present a framework that specifies (i) a reasoning model, grounded in a formal theory of argumentation, whereby agents can update their intentions as a result of receiving new arguments relating to their interests; (ii) a suitable communication language and protocol that enables agents to exchange such arguments; and consequently (iii) a clear account of how communication influences preferences by influencing the agent's mental state. Through an illustrative example dialogue, I show that the framework can allow agents to improve the outcome of negotiation.

The chapter makes two main contributions to the automated negotiation literature. First, the framework allows agents to explicitly influence each others' preferences during negotiation, thus dealing with fundamental limitations of bargaining and auction protocols. Second, the framework extends current emerging efforts on argumentation-based negotiation by separating epistemic argumentation of facts from value-based evaluation of outcomes based on preferences and desires.[1]

The chapter is organised as follows. The next section motivates the need for interest-based negotiation and discusses the structures and features needed in order to enable its automation. Section 3.3 presents a conceptual framework for interest-based negotiation, which is followed by a formal framework in Section 3.4. The formal framework is then used in Section 3.5 to characterises various types of conflicts among agents and how these conflicts may be resolved. A protocol for interest-based negotiation is presented in Section 3.6. In section 3.7, I discuss how the formal framework and protocol are combined, and demonstrate a comprehensive example which puts the various parts together. The chapter concludes in section 3.8.

---

[1] Refer to discussion in Sections 2.5.4 and 2.7.

3.2 Motivation

In this section, I motivate the need for interest-based negotiation and discuss the structures and features needed in order to enable its automation.

3.2.1 Drawbacks of Positional Negotiation

A *negotiation position* of an agent may be informally defined in terms of the resource(s) that agent wants to acquire from its negotiation counterpart. For example, a customer might approach a travel agent because he wants to hire a car, subject to specific budget constraints. On the other hand, an agent's *negotiation interests* reflect the underlying goals it wants to achieve using these resources. For example, the customer might need the car in order to drive from Melbourne to Sydney.

Positional negotiation refers to a negotiation process in which the dialogue between participants is focused on their negotiation positions. Examples of positional negotiations include bargaining, where participants alternately exchange potential agreements.

Consider the following dialogue between potential buyer **B** and seller **S**. Suppose that **B** wants to pay no more than $200 and that **S** wants to charge no less than $400.

Example 4. *Positional negotiation leading to no deal.*

B: I would like to rent a car for 4 days please.
S: I offer you one for $400.
B: I reject! How about $200?
S: I reject!

This dialogue ended with *no deal* because the negotiation positions of participants did not overlap, i.e., there was no agreement possible that would simultaneously satisfy the positions of both negotiators.

Let us consider a variant of the above example. Suppose, this time, that the buyer might concede and pay $300, but would be less satisfied. Suppose also that the seller could concede to $300.

Example 5. *Positional negotiation leading to sub-optimal deal.*

B: I would like to rent a car for 4 days please.

S: I offer you one for $400.

B: I reject! How about $200?

S: I reject! How about we meet in the middle with $300?

B: I guess that's the best I can do! I accept!

In this case, the negotiation ended with a *sub-optimal outcome* from the points of view of each participant. This happened because each agent had to concede into a different, less satisfactory position in order to enable an agreement.

The deficiencies of positional negotiation lie mainly in the underlying assumptions about agents' preferences over possible agreements. In particular, positional negotiation assumes each participant's preferences are *complete* in the sense that the negotiator is able to compare any two offers (which implicitly also means that the negotiator knows the set of all possible offers) and that these preferences are *proper* in the sense that they reflect the actual benefit the negotiator obtains by satisfying them.

However, there are many situations in which agents' preferences are *incomplete* and *improper*. An agent's preferences may be incomplete for a number of reasons. A consumer, for example, may not have preferences over new products, or products he is not familiar with. During negotiation with potential sellers, the buyer may acquire the information necessary to establish such preferences. An agent's preferences may be improper if they are based on inaccurate beliefs.[2] For example, a car buyer might prefer a car of make VOLV over a car of make BM based on false perception about the safety records of VOLVs. Suppose that BMs are in fact safer according to the latest accident statistics. By purchasing a VOLV, the buyer's actual gain (in terms of satisfying his/her safety requirement) is not optimal.

3.2.2 A Solution: Interest-Based Negotiation

Consider the following alternative to the dialogues presented in examples 4 and 5. Let us assume that the seller makes as much money by renting out cars for $400 as she[3] would make by selling Melbourne to Sydney airline tickets for $200.

Example 6. *Interest-based negotiation leading to optimal deal.*

---

[2] By "inaccurate," I mean that the agent's belief is not consistent with the *actual* state of the world.

[3] For clarity, I shall use she/her to refer to sellers and he/his to refer to buyers.

B: I would like to rent a car for 4 days please.

S: I offer you one for $400.

B: I reject! How about $200?

S: I reject! Why do you need the car?

B: Because I want to get to Sydney.

S: You can also go to Sydney by flying there! I can book you a ticket with Qanta airlines for $200.

B: Great, I didn't know flights were so cheap! I accept!

In this example, it was possible to reach an agreement that satisfies both the buyer and seller. This happened because participants discussed the underlying interests (the buyer's interests in this particular case). The interest-based approach to negotiation is advocated extensively in the literature on human negotiation, as illustrated in the following excerpts from Fisher et al. [1991, pp. 40–42]:

> "The basic problem in a negotiation lies not in conflicting positions, but in the conflict between each side's ... interests. Interests motivate people; ... Your position is something you have decided upon. Your interests are what caused you to decide. ...
>
> Behind opposed positions lie shared and compatible interests, as well as conflicting ones. ... In many negotiations, however, a close examination of the underlying interests will reveal the existence of many more interests that are shared or compatible than ones that are opposed."

It is argued by Fisher et al that by discussing the reasons behind positions, negotiators can:

1. *discover the true sources of conflict*, whether these are genuine conflicts between their basic desires, resolvable conflicts between their plans, or simply a result of incorrect choices caused by ignorance; and

2. *find appropriate means to resolve conflicts*;

My aim is to enable computational agents to conduct interest-based negotiation dialogues.

Broadly speaking, I assume that agents are *rational* in two senses of the word: (i) in the *economic-theoretic* sense that they attempt to maximise some value (e.g. by choosing the most preferred option among a set of alternative outcomes, or the option that maximises expected utility) [Mas-Colell et al., 1995]; and (ii) in the *argumentation-theoretic* sense that they are able to base their decisions on the "giving and receiving of reasons" [Johnson, 2000]. An agent ranks outcomes as to "maximise" the value it expects to receive from those outcomes, but at the same time is willing to engage in "argumentation" that could alter those expectations.[4]

I view *interest-based negotiation* (IBN) as a sub-class of *argumentation-based negotiation* (ABN). While ABN encompasses any negotiation framework where agents can exchange arguments to influence each others' mental states, IBN refers to those frameworks where the arguments involved pertain particularly to agents' interests. By "interests" I refer to *the agent's underlying desires and the goals adopted to fulfil those desires*. IBN, therefore, excludes non-interest-based arguments, such as those pertaining purely to authority [Sierra et al., 1998], or resorting to threats [Kraus et al., 1998].

3.3 A Conceptual Framework for IBN

In this section, I define a conceptual framework for interest-based negotiation among autonomous agents. The framework pins down the essential "features" required to enable IBN. I first identify the different types of conflicts among agents and various ways of resolving them. Then I discuss the dialogue abilities required to enable such resolution. This conceptual framework lays the ground for the specific *computational framework* I present in subsequent sections.

Recall from section 2.5.4 that agents capable of conducting argument-based negotiation must be able to: (i) evaluate incoming arguments, (ii) generate candidate outgoing arguments and (iii) select a particular candidate argument to use. IBN focuses on these processes in light of arguments pertaining to agents' interests. Therefore, a crucial first

---

[4] One way to see this in terms of classical decision theory is that "argumentation" is a rational process of altering the probability distributions over lotteries upon which the agent calculates the expected utility of alternative allocations [Mas-Colell et al., 1995].

step is to identify what an interest-based argument is, and how it may be useful in the context of negotiation. This would then enable the formalisation of the mechanism by which goals and preferences are updated in light of interest-based arguments. Argument generation and selection can then follow, though I leave their discussion until Chapter 5.

3.3.1 Identifying and Resolving Conflict in IBN

To enable agents to discuss their interests and discover the underlying conflicts and compatibilities between these interests, it is essential to have an explicit representation of these interests. This will then enable us to understand how conflict over resources can be resolved by exploring and manipulating these interests through dialogue. In this subsection, I discuss these requirements in detail.

The precise meaning of “interests” can be highly domain specific. But in general, we can view interests as the underlying *reasons for choice*. Choices are usually fundamentally motivated by one’s intrinsic *desire*.[5] To simplify the analysis, I assume (see Figure 3.1) that an agent requires resources because they are needed in order to achieve certain *goals*. And these goals are adopted because they contribute to achieving the agent’s fundamental *desires*.[6] Desires themselves are adopted because the agent believes it is in a particular state that justifies these desires. These concepts will become more precise in the discussion below.

The following example clarifies the abstract view of agent reasoning that I adopt.[7]

Example 7. *A customer requires a car hire (a resource) in order to go to Sydney (a goal), which in turn achieves the agent’s wish to attend an Artificial Intelligence conference (a desire). The customer desires to attend the AI conference because he believes it includes a workshop related to his research (a belief that justifies the desire).*

---

[5] E.g. to maximise one’s expected utility [Horvitz et al., 1988].

[6] There is no standard denotation of the terms “desire” and “goal” in the agent literature. For example, some use the term “goal” to describe “consistent desires,” while others use the term “sub-desires” instead of “goals.” My choice of terms is driven solely by convenience.

[7] It may be possible to ground the discussion in a richer theory of practical reasoning. However, I need to scope the discussion somewhat in order to avoid the explosion of formalism complexity.

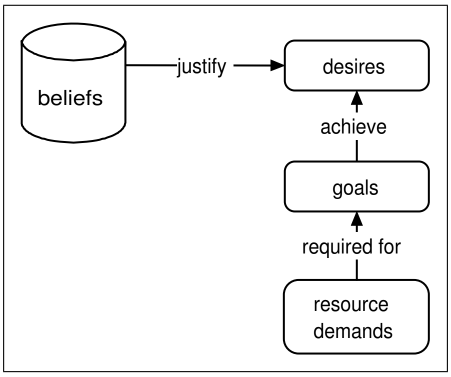


Figure 3.1: Abstract view of relationship between desires, beliefs, goals and resources

Addressing Resource Conflict by Discovering and Manipulating Goals

The most fundamental form of conflict we are dealing with is that over *resources*. Examples of resources include time, fuel, money, communication bandwidth, physical ability etc. Conflict arises because the resources available are not sufficient for all agents to achieve their individual desires. For simplicity, I assume that resources are described in terms of capabilities. Each agent is endowed with a set of resources: actions the agent can execute. For example, a customer may be capable of making a payment of $200 to a travel agent, who in turn is capable of making a travel booking. In the rest of the chapter, I use the terms "action" and "resource" interchangeably. This raises the question of how external or shared resources are represented. For example, if two agents are splitting a cake, then the cake belongs to neither of them. We can capture these by denoting a separate set of actions denoting *shared resources*. These actions are not part of any individual agent's capabilities and can only be executed by *both* agents (e.g. by cutting the cake in half, giving the first half to one agent and the second half to the other).

The direct way of resolving resource conflicts is by each agent individually exploring alternative resource allocations until a mutually acceptable allocation (i.e. deal) is found. This is often referred to as *bargaining*,[8] which may require *concessions* on behalf of the participants until a deal is reached or negotiation fails. This was shown in example 5, where each agent started by presenting his/her most preferred deal and then conceded by accepting an alternative, less preferable deal (exchanging the car rental with $300).

---

[8] Bargaining over resources has been studied extensively in game theory [Osborne and Rubinstein, 1994] and multi-agent systems [Faratin, 2000, Fatima et al., 2002, Larson and Sandholm, 2002].

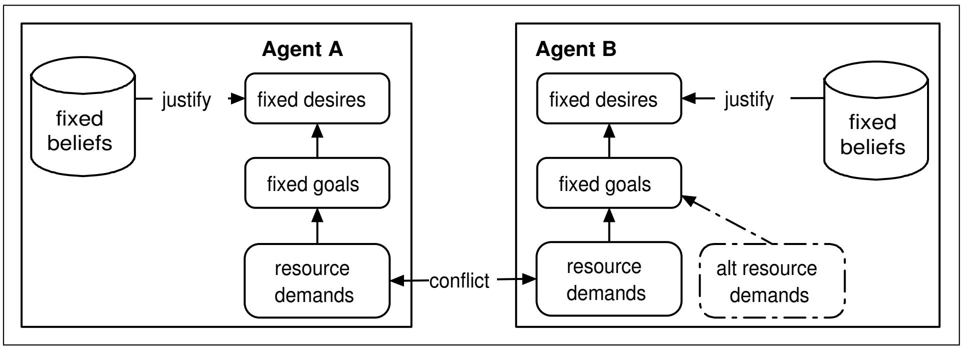


Figure 3.2: Resolving resource conflict by modifying resource demands or offers, either arbitrarily, or guided by knowledge of their and each others' goals

This method for conflict resolution is depicted in Figure 3.2, which shows how an agent accepts an alternative allocation of resources that achieve its (fixed) goals.

In example 6 above, the buyer states that he made a decision to hire a car because it enables him to go to Sydney. After informing the seller of that goal, the seller could propose alternative means for achieving the same goal. The buyer was not initially aware of this alternative and so would not have proposed it. Without knowing the underlying goal, the seller would not have proposed the flight option either, because it would have seemed irrelevant. So, essentially, after exploring the immediate underlying goal, an agent can propose novel alternative deals to its negotiation counterpart. This is an example of the situation described in Figure 3.2. The difference between this and a normal concession, however, is that the buyer did not initially know of the flight alternative.

Note that according to example 7 above, the choice to hire a car is fundamentally motivated by the desire to attend a conference. However, in example 6, it was sufficient for the buyer to inform the seller of the immediate *goal*, namely to get to Sydney. Whether the trip to Sydney is for leisure or academic conferences was an irrelevant detail.

More generally, however, agents can explore each others' goals in more detail. They can also attempt to *influence* each others' goals themselves in order to resolve conflicts. Consider the following dialogue, in which the buyer reveals more information about his goals:

Example 8. *Influencing goals.*

B: I would like to rent a car for 4 days please.

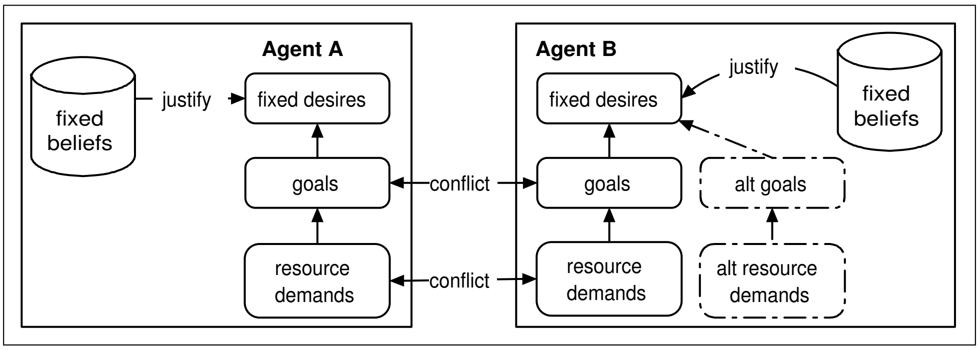


Figure 3.3: Resolving resource conflict by one or both agents adopting alternative goals that achieve the same desires

S: I offer you one for $400.

B: I reject! How about $200?

S: I reject! Why do you need the car?

B: Because I want to drive to Sydney.

S: Why do you need to go to Sydney?

B: Because I want to attend the next AI conference.

S: There is another AI conference in Adelaide; and we have a special offer on tickets to Adelaide now. Would you consider that?

B: I didn't know about this conference! Do you have more details?

In this example, the buyer reveals even more information about his goals, which enables yet another way of resolving the conflict. In this case, the seller influences the buyer's decision by presenting a new means for attending an AI conference, which involves an alternative goal (going to Adelaide instead of Sydney), which, in turn, can be achieved using different resources. The buyer now considers a reasonable deal that involves a ticket to Adelaide acceptable. This conflict resolution method is depicted in Figure 3.3. It shows how influencing an agent's goals offers another way of resolving conflicts over goals.

I summarise the above discussion in the following remark:

Remark 1. (IBN Requirement 1) *Negotiators can benefit from the ability to* reveal *their underlying goals and* influence *each others' underlying goals (or plans) by providing each other with new means for achieving their desires. This* lifting *of conflict from the* *resource level to the goal level can lead agents' to alter their preferences (and possibly demands) themselves and enable deals otherwise impossible.*

Addressing Desire Conflict via Beliefs

A natural question to ask now is: what if after exchanging information about their goals, agents discover that it is not possible to resolve conflicts by modifying the underlying goals? If desires are fixed and there is no conceivable state of the world in which the desires of all agents are fully satisfied, then the only way to resolve the conflict over resources is for some or all agents to concede on some (or all) of their desires. The good news, however, is that if desires themselves can be influenced during dialogue, then there is still room for conflict resolution! Consider the following example:

Example 9. *Influencing desires (variant on example 8).*

> S: Why do you desire to attend the AI conference in Sydney?
>
> B: Because there is an associated workshop relevant to my research.
>
> S: But the workshop has been relocated in conjunction with the Logic conference in Brisbane.
>
> B: Well in this case, do you have any bookings to Brisbane?

In this example, the buyer explains the inference that lead to the desire to attend the conference in the first place. The seller attacks the premises of that inference, thus causing the buyer to change his desire itself; the buyer now desires to attend the Logic conference rather than the AI conference. To achieve that desire, the buyer must adopt an alternative goal to go to Brisbane, which then leads to accepting Brisbane deals. This type of conflict resolution is informally depicted in Figure 3.4, where agent B changes his desire as a result of changing his beliefs. As above, I summarise this in the following remark.

Remark 2. (IBN Requirement 2) *Negotiators can benefit from the ability to find the underlying reasons for generating agents' desires. Hence, if it is not possible to satisfy all agents' goals, influencing the underlying desires can alter these goals and enable agreement.*

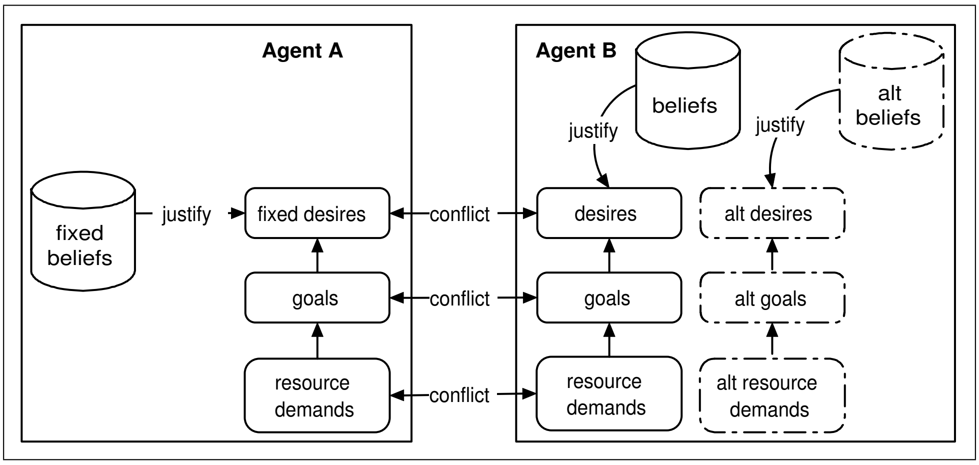


Figure 3.4: Resolving resource conflict by one or both agents changing their beliefs, resulting in different desires altogether

Table 3.1 summarises the different types of conflict discussed above and how they may be resolved through rational dialogue. We added conflicts among beliefs since resolving such conflicts may be required to influence desires and goals.

It is worth clarifying a point about agent beliefs. Recall that I assume agents exhibit a form of argumentation-theoretic rationality. As a consequence, I assume that agents try to maintain beliefs for "good reasons." These beliefs may be incorrect and hence cannot be considered *knowledge*. By receiving new arguments from other agents, an agent may alter its beliefs, but this still does not guarantee that these beliefs are true; they are only well-supported by arguments. The precise semantics of these beliefs is based on the particular argumentation theory I use, as will be seen later in the chapter.

3.3.2 Dialogue Types for Resolving Conflicts

It is clear from the above discussion that enabling interest-based negotiation (IBN) among computational agents requires a non-trivial form of communication. Agents need to bargain with one another over possible deals, seek information from one another about their underlying beliefs, goals, desires and preferences (I shall refer to these as "mental attitudes") and persuade one another to change these mental attitudes. Hence, they need to conduct *information-seeking* dialogues and *persuasion* dialogues as part of negotiation

Conflict Type	**Explanation**	**Resolution Method**
Resource conflict	Insufficient resources to achieve all individual goals	Explore alternative offers, possibly make concessions, or resolve underlying goals/desires conflicts
Goal conflict	Goals interfering with one another	Explore alternative goals, make concessions, or resolve higher-level goals/desires conflicts
Desire conflict	Agents fundamentally desire conflicting things	Make concessions, or influence desires themselves by influencing beliefs
Belief conflict	Agents have different and/or conflicting beliefs about the world	Persuasion

Table 3.1: Types of conflicts and their resolution methods

[Walton and Krabbe, 1995].[9] More specifically, agents need the following main communicative abilities:

- **Bargaining over Deals:**
    - *Propose a deal*: An agent must be able to communicate proposed contracts it is willing to accept as deals.
    - *Accept or reject a proposed deal*
    - *Retract a proposed deal*: An agent must be able to retract a proposal it has made previously. This is important since changes in an agent's preferences may make one of its own earlier proposals no longer acceptable to itself. For example, a buyer who offers to purchase a car for \$10K may retract that offer after finding out a defect in the car.

---

[9] Argumentation theorists Walton and Krabbe [1995] also allow embedding inquiry dialogues in negotiation. This is beyond the scope of this study. It is worth noting, however, that embedding of Walton and Krabbe's dialogues was first formalised by Reed [1998].

- **Seeking and Providing Information:**
    - *Request information*: An agent must be able to ask another agent to communicate its beliefs, desires, preferences and goals.
    - *Make assertions*: An agent must be able to make assertions (and answer questions) about its mental attitudes including its beliefs about the plans for achieving goals and desires.
    - *Decline to answer a question*: Since agents are autonomous, they should not be forced to answer questions, or even participate in the dialogue.
- **Persuasion over Beliefs:**
    - *Challenge assertions*: An agent must be able to ask another agent for reasons behind an assertion made by the latter.
    - *Provide justification for an assertion or assert counter arguments to attack an assertion*: This, combined with the ability to challenge assertions, enables agents to persuade one another to change their beliefs.

3.3.3 Abstract Agent Architecture

Based on the above conceptual analysis, I now provide an abstract description of an agent architecture that has the essential features required for interest-based negotiation.

The architecture is described in Figure 3.5, and is a particular instantiation of the generic ABN agent described in figure 2.2. I briefly discuss some of the architecture features below.

Desire Generation: The agent must be able to generate a set of desires based on its beliefs about the state of the world. And these desires must be "defeasible" in the sense that they can change as a result of receiving new information. One possible approach for implementing this capability, based on default logic, has been proposed by Thomason [2000]. Dastani et al. [2002] proposed another semantic framework, based on dynamic logic, for updating desires based on changes in context.

Plan Generation: Not all desires may be achieved together. This is because there may be sets of desires for which no plan exists (e.g. one may not be able to both write

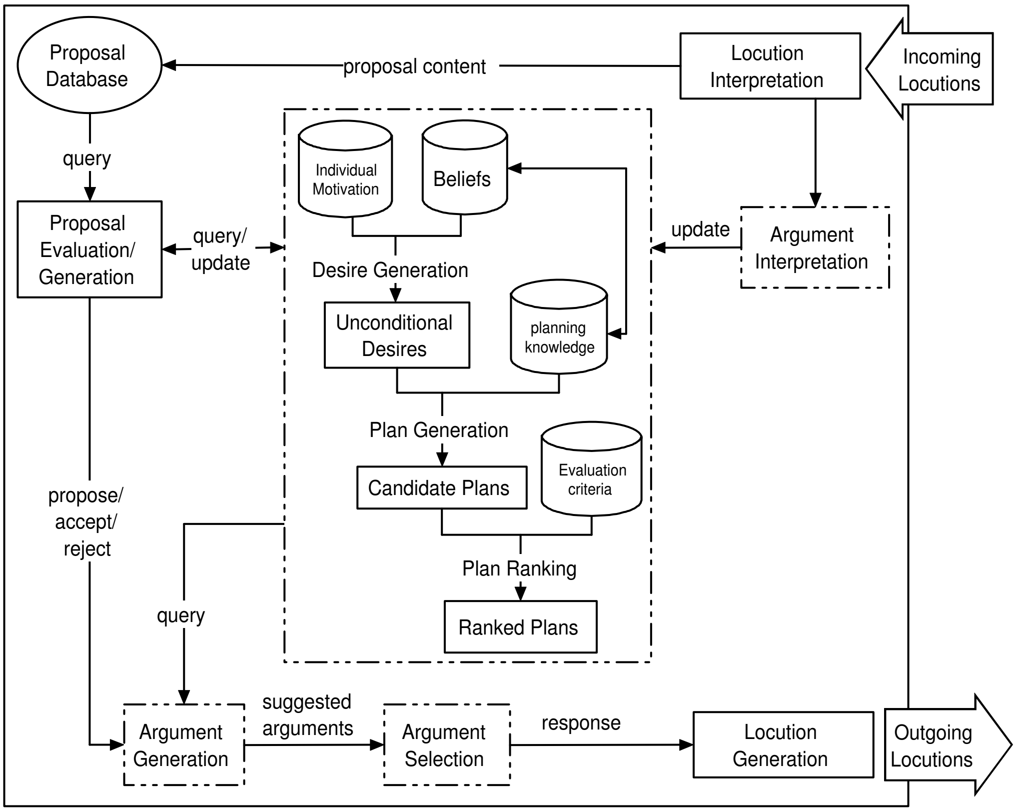


Figure 3.5: Interaction among different agent modules

a paper and enjoy the weekend). Hence, this procedure generates different *candidate plans*, each of which achieves a particular set of desires. There is a wealth of literature on planning in Artificial Intelligence which can be used to encode this capability. One candidate approach is hierarchical-task-network planning [Erol et al., 1994] because it requires explicit representations of abstract goals and hence is suitable for the types of dialogues I presented earlier.

Plan Ranking: This procedure uses some assessment criteria in order to produce a preference ordering over candidate plans. One way to implement this capability is using the notion of *utility* which assigns a number to each plan based on the desires it achieves and the costs of the actions it requires.

Argument Evaluation (or Interpretation): This procedure processes meta-information received from the counterpart. In particular, it decides whether to add or delete beliefs from the belief base and whether to modify its planning knowledge. These changes in beliefs and planning knowledge must then be propagated (through the desire generation and planning processes) in order to update the ranking of plans. In order to address this requirement, we must enable agents to influence each other's beliefs via rational means. Computational models of argument are ideal candidates and they have been already used to enable rich multi-agent dialogues [Amgoud et al., 1998, Maudet and Chaib-draa, 2003, McBurney, 2002]. Some operational frameworks for updating plans in light of new information exist in the literature. A notable framework has been proposed by Pollock [1998] and is grounded in argumentation theory. A more recent attempt to view plan generation in terms of argumentation has been presented by Amgoud [2003].

Argument Generation and Selection: This procedure constitutes the *negotiation strategy* of the agent. It decides what utterance to make based on the agent's current mental state as well as the state of the dialogue. This response may be a proposal, a challenge, a question, an assertion etc. Argument selection is also performed as part of response generation.

The reasoning cycle is abstractly presented in figure 3.6. The agent first generates its desires based on its beliefs and individual internal motivations. Then it generates candidate plans for achieving its desires. It selects the best plan (based on some appropriate criteria), which becomes an intention. If the agent can execute its intention on its own,

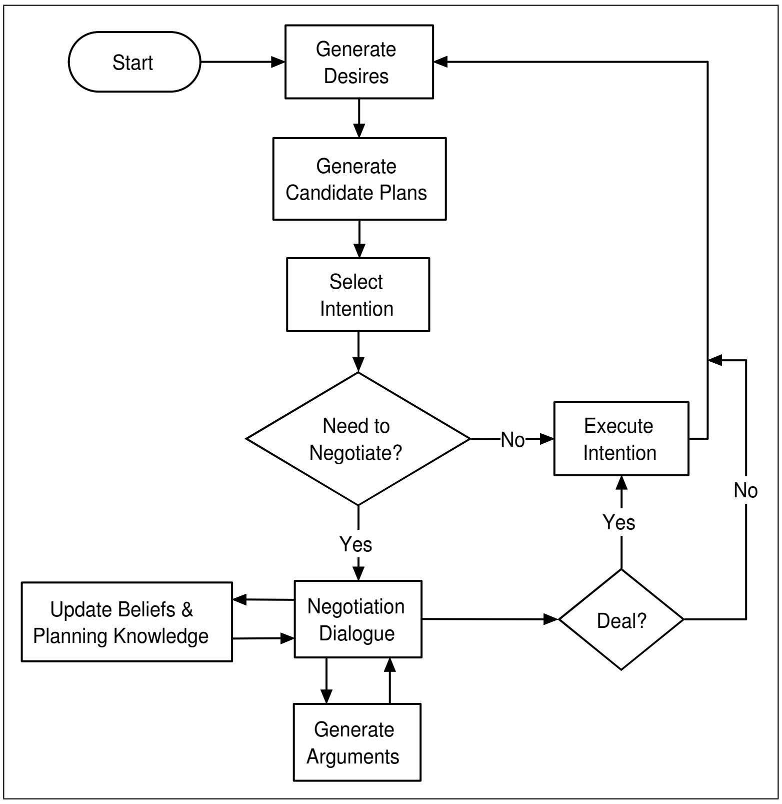


Figure 3.6: Abstract reasoning cycle for IBN Agent

then it would do so. Otherwise, if the agent needs to negotiate with another agent in order to contract out parts of its intention, it would initiate a negotiation dialogue with a particular counterpart. If negotiation results in a deal, the agent can now execute its intention. The agent exits the negotiation dialogue if it decides that no deal is attainable. Until then, during the process of negotiation, the agent may update its beliefs and planning knowledge (as a result of receiving arguments), which could result in updating its desires and intentions. Moreover, the agent may generate arguments to attempt to influence its counterpart. The dialogue should be controlled by some appropriate communication protocol.

The abstract reasoning cycle defined above does not indicate precisely how the agent's beliefs and planning knowledge are updated as a result of receiving arguments. It only indicates that they do change and that this change affects the agent's choices and preferences somehow. In the next section, I present a more precise framework, grounded in argumentation theory, that (i) constitutes a particular formalisation of the abstract concepts described above and (ii) gives precise semantics for the changes in beliefs, desires and intentions as a result of argument exchange. Based on this semantics, I specify a communication protocol that facilitates interest-based negotiation.

3.4 Argumentation Frameworks for IBN Reasoning

Having motivated the need for interest-based negotiation and provided a conceptual framework which contains an informal description of the dialogue features needed, I now present a formal framework for IBN grounded in a specific theory of argumentation.

Broadly speaking, an argument can be seen as "a reason that supports a particular conclusion." The nature of the *reason* and the *conclusion* can vary. For example, the conclusion may be a statement to believe, or an action to execute, or a desire to try to achieve. Similarly, the reason may be an observation, an empirical evidence, or past experience. Given this abstract description, it is possible to use computational models of argumentation to capture various forms of reasoning, by varying the precise meaning and structure of arguments and conclusions, as well as the meaning of the relationships among different arguments. That said, it is worth noting that the use of argumentation to describe non-epistemic (i.e. non-belief-related) reasoning patterns is a recent phenomenon in Artificial Intelligence [Amgoud, 2003, Gordon and Karacapilidis, 1997, Hulstijn and van der Torre, 2004, Kakas and Moraitis, 2003].

It is essential to distinguish between arguing over beliefs and arguing over goals or desires. In argumentation theory, a proposition is believed because it is *acceptable* and *relevant*. Desires, on the other hand, are adopted because they are *justifiable*. A desire is justifiable because the world is in a particular state that warrants its adoption. For example, one might desire to go for a walk because she believes it is a sunny day and may drop that desire if it started raining. Finally, a desire is *achievable* if the agent has an achievable plan that achieves that desire. As a consequence of the different nature of arguments for beliefs and desires, we need to treat them differently, taking into account the different way these arguments relate to one another. For example, a belief argument can be attacked by arguing that it is not consistent with observation, or because there is a reason to believe the contrary. Arguments for desires, on the other hand, could be attacked by demonstrating that the justification of that desire does not hold, or that the plan intended for achieving it is itself not achievable.

To deal with the different nature of the arguments involved, I present three distinct argumentation frameworks: one for reasoning about beliefs, another for arguing about what desires should be pursued, and a third for arguing about the best plan to intend in order to achieve these desires. The first framework is based on existing literature on argumentation over beliefs, originally proposed by Dung [1995] and later extended by Amgoud and Cayrol [2002]. For arguing about desires and plans, I draw on and extend preliminary work on argumentation-based desire generation and planning [Amgoud and Kaci, 2004, Hulstijn and van der Torre, 2004]. I chose this approach as a starting point because, being grounded in argumentation theory, it seems to fit naturally with the dialogical nature of negotiation and hence would provide a natural semantics for dialogues.[10]

3.4.1 Argumentation for Deducing Beliefs

In this section, I present a framework for arguing about beliefs based on the work of Amgoud and Cayrol [2002]. Let $\mathcal{L}$ be a propositional language with $\wedge$ for conjunction, $\rightarrow$ for implication and $\neg$ for negation. Let $\vdash$ denote classical inference and $\equiv$ logical equivalence.

Definition 3. (Basic Beliefs) *An agent's* basic beliefs *is a set* $\mathcal{B}_b = \{(\beta_i, b_i), i = 1, \ldots, n\}$*, where* β_i *is a consistent propositional formula of* $\mathcal{L}$ *and* b_i *its degree of certainty.*[11]

Using beliefs, an agent can construct *belief arguments*, which are defined as follows.

Definition 4. (Belief Argument) [Amgoud et al., 2000]
A belief argument *is a pair* $\langle H, h \rangle$ *such that:*

- $H \subseteq \mathcal{B}_b$ *and* $h \in \mathcal{L}$*;*

- $H \vdash h$*;*

---

[10] Non-argumentation based frameworks for reasoning about and resolving conflicts among beliefs, desires and intentions have also received considerable attention in the agent research community [Dastani et al., 2002, Boutilier, 1994].

[11] This notion of "certainty" was presented in the original framework by Amgoud and Kaci [2004]. It is required in order to generate an ordering over arguments, which is required by the underlying argumentation theory. Therefore, it does not have an (explicit) probabilistic interpretation.

- *H is consistent and minimal (for set inclusion).*

H is often referred to as the *support* of the argument, and h its *conclusion.* I denote by $\mathcal{A}_b$ the set of all possible belief arguments that can be generated from a belief knowledge base $\mathcal{B}_b$.

Definition 5. (Preference among belief arguments) [Amgoud et al., 2000]
Let $A = \langle H, h \rangle \in \mathcal{A}_b$.

- *The* strength level *of A is Level$(A) = \min\{a_i : (\varphi_i, a_i) \in H\}$. If $H = \emptyset$ then Level$(A) = 1$;*
- *Argument A_1 is* preferred *to A_2, denoted $A_1 \succ A_2$ if and only if Level$(A_1) \geq$ Level(A_2).*

The notion of preference between belief arguments is used to resolve conflicts that may arise among them. Conflicts are captured in the notion of attack.

Definition 6. (Attack and Defence among Belief Arguments) [Amgoud et al., 2000]
Let $A_1 = \langle H_1, h_1 \rangle, A_2 = \langle H_2, h_2 \rangle \in \mathcal{A}_b$.

- A_1 undercuts A_2 *if $\exists h'_2 \in H_2$ such that $h_1 \equiv \neg h'_2$*
- A_1 rebuts A_2 *if $h_1 \equiv \neg h_2$.*
- A_1 attacks A_2 *if:*
  1. *A_1 undercuts A_2 and $A_2 \not\succ A_1$; or*
  2. *A_1 rebuts A_2 and $A_2 \not\succ A_1$.*
- *Let $S \subseteq \mathcal{A}_b$. S* defends *A_1 if for every argument $A_2 \in \mathcal{A}_b$ where A_2 attacks A_1, there is some argument A_3 in S which attacks A_2.*

Having defined the basic concepts, the complete process of generating new beliefs can be described using a fixed-point characterisation of argumentation, due to Dung [1995].

Definition 7. (Belief Argumentation) [Amgoud et al., 2000]
Let $\mathcal{A}_b$ be the set of all belief arguments generated from a set of basic beliefs $\mathcal{B}_b$. Let $\succ$ be a preference relation over $\mathcal{A}_b$ induced by the certainty degrees associated with $\mathcal{B}_b$. We have:

- *The set* $Acc(\mathcal{A}_b)$ *of* acceptable belief arguments *is defined as:*

$$Acc(\mathcal{A}_b) = \mathcal{F}_{i \geq 0}(\emptyset) \textit{ where } \mathcal{F}(\mathcal{S}) = \{A \in \mathcal{A}_b : A \textit{ is defended by } \mathcal{S}\}$$

- *A proposition is believed if it is a member of the set of acceptable belief arguments.*

The argumentation framework I have just described enables us to specify how an agent's beliefs can be updated as a result of receiving such new arguments. As new arguments are received, the set of acceptable arguments is re-evaluated and the new set reflects the updated beliefs.[12]

3.4.2 Argumentation for Generating Desires

Amgoud and Kaci [2004] introduced *explanatory arguments* as a means for generating desires from beliefs. I extend their framework in this section and refine it in order to resolve some problematic features caused by the fact that they combine belief argumentation with desire argumentation in a single framework.[13]

I begin by presenting the concept of a *desire generation rule*. I extend the framework of Amgoud and Kaci [2004] by introducing the modality d, where $\mathsf{d}(\varphi)$ means that the agent desires the literal φ. A *conflict between desires* occurs when the agent has two desires $\mathsf{d}(\varphi)$ and $\mathsf{d}(\neg\varphi)$.[14]

Definition 8. (Desire Generation Rules) *A* desire generation rule *(or a* desire rule*) is an expression of the form* $C \Rightarrow \mathsf{d}(\varphi)$ *where* φ *is a literal of* $\mathcal{L}$ *and either:*

- $C = \phi$*; or*
- $C = \varphi_1 \wedge \cdots \wedge \varphi_n \wedge \mathsf{d}(\psi_1) \wedge \cdots \wedge \mathsf{d}(\psi_m)$ *where each* φ_i *and* ψ_i *is a literal of* $\mathcal{L}$*.*

A desire generation rule ($C \Rightarrow \mathsf{d}(\varphi)$) expresses that φ is desired in context C. If $C = \phi$, then the desire $\mathsf{d}(\varphi)$ is unconditional. Otherwise, the distinction between belief expressions and desire expressions enables us to express desires that follow from one another.

---

[12] A proof-theory based on a two-agent dialogues, and faithful to the fixed-point semantics, has been presented by Amgoud and Cayrol [2002].

[13] Note that in my work, as well as much other work on agent logics, desires are denoted by propositions, i.e. they have truth-functional denotations. Whatever problems this may pose are philosophical and beyond the scope of this thesis.

[14] For simplicity, I do not allow for expressions of the form $\neg\mathsf{d}(\varphi)$ for "the agent does not desire φ."

Therefore, the meaning of the rule is: if the agent *believes* the literals $\varphi_1, \ldots, \varphi_n$ and *desires* the literals $\psi_1, \ldots, \psi_m$, then the agent will *desire* φ as well.

Each agent has a set of desire generation rules which comprise its "desire generation knowledge-base."

Definition 9. (Desire Generation Knowledge Base) *A* desire generation knowledge-base *is a set* $\mathcal{B}_d = \{(C \Rightarrow \mathsf{d}(\varphi_i), w_i), i = 1, \ldots, n\}$*, where* $(C \Rightarrow \mathsf{d}(\varphi_i))$ *is a desire generation rule and* w_i *is the* worth *(or degree of desirability) of* φ_i*. I will sometimes use Worth*(φ_i) *to denote* w_i*.*

Using desire generation rules, we can characterise *potential desires*.[15]

Definition 10. (Potential Desire) *Suppose an agent is equipped with a desire generation knowledge base* $\mathcal{B}_d$*. The set of* potential desires *is:* $\mathcal{PD} = \{\varphi : \exists(C \Rightarrow \mathsf{d}(\varphi)) \in \mathcal{B}_d\}$

These are "potential" desires because the agent does not know yet whether the antecedents (i.e. bodies) of the corresponding rules are true. To reason about the truth of these antecedents, the agent constructs *explanatory arguments*, arguments that justify desires. These arguments make use of an extended logical implication symbol $\vdash_d$ which makes use of both classical inference as well as forward inference using desire generation rules to infer *tentative desires* $\mathcal{TD}$. More formally, given desire generation rule $(\varphi_1 \wedge \cdots \wedge \varphi_n \wedge \mathsf{d}(\psi_1) \wedge \cdots \wedge \mathsf{d}(\psi_m) \Rightarrow \mathsf{d}(\varphi))$, then if the agent believes $\varphi_1, \ldots, \varphi_n$ and has desires $\mathsf{d}(\psi_1), \ldots, \mathsf{d}(\psi_m) \in \mathcal{TD}$, then $\mathsf{d}(\varphi)$ will be added to $\mathcal{TD}$.

Definition 11. (Explanatory Argument)

An explanatory argument *is a pair* $\langle H, h \rangle$ *such that:*

- $H \subseteq \mathcal{B}_b \cup \mathcal{B}_d$*;*
- $H \vdash_d h$*;*
- H *is consistent*[16] *and minimal (for set inclusion).*

Note that in my treatment, I do not provide a systematic axiomatisation of conditions over modalities. For example, the language I offer does not allow expressions such as

---

[15] Amgoud and Kaci [2004] refer to these as "potential initial goals."

[16] I.e. there is no p and $\neg p$, and there is no $\mathsf{d}(\varphi)$ and $\mathsf{d}(\neg\varphi)$ in H.

$\mathsf{d}(\neg x) \rightarrow \neg\mathsf{d}(x)$. Similarly, the language does not have a belief modality b and hence cannot express statements such as $\mathsf{d}(x) \rightarrow \mathsf{b}(\mathsf{d}(x))$ (for "if the agent desires x then it must believe that it desires x"). While these widely explored issues [Wooldridge, 2000] are indeed relevant to negotiation (e.g. in strategic reasoning), investigating them is outside the scope of this thesis. I do attempt, however, to overcome some of the limitations of the original framework by Amgoud and Kaci [2004]. Note that in the above definition, we have $h \in \mathcal{PD}$, and hence h is of the form $\mathsf{d}(x)$. This contrasts with Amgoud and Kaci [2004], who do not distinguish between desires and beliefs in their desire generation rules (since there is no desire modality). This is problematic as it leads to incorrect inferences where an agent may conclude beliefs on the basis of its yet-unachieved desires, hence exhibiting a form of wishful thinking. To overcome this problem, I do not allow beliefs to follow from desires.

When there is an explanatory argument for some desire, we say that the desire is *justified* by that argument. $\mathcal{A}_d$ denotes the set of all explanatory arguments that can be generated from a given knowledge base. We have that $\mathcal{A} = \mathcal{A}_d \cup \mathcal{A}_b$.

The following example shows how an agent constructs a desire using a desire generation rule.

Example 10. *Let:*

waic = "there is a relevant workshop at the Sydney AI conference";
aic = "attend the Sydney AI conference";

Suppose we have the following belief bases:

1. $\mathcal{B}_b = \{(waic, 1)\}$
2. $\mathcal{B}_d = \{(waic \Rightarrow \mathsf{d}(aic), 0.6)\}$

The agent can construct an explanatory argument in favour of the desire to attend the Sydney AI conference: $\langle\{waic, waic \Rightarrow \mathsf{d}(aic)\}, \mathsf{d}(aic)\rangle$

Note that the above example involves a desire generation rule that contains beliefs only in its body. The following extended example shows how a desire can follow from another, already generated *tentative* desire.

Example 11. *Extending example 10, let:*

keynote = “interesting key note speech”;

attendkey = “attend the key note speech”;

Suppose we have the following additional desire generation rule, which states that if there is an interesting keynote speech at a conference I already desire to attend, then I would also desire to attend that speech:

3. $(\mathit{keynote} \wedge \mathsf{d}(\mathit{aic}) \Rightarrow \mathsf{d}(\mathit{attendkey}), 0.8)$

The agent can construct an explanatory argument for the desire to attend the keynote speech: $\langle\{\mathit{waic}, \mathit{waic} \Rightarrow \mathsf{d}(\mathit{aic}), \mathit{keynote}, \mathit{keynote}\wedge\mathsf{d}(\mathit{aic}) \Rightarrow \mathsf{d}(\mathit{attendkey})\}, \mathsf{d}(\mathit{attendkey})\rangle$

As with belief arguments, explanatory arguments may conflict. Therefore, we also require preferences to help resolve such conflicts. The notion of strength of an explanatory arguments is used (see below) when there are conflicts among arguments for different desires. Strength provides a criteria for selecting what to desire when there are conflicting reasons to desire different things. Moreover, strength enables an explanatory argument to defend itself against an attacking belief argument, as I shall explain subsequently.

Definition 12. (Preference among explanatory and belief arguments) [Amgoud and Kaci, 2004]
Let $A = \langle H, h\rangle \in \mathcal{A}$ *be an explanatory argument or a belief argument.*

- *The* strength level *of* A *is* $\mathit{Level}(A) = \min\{a_i : (\varphi_i, a_i) \in H \cap \mathcal{B}_b\}$. *If* $H \cap \mathcal{B}_b = \emptyset$ *then* $\mathit{Level}(A) = 1$*;*
- *Argument* A_1 *is* preferred *to* A_2*, denoted* $A_1 \succ A_2$ *if and only if* $\mathit{Level}(A_1) \geq \mathit{Level}(A_2)$.

Note that the *worth* of desires in the argument premise is not taken into account when calculating the strength of the argument. Only the certainty of underlying beliefs is considered (i.e. strength is calculated exactly as with belief argumentation). The intuition is that, in case of conflict between two desires, the agent should adopt the desire with the stronger *objective* supporting beliefs. Once a desire is justified this way, it would be adopted with the given *subjective* worth. In other words, in order to avoid wishful thinking, an agent should not prefer the desire with higher worth if it is justified by weaker beliefs. Note that this handling is different from that of decision-theoretic expected utility, whereby the probabilistic likelihood of an outcome is combined with the utility received from that outcome. In the current focus on desire generation, we are not dealing with the likelihood of outcomes, but rather with the strength of beliefs about the current state of the world, and how this may affect desires. How this may relate to expected-utility is beyond the scope of this work.

Attacks against explanatory arguments have a different interpretation from those among belief arguments. An explanatory argument for some desire can be undercut either by a belief argument (which undermines the truth of the underlying belief justification), or by another explanatory argument (which undermines one of the existing desires the new desire is based on). A rebutter, on the other hand, can only be another explanatory argument, which provides a reason to desire the opposite. These intuitions are clarified in the definition below and subsequent example.

Definition 13. (Attack and Defence among Explanatory and Belief Arguments) *Let* $A_1 = \langle H_1, h_1 \rangle$, $A_2 = \langle H_2, h_2 \rangle$, $A_3 = \langle H_3, h_3 \rangle$. *Suppose that* $A_1, A_2 \in \mathcal{A}_d$ *and that* $A_3 \in \mathcal{A}_b$.

- A_3 b-undercuts A_2 *if* $\exists h'_2 \in H_2 \cap \mathcal{B}_b$ *such that* $h_3 \equiv \neg h'_2$*;*
- A_1 d-undercuts A_2 *if* $\exists h'_2 \in H_2 \cap \mathcal{PD}$ *such that* $h_1 = \mathsf{d}(p)$ *and* $h'_2 = \mathsf{d}(\neg p)$*;*[17]
- A_1 d-rebuts A_2 *if* $h_1 = \mathsf{d}(p)$ *and* $h_2 = \mathsf{d}(\neg p)$.
- *An argument* $A' \in \mathcal{A}$ attacks $A_2 \in \mathcal{A}_d$ *if:*
    1. A' *b-undercuts or d-undercuts* A_2 *and* $A_2 \not\succ A'$*; or*
    2. A' *d-rebuts* A_2 *and* $A_2 \not\succ A'$.
- *Let* $S \subseteq \mathcal{A}$. S defends A_1 *if for every argument* $A' \in \mathcal{A}_b \cup \mathcal{A}_d$ *where* A' *attacks* A_1*, there is some argument* A'' *in* S *which attacks* A'.

Note that the notion of defence described in definition 13 also uses attacks among belief arguments. This enables belief arguments to defend explanatory arguments against other

---

[17] The reader may have noticed that $\neg\mathsf{d}(p)$ is not a well formed formula, hence it is not possible to express "not-desiring p" or, for that matter, complex nesting of various BDI modalities. Such important distinctions are worthy of separate treatment, such as that based on modal logic [Wooldridge, 2000].

belief arguments. Moreover, the above notion of argument advances that presented by Amgoud and Kaci [2004] to resolve a problematic issue. Since Amgoud and Kaci do not have an explicit desire modality, a belief argument for "I am not at the beach" can rebut an explanatory argument for "I desire to be at the beach." This is inappropriate since we want to allow an agent to desire things that are not believed to be true.

The following example illustrates the above concepts.

Example 12. *(Builds on example 10) Recall that:*

waic = "there is a relevant workshop at the Sydney AI conference";

aic = "attend the Sydney AI conference";

$\langle\{waic, waic \Rightarrow \mathsf{d}(aic)\}, \mathsf{d}(aic)\rangle$

The argument for going to the Sydney AI conference can be undercut by arguing that the workshop has been cancelled: $\langle\{wcancel, wcancel \rightarrow \neg waic\}, \neg waic\rangle$. *It can be rebutted by arguing that one should not desire to attend the conference because it is not of international standing:* $\langle\{\neg intl, \neg intl \Rightarrow \mathsf{d}(\neg aic)\}, \mathsf{d}(\neg aic)\rangle$.

The argument for desire "aic" can also be defended. It can be defended against the undercutter by, for example, arguing that the workshop website is still on and therefore it was not cancelled: $\langle\{webinfo, webinfo \rightarrow \neg wcancel\}, \neg wcancel\rangle$. *The argument above can also be defended against its rebutter by rebutting the rebutter itself, e.g. by presenting another (explanatory) argument which states that the conference is still worth attending because it has a doctoral symposium:* $\langle\{symp, symp \Rightarrow \mathsf{d}(aic)\}, \mathsf{d}(aic)\rangle$. *It can also be defended against the rebutter by undercutting the rebutter, e.g. by arguing that the conference is in fact an international conference since it was held in the UK last year:* $\langle\{aicUK, aicUK \rightarrow intl\}, intl\rangle$. *This undercutter can then be counter-attacked and so on.*

Figure 3.7 summarises the notions of attack described so far. Belief arguments attack one another by the common undercuts and rebuts. Explanatory arguments can be undercut by belief or explanatory arguments, but can only be rebutted by explanatory arguments.

Having defined the basic concepts, the complete process of generating desires can be described using a fixed-point characterisation of argumentation.

Definition 14. (Desires) *Let* $\mathcal{A}$ *be the set of all belief arguments and explanatory arguments generated from a set of basic beliefs* $\mathcal{B}_b$ *and desire generation knowledge-based*

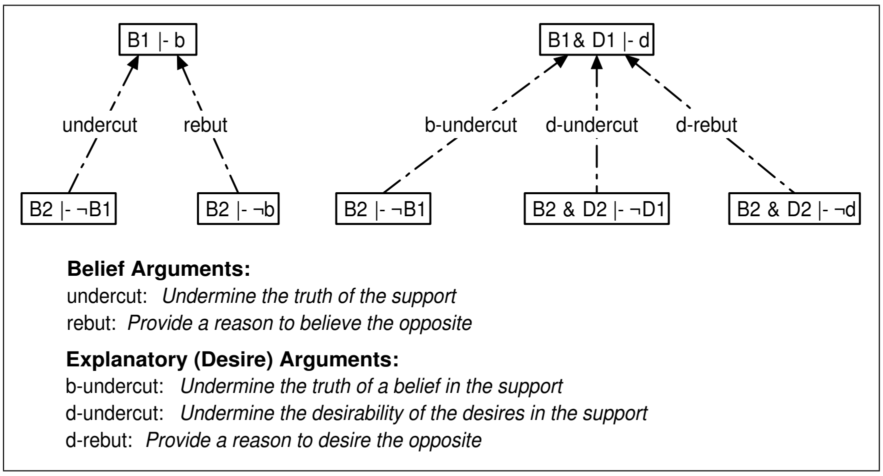


Figure 3.7: Summary of attacks involving belief and explanatory arguments

$\mathcal{B}_d$. *Let* $\succ$ *be a preference relation over* $\mathcal{A}$ *induced by the certainty degrees associated with* $\mathcal{B}_b$.

- *The set* $Acc(\mathcal{A})$ *of* acceptable explanatory and belief arguments *is defined as:*[18]

$$Acc(\mathcal{A}) = \mathcal{F}_{i \geq 0}(\emptyset) \text{ where } \mathcal{F}(\mathcal{S}) = \{A \in \mathcal{A} : \ A \text{ is defended by } \mathcal{S}\}$$

- *The set* $Acc(\mathcal{A}_d)$ *of* acceptable explanatory arguments *is defined as:* $Acc(\mathcal{A}_d) = Acc(\mathcal{A}) - Acc(\mathcal{A}_b)$

- *The agent's* desires, *denoted* $\mathcal{D}$, *are the set of potential desires which are supported by acceptable explanatory arguments. Formally,* $\mathcal{D} \subseteq \mathcal{PD}$ *such that* $\forall d \in \mathcal{D}, \exists \langle H, d \rangle \in Acc(\mathcal{A})$.

Desires supported by acceptable explanatory arguments are justified and hence the agent will pursue them (if they are achievable).

3.4.3 Argumentation for Planning

In the previous subsection, I presented a framework for arguing about desires and producing a consistent set of desires. I now present an argumentation-based framework for generating non-conflicting plans for achieving a (sub)set of such desires.

---

[18] Note that this includes all acceptable belief arguments.

Let *RES* be a set of resources available in the system. I denote by $\mathcal{R}^i \subseteq \mathit{RES}$ the set of all resources agent i possesses. Agent i can use another agent j's resource $r \in \mathcal{R}^j$ in order to achieve its desires. But to do so, j must agree to *commit* resource r to agent i. Finally, I denote by $\mathcal{R}^{i,j}$ the set of *shared resources* between agents i and j. No single agent can unilaterally access shared resources and hence both agents must agree on their allocation. Finally, we have a function $\mathit{Cost} : \mathit{RES} \rightarrow \mathbb{R}$ which returns the cost of using a particular resource.

I now define the notion of *planning rule*, which is the basic building block for specifying plans.

Definition 15. (Planning Rules) *A* planning rule *is an expression of the form* $(\varphi_1 \wedge \cdots \wedge \varphi_n \rightarrowtail \varphi)$ *where each* $\varphi_i \in \mathcal{L} \cup \mathit{RES}$ *and* $\varphi \in \mathcal{L}$.

A planning rule expresses that if $\varphi_1 \wedge \cdots \wedge \varphi_n$ (the plan "body") is achieved, then φ (the plan "head") is achieved.[19] If φ_i is a literal in $\mathcal{L}$, it can be achieved by applying another planning rule of the form $(\varphi_i^1 \wedge \cdots \wedge \varphi_i^m \rightarrowtail \varphi_i)$ and achieving each of $\varphi_i^1, \ldots \varphi_i^m$. If, on the other hand, $\varphi_i \in \mathit{RES}$ (e.g. money), then φ_i can be achieved by consuming the resource (e.g. spending the money). Each agent has a set of planning rules it is aware of, which constitute its "planning knowledge base."

Definition 16. (Planning Knowledge Base) *A* planning knowledge-base *is a set* $\mathcal{B}_p = \{\mathit{PR}_i, i = 1, \ldots, n\}$, *where* PR_i *is a planning rule.*

The basic building block of a plan is the notion of "partial plan," which corresponds to a planning rule.

Definition 17. (Partial Plan) *A* partial plan *is a pair* $[H, h]$ *such that either:*

- $h \in \mathcal{L}$ *and* $H = \{\varphi_1, \ldots, \varphi_n\}$ *such that* $\exists(\varphi_1 \wedge \cdots \wedge \varphi_n \rightarrowtail h) \in \mathcal{B}_p$; *or*
- $h \in \mathit{RES}$ *and* $H = \emptyset$

A partial plan of the form $[\emptyset, \varphi]$ is called an *elementary partial plan* or (equivalently) an *atomic action*, since it can be achieved directly using a resource. I sometimes add the

---

[19] Note that the implications defined in desire generation rules and planning rules are not material. So for example, from $\neg y$ and $x \rightarrowtail y$, we cannot deduce $\neg x$.

syntactic predicate $\mathbf{do}(.)$ to distinguish elementary partial plans, which are then written $[\emptyset, \mathbf{do}(\varphi)]$.

Definition 18. (Instrumental Argument, or Complete Plan) *An* instrumental argument *is a pair $\langle G, d\rangle$ such that $d \in \mathcal{L}$, and G is a finite tree such that:*

- *the root of the tree is a partial plan $[H, d]$;*
- *each node $[\{\varphi_1 \wedge \cdots \wedge \varphi_n\}, h']$ has exactly n children $[H'_1, \varphi_1], \ldots [H'_n, \varphi_n]$, where each $[H'_i, \varphi_i]$ is a partial plan for φ_i;*
- *the leaves of the tree are elementary partial plans.*

I define a function *Nodes*(G), which returns the set of all partial plans of tree G, and a function *Resources*(G), which returns the set of all resources needed to execute G (i.e. the union of single resources needed for each leaf nodes in G). We can now encode example 7 presented informally above.

Example 13. *(Formalises example 7)*
Let:

waic = "there is a relevant workshop at the Sydney AI conference";
aic = "attend the Sydney AI conference";
syd = "go to Sydney";
reg = "pay conference registration";
rent = "rent a car";
ford = "get a particular car of make Ford";
pay\$100 = "pay \$100";
pay\$200 = "pay \$200";[20]

We can now specify the following, for the buyer agent B and seller agent S:

1. $\mathcal{B}_b^B = \{(\mathit{waic}, 1)\}$
2. $\mathcal{B}_d^B = \{(\mathit{waic} \Rightarrow d(\mathit{aic}), 0.6)\}$

---

[20] Realistically, one requires a more elaborate treatment of actions, e.g. the agent must also be able to pay \$300, or pay \$50 six times. For simplicity, I suffice with these illustrative unique actions.

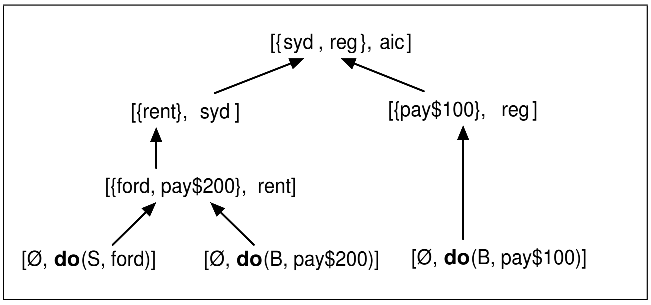


Figure 3.8: Complete plan for example 13

3. $\mathcal{B}_p^B = \begin{cases} syd \wedge reg & \rightarrowtail aic \\ rent & \rightarrowtail syd \\ ford \wedge pay\$200 & \rightarrowtail rent \\ pay\$100 & \rightarrowtail reg \end{cases}$

4. $RES = \{pay\$100, pay\$200, ford\}$

5. $\mathcal{R}^B = \{pay\$100, pay\$200\}$

6. $\mathcal{R}^S = \{ford\}$

Figure 3.8 shows an instrumental argument, for attending the Sydney AI conference, that agent B can construct using the above information. Note that this plan involves the execution of action *ford* by agent S, because B does not have "*ford*" as one of its resources. Without getting the car from S, B cannot make it to Sydney using this plan.

A qualification regarding my use of the term "plan" is necessary here. Planning is a substantial and well-developed area in AI [Erol et al., 1994, Kambhampati, 1995, Georgeff, 1987]. My aim here is not to propose a novel planning framework and it is clear that the notion of "plan" described above does not handle temporal order or complex interactions among actions. Instead, the notion of "instrumental argument" is intended to capture dependencies of interest between low-level actions/goals and higher-level goals/desires for the sake of enabling dialogue about these.

As with belief and explanatory arguments, I now present the notion of an *acceptable set of instrumental arguments*. I adopt a different definition from those presented by Amgoud [2003] and Hulstijn and van der Torre [2004], as will be clarified below. My definition requires a syntactic distinction of the part of beliefs that denotes material implication rules. Hence, I use the symbol $\mathcal{B}_b^{\rightarrow}$ to refer to the part of the agent's beliefs that includes material implication rules (e.g. $p \rightarrow q$) and does not include propositional facts about the world (e.g. p, $q \wedge r$ etc.). Let $\vdash_p$ denote an extended logical implication which makes use of both classical inference as well as forward inference using planning rules.

Definition 19. (Acceptable Set of Instrumental Arguments) *Let* $S = \{\langle G_i, d_i\rangle i = 1, \ldots, n\} \subseteq \mathcal{A}_p$*.* S *is* acceptable *if:*

- $\bigcup_{\langle G,d\rangle \in S}[\bigcup_{[H,h]\in \mathit{Nodes}(G)}(H \cup \{h\})] \cup \mathcal{B}_b^{\rightarrow} \cup \mathcal{B}_p \nvdash_p \perp$.
- $\forall \langle G_k, d_k\rangle, \langle G_l, d_l\rangle \in S$*, we have that* $d_k = d_l$ *implies* $G_k = G_l$*.*
- $\bigcup_{\langle G,d\rangle \in S} \mathit{Resources}(G_i) \subseteq \mathcal{R}^i \cup \mathcal{R}^j \cup \mathcal{R}^{i,j}$ *where* j *is another agent in the system.*

In words: (i) I first require that all plan nodes are logically consistent with one another and that the agent believes their consequences are consistent. By "consequences" I mean both their *logical* consequences, as derived by material implication rules, as well as their *physical* consequences, as derived by applying additional planning rules. This differs from the definition presented by Amgoud [2003] which requires that plans are consistent with *all* of the agent's beliefs $\mathcal{B}_b$. Amgoud's definition implies that agents cannot intend to achieve things they believe are *now* untrue. Nevertheless, my definition of conflict suffers from the lack of temporality, which is also a limitation in the original framework by Amgoud [2003]. It assumes that the order of executing actions is not significant and hence does not allow conflict avoidance by reordering actions. I stick to this, however, to avoid non-trivial temporal issues.

(ii) Second, definition 19 excludes redundant alternative plans for the same desire. This ensures that each set of acceptable instrumental arguments contains at most one plan per desire.

(iii) Finally, in definition 19 I require that there are sufficient resources in the system to execute all plans. If $\mathit{Resources}(G_i) \subseteq \mathcal{R}^i$, we say that agent i can individually execute G_i. If, on the other hand, we have $\mathit{Resources}(G_i) \subsetneq \mathcal{R}^i$ but at the same time $\mathit{Resources}(G_i) \subseteq \mathcal{R}^i \cup \mathcal{R}^j \cup \mathcal{R}^{i,j}$, then the agent can try to secure the required resources through agreement with j. If such agreement is not reached, the set of plans is no longer acceptable. Here is an example to illustrate this point. In example 13, if the agent's resources are $\mathcal{R}^B = \{\mathit{pay\$200}, \mathit{pay\$100}, \mathit{ford}\}$, then it would be able to execute the plan individually. But since $\mathcal{R}^B = \{\mathit{pay\$200}, \mathit{pay\$100}\}$, the buyer needs some other agent that possesses the resource "*ford*" (e.g. a travel agent) in order to execute its plan.

The different acceptable sets of instrumental arguments denote possible intentions the agent might adopt. We are interested in acceptable sets of instrumental arguments that are "maximal" in some sense. To deal with this, I define a notion of optimal sets of instrumental arguments, taking into account the *utilities* of the plans involved, as defined below.

Definition 20. (Weight –or Utility– of Instrumental Argument) *Let $A = \langle G, g\rangle$ be a complete plan. And let $Des(A, \mathcal{D})$ be the set of desires from the set $\mathcal{D}$ that plan A achieves. The* utility *of A to agent i is $Utility^i(A) = \sum_{k=1}^{n} Worth(d_k) - \sum_{l=1}^{m} Cost(r_l)$, where $\{d_1, \ldots, d_n\} = Des(A, \mathcal{D}^i)$ and $\{r_1, \ldots, r_m\} = Resources(G)$.*

In other words, the utility of an instrumental argument for some agent is the difference between the worths of agent i's desires that the plan achieves and the cost of the resources needed to execute the plan.[21] From the utilities of instrumental arguments, we can construct a complete partial ordering $\succeq_p$ over $\mathcal{A}_p$ (the set of all plans). Based on this ordering, we can select optimal sets as candidate intentions.[22]

Definition 21. (Candidate Intention)

Let $\mathcal{A}_p$ be the set of all possible instrumental arguments that can be generated from a planning knowledge base $\mathcal{B}_p$ and supporting elements of a desire set $\mathcal{D}$. And let $S \subseteq \mathcal{A}_p$ be an acceptable set of instrumental arguments. We say that S is a candidate intention *if*

---

[21] Amgoud and Kaci [2004] describe the weight of an instrumental argument (i.e. complete plan) in terms of the worth of the desire it achieves. Their definition does not account for the cost of executing the plan.

[22] Amgoud [2003] uses the notion of "preferred extensions" in order to identify the preferred set of instrumental arguments. This approach not suitable to capture planning dependencies since it gathers the extensions with the maximum *number* of arguments, not the maximum number of desires. Hulstijn and van der Torre [2004] deal with this problem by excluding alternative plans for the same desire, but still aim at a maximal number of desires, with no regard to their priority or the cost of different plans. The framework presented here considers priority and cost and hence allows for cases where a single desire/plan pair can be more preferred –in terms of utility– to two or more desire/plan pairs. More on this can be found in Appendix A.

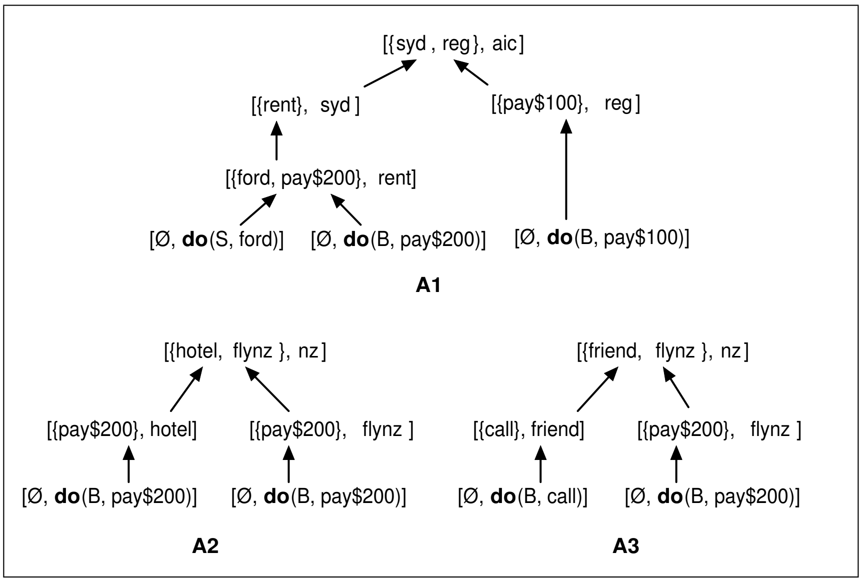


Figure 3.9: Plans for example 14

S is maximal with respect to the sum of plan utilities.

If there is more than one candidate intention, a single one must be selected (e.g. at random) to become the agent's *intention*. The chosen intention is denoted by $\mathcal{I}$. The agent's *intended desires* $\mathcal{ID} \subseteq \mathcal{D}$ are those desires achievable through the intention; i.e. desires that are both justifiable (via acceptable explanatory arguments) and achievable (via an acceptable set of instrumental arguments). Finally, the *intended resources*, denoted $\mathcal{IR} \subseteq$ *RES* denote the resources needed by plans in $\mathcal{I}$.

I now give an example, depicted in Figure 3.9, to put the above concepts together.

Example 14. *(Extends example 7)*

Suppose the buyer also would like to go on holiday to New Zealand and must reason with a limited budget. Let:

nz = "take a holiday in New Zealand";

flynz = "fly to New Zealand";

hotel = "book a hotel accommodation";

friend = "stay at a friend's place";

call = "call a friend";

Suppose the agent has the following new desire generation knowledge base: $\mathcal{B}_d^B = \{(waic \Rightarrow aic, 0.6), \Rightarrow nz, 0.5)\}$ *and that both desires are justified. Finally, suppose costs* *are assigned as follows: Cost(pay\$200)* $= 0.2$*, Cost(pay\$100)* $= 0.1$*, Cost(pay\$200)* $=$ 0.2*, Cost(call)* $= 0$*, Cost(ford)* $= 0$*).*[23]

Suppose the buyer has two instrumental arguments for going to New Zealand: one requires booking a hotel (and paying \$150), while the other involves calling a friend to arrange a stay at his place. There are no conflicts between arguments A_1 *and* A_2*, or between* A_1 *and* A_3*. However, arguments* A_2 *and* A_3 *attack each other because they are alternatives for the same desire. Hence, we have two candidate intentions:* $\{A_1, A_2\}$ *and* $\{A_1, A_3\}$*. But we have:*

- *Utility*$(A_1) = 0.6 - (0.2 + 0.1) = 0.3$,
- *Utility*$(A_2) = 0.5 - (0.2 + 0.2) = 0.1$,
- *Utility*$(A_3) = 0.5 - (0.2) = 0.3$,

The agent's intention will be $\{A_1, A_3\}$ *because it is the only candidate intention that maximises the total utility of plans involved, achieving a total utility of* $0.3 + 0.3 = 0.6$.

3.5 Conflict among Agents: Revisited

So far, I used argumentation to specify single-agent reasoning about desires and plans. In this section, I use the argumentation frameworks presented above in order to give a precise account of the types of conflicts discussed in section 3.3.1.

Suppose we have two agents x and y, each endowed with a set of beliefs and resources denoted by $\mathcal{B}^i_b$, $\mathcal{B}^i_d$, $\mathcal{B}^i_p$ and $\mathcal{R}^i$, where $i \in \{x, y\}$. Suppose each agent has generated a set of desires $\mathcal{D}^i$ among which it has intended desires $\mathcal{ID}^i \subseteq \mathcal{D}^i$ to be achieved using intention $\mathcal{I}^i = \{A^i_{p,1}, \dots, A^i_{p,n}\}$ where each $A^i_{p,k}$ is a complete plan. Finally, let *Goals*$(A^i_{p,k})$ denote the goals of plan $A^i_{p,k}$, i.e. the set of nodes of instrumental argument $A^i_{p,k}$ for which the conclusion is not in D^i or *RES*. We can distinguish the following types of conflicts between agents.

---

[23] The cost of "*ford*" to the buyer is zero because this resource is possessed by the seller and hence would only incur a cost to the seller. It may only *indirectly* cost the buyer if he provides a resource in return.

Desire-Desire Conflict: This conflict occurs when agents' intended desires are contradictory.[24]

$$\mathit{DDConflict}(i,j) \equiv \exists h^i \in \mathcal{ID}^i \text{ and } \exists h^j \in \mathcal{ID}^j \text{ such that } \{h^i, h^j\} \cup \mathcal{B}_b^{i\rightarrow} \vdash \bot \tag{3.1}$$

Desire-desire conflicts are situations where agents ultimately want contradictory things. For example, a pregnant woman may desire to bear a boy while her husband desires a daughter. There is no way that both their desires can be achieved simultaneously. Only one agent can achieve his/her desire and the other must concede or be persuaded to change his/her desire.

Goal-Goal Conflict: This occurs if the agents' intentions conflict but their desires do not. In other words, the agents' desires are consistent, but the means for achieving these desires are in conflict.

$$\begin{aligned} \mathit{PPConflict}(i,j) \equiv\ & \exists [H^i, h^i] \in \mathit{Goals}(A^i_{p,1}) \textit{ where } A^i_{p,1} \in \mathcal{I}^i \text{ and} \\ & \exists [H^j, h^j] \in \mathit{Goals}(A^j_{p,1}) \textit{ where } A^j_{p,1} \in \mathcal{I}^j \text{ such that} \\ & \{h^i, h^j\} \cup H^i \cup H^j \cup \mathcal{B}_b^{i\rightarrow} \cup \mathcal{B}_p^i \vdash_p \bot; \end{aligned} \tag{3.2}$$

An example of a goal-goal conflict is a situation in which an academic wants to attend a conference, and a student desires to get some feedback on his thesis and wants to achieve that by holding a meeting with the academic. Even though no desire-desire conflicts or physical resource-resource conflicts exist, by attending the conference, the academic can no longer hold a supervision meeting with his student. That is, the goals of the student and supervisor conflict.

Agents may resolve goal-goal conflicts either by finding alternative plans to avoid the conflict (e.g. the academic provides feedback by email), or, as above, by influencing each other's desires themselves such that no conflicting plans are needed (e.g. the academic no longer desires to attend the conference).

---

[24] Note that conflicts here are defined from the perspective of the first agent i.

Goal-Desire Conflict: This happens if one agent's plan conflicts with another agent's desire.

$$\begin{aligned} \mathit{PDConflict}(i,j) \quad \equiv \quad & \exists [H^i, h^i] \in \mathit{Goals}(A^i_{p,1}) \ \mathit{where} \\ & A^i_{p,1} \in \mathcal{I}^i \text{ and } \exists h^j \in \mathcal{ID}^j \text{ such that} \\ & H^i \cup \{h^i\} \cup \{h^j\} \cup \mathcal{B}^{i\rightarrow}_b \cup \mathcal{B}^i_p \vdash_p \perp \end{aligned} \tag{3.3}$$

An example of this sort of conflict is a situation where the academic (described above) simply does not desire to meet with the student, say because he does not like the student. This desire conflicts with the student's goal to meet the supervisor.

Resolving this conflict requires the agent whose plan causes conflict to find another alternative plan, or the agent whose desire causes conflict to change his desire (as in desire-desire conflicts).

Resource-Resource Conflict: Two sub-types are distinguished:

- *shared-resource conflict*: This conflict takes place when both agents are competing on the same shared resource.

$$\begin{aligned} \mathit{RRConflict1}(i,j) \quad \equiv \quad & \exists r \in \mathcal{R}^{i,j} \text{ and} \\ & \exists A^i_{p,1} \in \mathcal{I}^i, A^j_{p,1} \in \mathcal{I}^j \text{ such that} \\ & r \in \mathit{Resources}(A^i_{p,1}) \cup \text{ and} \\ & r \in \mathit{Resources}(A^j_{p,1}) \end{aligned} \tag{3.4}$$

- *private-resource conflict*: This takes place when one agent needs a resource that is in another agent's possession.

$$\begin{aligned} \mathit{RRConflict2}(i,j) \quad \equiv \quad & \exists r \in \mathcal{R}^{j} \text{ and} \\ & \exists A^i_{p,1} \in \mathcal{I}^i, \text{ such that} \\ & r \in \mathit{Resources}(A^i_{p,1}) \end{aligned} \tag{3.5}$$

An example of a resource conflict is example 13, in which agent B requires resource "*ford*" from agent S, leading to a private resource conflict.

As mentioned in section 3.3.1, resource conflicts may be resolved by many ways: bargaining over the exchange of private resources (e.g. exchanging "*ford*" for some money), bargaining over the allocation of shared resources, finding alternative plans (e.g. B gets a lift with a friend), or influencing underlying intended goals and desires (e.g. B no longer desires to attend the conference).

Belief-Belief Conflict: This is the conflict addressed in classical argumentation theory, where different agents have conflicting beliefs about the world. They can resolve these conflicts by exchanging arguments over their basic beliefs, through *persuasion* dialogues.

$$\mathit{BBConflict}(i,j) \quad \equiv \quad \exists p \in \mathcal{B}_b^i \text{ and } \neg p \in \mathcal{B}_b^j \tag{3.6}$$

In the context of this thesis, a belief-belief conflict only matters if it leads to other negotiation conflicts. Note that in some definitions above, I included the logical consequences (according to beliefs) of desires and/or plans in the definition of conflicts. This was done by adding $\mathcal{B}_b$ to the condition. Conflicts caused by these logical consequences may be resolved through persuasion dialogues over beliefs. Such persuasion could influence what the agents' believe as being consequences of their desires or plans. For example, suppose one agent desires that the car is empty. Suppose also that another agent has a plan that involves putting a box in the car. If the first agent believes that having the box in the car has the consequence of making the car full, then there will be a conflict between the first agent's desire and the consequence of the second agent's plan. This conflict may be resolved, for example, by persuading the first agent that the car will not be full, say because the box is small.

3.6 A Protocol for IBN

In the previous section, I articulated the types of conflicts among negotiating agents within a precise formal framework. The next step is to provide a communication language and protocol that enable agents to negotiate and exchange the information necessary to resolve their conflicts.

3.6.1 Contracts

An agent negotiates with others in order to access external resources that help it better achieve its desires. Conflict over resources exists because agents are assumed not to have default access to resources other than their own. Through negotiation, agents can attempt to reach agreement on the division of the shared resources or on a mutually useful exchange of their own resources. Such agreements are specified in terms of "contracts." A contract can be seen as a specification of different actions each party must perform for one another.[25] For example, a TV purchasing contract usually involves a buyer making payment in some form to a seller, who in turn provides the TV to the buyer and possibly performs other actions, such as delivering the TV to a designated address, making appropriate installations and so on. A *contract* Ω is an expression of the form:

$$\mathbf{do}(x_1, \alpha_1) \wedge \cdots \wedge \mathbf{do}(x_n, \alpha_n)$$

where x_i denotes the agent that executes the action α_i. Abusing notation, I use $\mathbf{do}(i, \alpha) \in \Omega$ to denote that action $\mathbf{do}(i, \alpha)$ is part of contract Ω. I assume that the cost of shared resources is zero. Executing shared resource r is denoted by $\mathbf{do}(All, r)$ since all agents need to agree on it.

A *proposal* is a contract that has been suggested by some participant(s) but has not yet been accepted by all the parties that are mentioned in the contract. After all relevant parties accept a proposal, it becomes a *deal*. The agent that makes a proposed deal is said to *need* the action(s) he is asking others to perform for him. So for example, if a buyer agent B makes the following proposal to a seller agent S:

$$\mathbf{do}(S, rent) \wedge \mathbf{do}(B, pay\$200)$$

then we say that the buyer agent b has announced that he intends that the other agent perform the action "$rent$" and hence we also say that the buyer *needs* action "$rent$."

---

[25] Elaborate axiomatic treatments of complex interaction towards agreements has been presented through the *joint intention theory* in agent cooperation [Cohen and Levesque, 1991, Grosz and Kraus, 1996, Jennings, 1995]. It would be interesting to explore how joint intention theory can be applied to non-cooperative negotiations of the form explored in this thesis.

3.6.2 Commitments

During an IBN dialogue, an agent may assert information about its private intentions, beliefs and other mental attitudes. Such assertions must, in a sense, influence the way the dialogue unfolds. For example, when an agent Λ_1 asserts a particular intended desire d, its negotiation counterpart Λ_2 should be able to ask for the justification of that desire, or enquire about how Λ_1 intends to achieve that desire and so on. Similarly, if Λ_2 makes an offer Ω to Λ_1, this offer should incur a commitment on Λ_2 to follow through if Λ_1 were to accept Ω. Hence, we need to capture and store the *commitments* agents make during the dialogue and to specify how these commitments influence the subsequent unfolding of the dialogue.

Following the tradition of argumentation theory [Hamblin, 1970], I make use of so-called *commitment stores*. A commitment store $\mathcal{CS}(i)$ is a public-read information store that contains the information that an agent i has publicly committed to. For example, when an agent asserts a proposition p, it is committed to believe that p holds and, additionally, this may imply that it is committed to defend that p (if challenged), not to deny that p, give evidence that p and so on [Walton and Krabbe, 1995]. I do not consider the intricate semantics of commitments themselves, as they pose non-trivial challenges in their own right [Maudet and Chaib-draa, 2003]. I only consider rules of the dialogue that specify when and how commitments are inserted to and removed from commitment stores as a result of making utterances and how this affects agents' subsequent utterances.

Entries in commitment stores are of the form $\mathsf{t}(X)$ where X is the information the agent is committed to and "t" denotes the modality of the information. The modality of a statement indicates which mental attitude applies to it. It can be a belief "bel", a desire "des", or an intention "int." We also have a separate modality "prule" for asserting planning rules agents believe to hold. Commitment stores may also have expressions of the form $\neg\mathsf{t}(X)$; however these have a nonstandard interpretation. A statement of the form $\neg\mathsf{des}(\psi)$ in an agent's commitment store does not mean that the agent has $\neg\mathsf{d}(\psi)$ in its knowledge base (in fact, such expressions are not part of the syntax). Instead, $\neg\mathsf{des}(\psi)$ means that $\mathsf{d}(\psi)$ is *not* in the knowledge base.

Commitments also enable us to specify the consequences of making an offer precisely. Recall the following example offer by agent B:

$$\mathbf{do}(S, rent) \wedge \mathbf{do}(B, pay\$200)$$

This results in the statement $\mathsf{int}(\mathbf{do}(S, rent))$ being added to the commitment store $\mathcal{CS}(b)$. I.e. agent B intends that agent S performs the action "*rent*". The above offer also incurs a (special kind of) *conditional* intention statement on agent b's commitment store: $\mathsf{int}(\mathbf{do}(B, pay\$200) \mid \mathsf{int}_S(\mathbf{do}(S, rent)))$. This means that B is committed to executing "*pay$200*" if S makes a public commitment to execute "*rent*."

Note that an agent is said to need a set of actions in a deal only if the agent is the proposer of that deal. Hence, we cannot say that the seller agent above necessarily needs the buyer to perform action "$pay\$200$." This follows only after the seller has accepted the proposal.

Finally, we have a special predicate "instr" for denoting "instrumentality relations" in plans. By asserting $\mathsf{instr}(X, Y)$, an agent denotes that it adopts a set of goals X because they belong to a plan or set of plans that achieve the higher-level goals or desires Y. The usefulness of this predicate lies in that it allows an agent to reveal information about higher level goals without revealing all intermediate nodes or other planning rules. Consider, for example, a travel agent asking our customer of example 13 why he wants to hire a car. The customer may wish to disclose that this is part of a plan to go to a conference, but may not want to disclose that the conference is in Sydney, or that he also needs to pay registration. Such lack of disclosure may be required for the sake of privacy or be part of the agent's dialogue strategy.

3.6.3 Locutions

I now present the locutions of a formal dialogue game which enable interest-based negotiation. A *dialogue game* is an interaction among two or more participants who make "moves" by uttering locutions, according to certain rules. Dialogue games have been studied since the times of Aristotle [1928] and have been used by philosophers to study fallacious arguments [Walton and Krabbe, 1995]. More recently, they have been used in computer science, for example to build models for human computer interaction [Hulstijn, 2000] and to specify interaction protocols among autonomous software agents [Bench-Capon et al., 1991, Parsons and Jennings, 1996, McBurney and Parsons, 2003].

Dialogue game rules are specified in terms of a *dialogue game protocol*. The protocol is defined in terms of a set of locutions, as well as a number of different rules: *commencement rules*, *combination rules*, *commitment rules* and *termination rules* [McBurney and Parsons, 2003]. Commencement and termination rules specify when a dialogue commences and how it terminates. Commitment rules specify how the contents of commitment stores change as a result of different locutions. Finally, combination rules specify the legal sequences of dialogue moves.

Locutions may be used by agents to construct *dialogue moves* or *utterances*, which are particular instantiations of the locutions within a dialogue. A dialogue move is of the form

$$\mathsf{SPEECH\_ACT}(i, j, \Gamma),$$

where $\mathsf{SPEECH\_ACT}$ denotes the type of message, i is the sender, j is the recipient and Γ the content part of the utterance. For simplicity, we assume that negotiation takes place between two participants, however the protocol may be extended to deal with multi-party negotiations, as long as commitments are managed in the same way.

In the following I describe for each locution the syntax of the speech act implementing it. I give a short intuitive description. Then I list the formal preconditions that should hold before the locution can be used. The postconditions consist of two parts. The first part includes the constraints that are posed on the locutions that can follow (i.e., the legal responses). The second part describes the effects of the locution on the commitment stores.[26].

Table 3.2 shows the main locutions used for bargaining. After agent i makes a proposal using locution **L1**, it is committed to intend that the other agent perform its part in the contract. It also incurs a conditional intentions to execute certain actions in return.

After acceptance using locution **L2**, the agents that proposed and accepted the contract are both conditionally committed to keep their part of the contract. They also intend the other parties to perform their part of the contract. In order to get unconditional commitments we need a confirmation from the proposing party (using the OK locution **L3**). This makes the commitments already present unconditional. Therefore i will intend its

---

[26] This approach to specifying locutions, with semi-formal semantics, shares limitations with similar approaches (e.g. FIPA ACL [FIPA, 2001] based on KQML [Mayfield et al., 1996]). More elaborate semantics can be explored in the future.

L1. $\mathsf{PROPOSE}(i,j,\Omega)$ where Ω is a contract. **Description:** Agent i proposes a contract $\Omega = \mathbf{do}(i,\alpha_1) \wedge \cdots \wedge \mathbf{do}(i,\alpha_n) \wedge \mathbf{do}(j,\beta_1) \wedge \cdots \wedge \mathbf{do}(j,\beta_m)$ to agent j. **Preconditions:** No specific preconditions. **Response:** Agent j can either accept or reject the proposal, make a (counter-)proposal, or initiate an information-seeking or persuasion dialogue. **Commitment Store:** We add to $\mathcal{CS}(i)$ the conditional intention: $\mathsf{int}(\mathbf{do}(i,\alpha_1) \wedge \cdots \wedge \mathbf{do}(i,\alpha_n) \mid \mathsf{int}_j(\mathbf{do}(j,\beta_1) \wedge \cdots \wedge \mathbf{do}(j,\beta_m)))$	**L2.** $\mathsf{ACCEPT}(i,j,\Omega)$ where Ω is a contract. **Description:** Agent i accepts contract $\Omega = \mathbf{do}(j,\alpha_1) \wedge \cdots \wedge \mathbf{do}(j,\alpha_n) \wedge \mathbf{do}(i,\beta_1) \wedge \cdots \wedge \mathbf{do}(i,\beta_m)$ previously proposed by agent j. **Preconditions:** Another agent $j \neq i$ must have previously uttered $\mathsf{PROPOSE}(j,i,\Omega)$ and has not subsequently retracted this proposal. **Response:** An OK by agent j should follow activate the contract (i.e., make it a "deal"). **Commitment Store:** We add to $\mathcal{CS}(i)$ the unconditional intention: $\mathsf{int}(\mathbf{do}(i,\beta_1) \wedge \cdots \wedge \mathbf{do}(i,\beta_m))$
L3. $\mathsf{OK}(i,j,\Omega)$ where Ω is a contract. **Description:** Agent i acknowledges Ω previously proposed and accepted. **Preconditions:** Another agent $j \neq i$ must have previously uttered $\mathsf{ACCEPT}(j,i,\Omega)$. **Response:** No response is needed. **Commitment Store:** Replace the conditional intention in $\mathcal{CS}(i)$ resulting from Ω with an unconditional intention $\mathsf{int}(\mathbf{do}(i,\alpha_1) \wedge \cdots \wedge \mathbf{do}(i,\alpha_n))$	**L4.** $\mathsf{REJECT}(i,j,\Omega)$ where Ω is a contract. **Description:** Agent i states that contract Ω is not acceptable to it. **Preconditions:** Another agent $j \neq i$ must have previously uttered $\mathsf{PROPOSE}(j,i,\Omega)$ and has not subsequently retracted this proposal. **Response:** No response required. **Commitment Store:** No effects.

Table 3.2: Basic Bargaining Locutions

own part of the contract (which now becomes a "deal"). Because this is in its commitment store it is visible for the other agents. They can in principle derive their own belief that i will do its part and thus make their own commitments unconditional as well.

Although there are no immediate effects on the commitment stores after a rejection (locution **L4**), the receiving agent can retract its own (conditional) commitments that were based on the proposed contract, using locution **L8** discussed below.

Table 3.3 shows the locutions used for basic information-seeking and persuasion. Agents make assertions using locution **L5**, and can ask one another questions using locution **L6**. The challenge locution **L7** allows an agent to request justification for beliefs, desires and

L5. $\mathsf{ASSERT}(i, j, X)$ where X is either a belief or explanatory argument of the form $\langle H, h\rangle$ or is an expression of the form $\mathsf{t}(Y)$, where t is either bel, des, int, prule, or instr. **Description:** Agent i states that X is believed to be true. **Preconditions:** No specific preconditions. **Response:** No response required. **Commitment Store:** Statement X is added to $\mathcal{CS}(i)$. If the statement is of the form $\mathsf{instr}(S, S')$ then $\mathsf{int}(p)$ is added for each $p \in S'$.	**L6.** $\mathsf{QUESTION}(i, j, X)$ where X is of the form $\mathsf{t}(Y)$. **Description:** Agent i asks agent j whether it believes X. **Preconditions:** No specific preconditions. **Response:** Agent j must assert either X, $\neg X$, or declare undecidedness, in which case it asserts $\mathcal{U}$. **Commitment Store:** No effects.
L7. $\mathsf{CHALLENGE}(i, j, X)$ where X is of the form: $\mathsf{bel}(Y)$, $\mathsf{des}(Y)$. **Description:** Agent i asks agent j to provide a justification for formula X. **Preconditions:** $X \in \mathcal{CS}(j)$. **Response:** If X is of the form $\mathsf{bel}(p)$, then agent j must assert a belief argument $\langle H, p\rangle$; whereas if X is of the form $\mathsf{des}(p)$, then agent j must assert an explanatory argument $\langle H, \mathsf{d}(p)\rangle$. Otherwise, j must retract formula X. **Commitment Store:** No effects.	**L8.** $\mathsf{RETRACT}(i, j, X)$. **Description:** Agent i retracts formula X that it has previously asserted. **Preconditions:** $X \in \mathcal{CS}(i)$. **Response:** No response required. **Commitment Store:** X is removed from $\mathcal{CS}(i)$.

Table 3.3: Basic information-seeking and persuasion locutions

planning rules. However, the type of justification given (using locution **L5**) is different for each modality. A justification for a belief will be a set of other beliefs that imply the asserted belief. The justification for a desire will be an instrumental argument (or part thereof) that specifies the context in which this desire holds. Agents can then argue, in accordance with the argumentation framework I presented earlier, about these desires and beliefs.

Note that the challenge locution cannot be used to request for justifications of intentions $\mathsf{int}(Y)$, or instrumentality statements $\mathsf{instr}(Y, Z)$. For the case of intentions, an agent adopts an intention because this intention is *instrumental* towards achieving a higher-level intention and because it is *achievable* by achieving some other set of (sub-

)intentions. To allow for exploration of both directions, I provide two separate locutions **L9** and **L10** defined in table 3.4 below. For statements of the form $\mathsf{instr}(Y, Z)$, no challenges are allowed either. This is because such statements are made because the utterer wishes to withhold some information about its plans. To challenge this statement is to ask for the full plans, which the utterer, obviously, is not willing to provide at this stage.

Locution **L8** allows agents to retract their commitments, including those incurred by making proposals. Note that in addition to retracting X, i may also be required to retract all things that depend in some way on X. Thus if X is a belief that is necessary to derive a previously asserted belief Y, then Y has also to be retracted. An extensive relevant study on managing commitment in dialogue can be found in the seminal book by Walton and Krabbe [1995].

L9. REQ-PURPOSE(i, j, x) where $x \in \mathit{ACTS} \cup \mathit{PROPS}$.	**L10.** REQ-ACHIEVE(i, j, x) where $x \in \mathit{PROPS}$.
Description: Agent i asks agent j to assert one of the super-goals of the action or goal denoted by x. **Preconditions:** $\mathsf{int}(x) \in \mathcal{CS}(i)$. **Response:** Agent j must either assert a statement of the form $\mathtt{instr}(\{x\}, \mathit{Sup})$, or assert that x is a desire. **Commitment Store:** No effects.	**Description:** Agent i asks agent j to explain how it intends to achieve the goal or desire denoted by x. **Preconditions:** Either $\mathsf{int}(x) \in \mathcal{CS}(i)$ or $\mathsf{des}(x) \in \mathcal{CS}(i)$. **Response:** Agent j must either assert a planning rule of the form $(\varphi_1 \wedge \cdots \wedge \varphi_n \rightarrowtail x)$, or assert a statement $\mathtt{instr}(\{\psi_1, \ldots, \psi_m\}, \{x\})$. **Commitment Store:** No effects.

Table 3.4: Locutions for seeking information about plan structure

Note that it does not make sense for an agent to ask another agent how it would achieve a primitive action (i.e. a resource), since actions can only be executed directly. Therefore, the REQ-ACHIEVE(i, j, x) locution is only allowed if x is a goal or a desire of agent i, not an intended action. This condition is imposed by the locution syntax.

Finally, table 3.5 defines a locution for leaving the dialogue and a locution PASS(i) which allows an agent to stay silent and see what else the other has to say, hence avoiding strict turn taking in the dialogue.

L11. QUIT(i). **Description:** Agent i announces its withdrawal from the negotiation dialogue. **Preconditions:** No specific preconditions. **Response:** None required. **Commitment Store:** No effects.	**L12.** PASS(i) allows agent i to pass a turn by saying nothing.

Table 3.5: Additional locutions

3.6.4 Protocol

In the previous section, I listed locutions that can be used to conduct interest-based negotiation dialogues. In this section, I present a set of structured protocols which demonstrate how the locutions can be used for interest-based negotiation. Protocols are specified in a similar fashion to the way information-seeking, inquiry and persuasion protocols are presented by Parsons et al. [2002].

Information Seeking

Parsons et al. [2002] define a basic *information-seeking* dialogue in which an agent seeks an answer to some question from another and may challenge the answer given until it is satisfied with the reasons supporting that answer. In this section, I define a number of protocols for information-seeking to cater for IBN dialogues. The most basic form of information-seeking is a yes/no question.

InfoSeek1[A,B]:

1. A questions statement X.
2. B asserts X, $\neg X$ or $\mathcal{U}$.

Information-seeking can also be done by “challenging” statements. When an agent challenges a belief, the challenged agent must provide a belief argument to support that belief. When an agent challenges a desire, the challenged agent must provide an explanatory argument to support that desire. These two schemas are captured in the following protocol.

InfoSeek2[A,B]:
Precondition: B has a statement $\mathsf{bel}(p)$ or $\mathsf{des}(p)$ –call it X– in its commitment store.

1. A challenges a belief or desire of B.
2. B asserts $\left\{ \begin{array}{ll} S \text{ such that } S \vdash p & \text{if } X = \mathsf{bel}(p) \\ S \text{ such that } S \vdash_d \mathsf{d}(p) & \text{if } X = \mathsf{des}(p) \end{array} \right.$
3. A either $\left\{ \begin{array}{ll} \text{passes its turn} & \text{if it has received enough information} \\ \text{For each } s \in S\text{, go to 1} & \text{otherwise} \end{array} \right.$

Having defined protocols for seeking information about beliefs and desires and their justifications, what remains are protocols for seeking information about intentions. Given an intention, we require two protocols: one for seeking information (upwards) about the purpose of that intention, and another for seeking information (downwards) about how this intention is to be achieved. The following two protocols enable this.

InfoSeek3[A,B]:
Precondition: B has a statement $\mathsf{int}(p)$ in its commitment store.

1. A asks B for the purpose of p –using locution **L9**.
2. B asserts $\left\{ \begin{array}{ll} \mathsf{des}(p)\text{, and the protocol terminates} & \text{if } p \text{ is a desire of } B \\ \mathsf{instr}(\{p\}, \mathit{Sup}) \text{ for some } \mathit{Sup} & \text{otherwise} \end{array} \right.$
3. A either $\left\{ \begin{array}{ll} \text{passes its turn} & \text{if it has received enough information} \\ \text{For each } s \in \mathit{Sup}\text{, go to 1} & \text{otherwise} \end{array} \right.$

InfoSeek4[A,B]:
Precondition: B has a statement $\mathsf{int}(p)$ in its commitment store, such that $p \notin \mathit{ACTS}$ (i.e. p is not a resource/primitive action).

1. A asks B about how it intends to achieve p –using locution **L10**.
2. B asserts a planning rule $\texttt{prule}(\varphi_1 \wedge \cdots \wedge \varphi_n \rightarrowtail p)$ or an instrumental statement $\texttt{instr}(\{\varphi_1, \ldots, \varphi_m\}, \{p\})$
3. A either $\left\{ \begin{array}{ll} \text{passes its turn} & \text{if it has received enough information} \\ \text{For each } \varphi_i \notin \mathit{ACTS}\text{, go to 1} & \text{otherwise} \end{array} \right.$

Persuasion

Persuasion dialogues start with one agent believing p or $\neg p$ and another agent intending to change that belief. I list a slightly modified version of the protocol presented by Parsons et al. [2003].

Persuade1[A,B]:

1. A asserts p.
2. B either $\begin{cases} \text{accepts by reasserting } p\text{, the protocol terminates} & \text{if it agrees,} \\ \text{asserts } \neg p & \text{if it disagrees,} \\ \text{challenges the assertion} & \text{otherwise} \end{cases}$
3. If B has just asserted $\neg p$, then go to 2 with the roles reversed and $\neg p$ in place of p.
4. If B has challenged p, then:
   (a) A asserts S, the support of p;
   (b) Go to 2 for each $s \in S$ in turn.
5. B accepts p or the protocol terminates

It would be expected that similar protocols should be presented to enable an agent to persuade another agent to adopt a desire or an intention. However, unlike beliefs, which are based on objective facts that both agents can jointly assess, desires and intentions are more subjective and agent-dependent. An agent A may not desire ψ but believe that there are sufficient conditions for another agent B to desire ψ. This is because A appeals to desire-generation rules that apply to B but not A.

Suppose the desire-generation rule $\varphi_1 \wedge \cdots \wedge \varphi_n \wedge \mathsf{d}(\psi_1) \wedge \cdots \wedge \mathsf{d}(\psi_m) \Rightarrow \mathsf{d}(\varphi)$ is in agent B's knowledge base. Agent A may try to persuade B to adopt desire $\mathsf{d}(\psi)$ even though A itself does not have such desire. This may be because (i) A does not have such rule in its own knowledge base; or (ii) although A has such a rule, A does not desire $\mathsf{d}(\psi_1) \wedge \cdots \wedge \mathsf{d}(\psi_m)$ and so the rule would not apply to A, but may apply to B if it has such desires.

Persuade2[A,B]:
Precondition: $[\varphi_1 \wedge \cdots \wedge \varphi_n \wedge \mathsf{d}(\psi_1) \wedge \cdots \wedge \mathsf{d}(\psi_m) \Rightarrow \mathsf{d}(\varphi)] \in \mathcal{CS}(B)$, and $\neg\mathsf{des}(\psi) \in \mathcal{CS}(B)$.[27]

1. A attempts **Persuade1[A,B]** for each $\varphi_1 \wedge \cdots \wedge \varphi_n$. If any fails, the protocol terminates.
2. A attempts **Persuade2[A,B]** for each $\mathsf{d}(\psi_1) \wedge \cdots \wedge \mathsf{d}(\psi_m)$. If any fails, the protocol terminates.
3. If protocol has not terminated earlier, B asserts $\mathsf{des}(\psi)$.

---

[27] The first condition means that B agrees on the plan generation rule, either because it already has it before the dialogue commenced, or because it has agreed on it upon receiving it during the dialogue. The

A slight variant of the above protocol can be used to allow agent A to attempt to persuade agent B to *drop* an existing desire ψ. The only difference is that the commencement condition would be that $\mathsf{des}(\psi) \in \mathcal{CS}(B)$ and that A would instead attempt to disqualify the condition of the rule rather than establish it.

Finally, we require a means to enable an agent to persuade another agent to adopt an intention. This is basically done by presenting new planning rules to enable the persuadee to construct a new plan for a given desire.

Negotiation

The most basic form of negotiation is based on a simple *request* protocol.

RequestDialogue[A,B]:

1. A proposes deal Ω.
2. B either accepts of rejects.

A *bargaining* dialogue is a series of request dialogues that terminate once an agent accepts a proposal. An *interest-based negotiation* dialogue is a mix of bargaining, information-seeking and persuasion dialogues.

Additional Rules

I now present some additional rules to control the dialogue. One of the most commonly used rules is that arguments cannot be repeated. This rule is not suitable for IBN. Suppose an agent proposes a contract Ω, then retracts Ω after being persuaded that it is not useful. If the agent is somehow persuaded again of the usefulness of Ω, then it should be allowed to propose it again. Therefore, I require instead that agents do not make utterances that have effects already present in the commitment stores.[28]

As a second rule I require that an agent can never challenge an assertion of another agent if that assertion is already part of its own commitment store. Note that this does not mean that an agent cannot, for example, challenge a belief of a counterpart while believing

---

second condition means that B was asked whether it desires ψ and replied negatively.

[28] This rule does not apply to utterances that have no effect on the commitment store, such as rejections.

it himself (privately). An agent man believe something but doubt whether another agent has enough information to also believe it.

I use the following rules for the commencement and termination of the dialogue:

Definition 22. ***(Commencement Condition)*** *A negotiation dialogue commences when an agent utters the* PROPOSE(.) *locution.*

Definition 23. ***(Termination Conditions)*** *A negotiation dialogue may terminate either successfully or unsuccessfully, as follows:*

- *A dialogue* terminates unsuccessfully *when some participant i utters the locution* QUIT(i).
- *A dialogue* terminates successfully *with deal Ω if and only if each agent announces an unconditional intention to perform its part of the deal. Note that this takes place after some agent utters the final* OK(.) *locution.*

A dialogue can now be defined as follows:

Definition 24. ***(Dialogue)*** *A* dialogue *is a finite sequence of dialogue moves, constructed using the locutions* ***L1*** *to* ***L12****, made by participants according to the rules specified in the locution specifications, protocols and dialogue rules above and conforming to the commencement and termination conditions.*

3.7 Putting the Pieces Together

Up to this point, the discussion of different components has been rather scattered. In this section, I put all the pieces of the framework together, making clear what modules have and have not been formalised.

Let Λ be the set of agents participating in the dialogue. Let *UTTR* be the set of all possible utterances that can be generated according to the protocol. And let a *negotiation thread TH* be the set of all possible sequences of utterances that can be generated according to the IBN protocol. An agent can then be defined as follows:

Definition 25. **(Agent)** *An* agent *is a tuple $\langle \mathcal{B}_b, \mathcal{B}_d, \mathcal{B}_p, \mathcal{D}, \mathcal{ID}, \mathcal{I}, \mathcal{R}, \Theta \rangle$ such that:*

1. *$\mathcal{B}_b$ is a set of basic beliefs;*

2. *$\mathcal{B}_d$ is a set of desire generation rules;*

3. *$\mathcal{B}_p$ is a set of planning rules;*

4. *$\mathcal{D}$ is a set of desires initialised to $\emptyset$;*

5. *$\mathcal{ID}$ is a set of intended desires initialised to $\emptyset$;*

6. *$\mathcal{I}$ is a set of intentions initialised to $\emptyset$;*

7. *$\mathcal{R} \subseteq$ RES is a set of resources the agent owns;*

8. *$\Theta : TH \times \bigcup_{i \in \Lambda} \mathcal{CS}(i) \rightarrow UTTR$ is a dialogue strategy;*

It is worth redescribing the abstract reasoning cycle presented in figure 3.6 in terms of the formal framework. The reasoning cycle for agent i goes as follows. Using its basic beliefs $\mathcal{B}_b^i$ and desire generation knowledge-base $\mathcal{B}_d^i$, the agent generates its desires $\mathcal{D}^i$ using the formal framework presented in section 3.4. Then using planning rules in $\mathcal{B}_p^i$, the agent produces an intention $\mathcal{I}^i$ (i.e. plan) for achieving a subset of its desires $\mathcal{ID}^i$ called its intended desires. If the agent can achieve its intention on its own (because it possesses all the resource required), then it does so. Otherwise, if the agent requires resources that other agents possess, or resources that are shared with other agents, then a negotiation dialogue is initiated. The function Θ^i specifies the negotiation strategy of agent i. Based on the history of utterances exchanged so far and the contents of participants' commitment stores, the strategy function generates an utterance to make according to the rules of the protocol. Of course, Θ^i also implicitly takes into account the state of the agent itself.

The main contribution of this chapter is the specification of an argumentation-theoretic account of the *argument interpretation* component of the agent architecture presented in figure 3.5. In other words, my focus thus far was on providing a formal framework that specifies the *effect* of arguments on agents' preferences, rather than on the *strategies* used to generate these arguments (i.e. the details of Θ^i). I therefore did not specify how *argument interpretation* and *argument selection* modules may be implemented. I explore strategic issues using a less formal perspective in chapter 5. Nevertheless, I present an example that uses simple ad hoc strategies in the following subsection.

A Complete Example

I now present an illustrative example, which builds on examples 13 and 14 discussed earlier, to show how the argumentation frameworks and protocol can be used to specify agents capable of conducting interest-based negotiation dialogues. I shall present the various stages of the scenario progressively, beginning with an agent inferring a desire.

Recall that:

aic = "attend the Sydney AI conference";

syd = "go to Sydney";

reg = "pay conference registration";

rent = "rent a car";

ford = "get a particular car of make Ford";

call = "call a friend";

friend = "stay at a friend's place";

Inferring Desires: Recall from earlier examples above that buyer agent B generated two intended desires $\mathcal{ID}^B = \{aic, nz\}$. Now, suppose a seller S has an unconditional desire to conduct business, denoted *biz*. Hence $\mathcal{ID}^S = \{biz\}$.

Intention Formation: After inferring a set of desires from the contextual beliefs, an agent uses planning rules to construct a consistent set of plans that achieve (a sub-set of) his desires. The agent, therefore, intends each of these desires, along with all the goals and actions that constitute the selected plans. Recall that agent B intended instrumental argument (or plan) A_1 to achieve *aic* and plan A_3 to achieve *nz*. Suppose S has the following knowledge base:

1. $\mathcal{B}_b^S = \emptyset$

2. $\mathcal{B}_d^S = \{\Rightarrow biz, 1)\}$

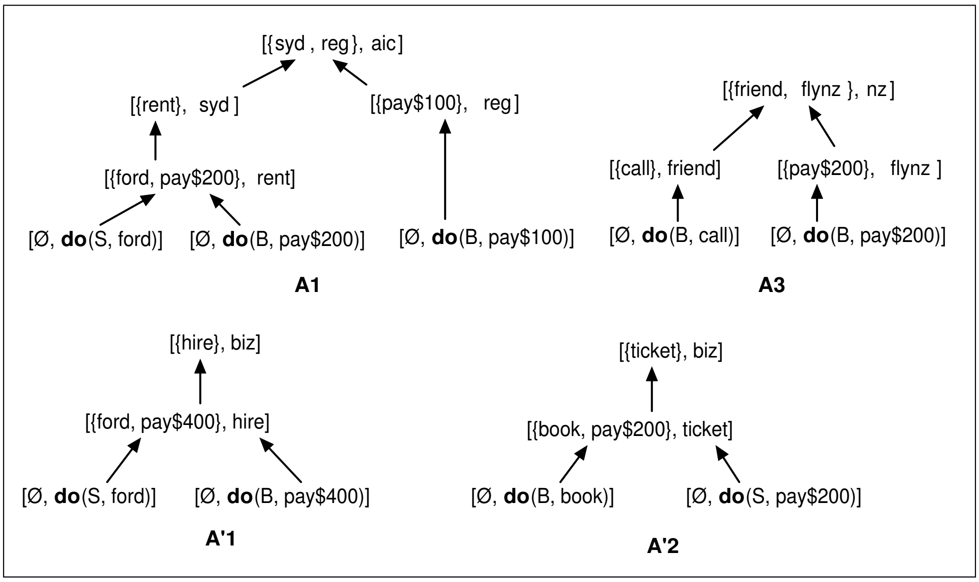


Figure 3.10: Intentions of two agents

3. $$\mathcal{B}_p^S = \begin{cases} hire & \rightarrowtail biz \\ ticket & \rightarrowtail biz \\ ford \wedge pay\$400 & \rightarrowtail hire \\ book \wedge pay\$200 & \rightarrowtail ticket \\ flysyd & \rightarrowtail syd \\ book \wedge pay\$200 \rightarrowtail flysyd & \end{cases}$$

where:

biz = “conduct a profitable business transaction”;

hire = “hire a car”;

ticket = “provide a ticket to Sydney”;

book = “book a ticket”;

Figure 3.10 depicts the resulting intentions of both agents.

Need for Negotiation: Suppose agent S has resource set $\mathcal{R}^S = \{ford, book\}$ and suppose that agent B’s resources are $\mathcal{R}^B = \{pay\$200, pay\$100, pay\$400\}$. In this case, we have a private resource conflict. Agent B needs S to execute action *ford*. On the other hand, agent S needs B to execute action *pay$400*.

Proposal Exchange: After one or more agents indicate certain *needs* (i.e., intentions to receive resources or services from each other), they may start a negotiation dialogue by exchanging proposed contracts that (more or less) fulfil these needs. Proposals may be accepted, rejected, or simply ignored. The following dialogue fragment shows two proposals exchanged between S and B, both rejected.

Suppose the buyer starts the negotiation by making the following proposal:

1. Buyer: $\mathsf{PROPOSE}(B, S, \mathbf{do}(S, \mathit{ford}))$

This adds $\mathtt{int}(\mathbf{do}(S, \mathit{rentCar})$ to $\mathcal{CS}(B)$. Of course, S would not accept this proposal because it would only incur a cost without any benefit in return. Suppose agents make the following two counterproposals, both rejected.

2. Seller: $\mathsf{PROPOSE}(S, B, \mathbf{do}(S, \mathit{ford}) \wedge \mathbf{do}(B, \mathit{pay\$400}))$
3. Buyer: $\mathsf{REJECT}(B, S, \mathbf{do}(S, \mathit{ford}) \wedge \mathbf{do}(B, \mathit{pay\$400}))$
4. Seller: $\mathsf{PASS}(S)$
5. Buyer: $\mathsf{PROPOSE}(B, S, \mathbf{do}(S, \mathit{ford}) \wedge \mathbf{do}(B, \mathit{pay\$200}))$
6. Seller: $\mathsf{REJECT}(S, B, \mathbf{do}(S, \mathit{ford}) \wedge \mathbf{do}(B, \mathit{pay\$200}))$
7. Buyer: $\mathsf{PASS}(b)$

Each of the above proposals creates a conditional commitment in the proposer's commitment store. It might be that both buyer (academic) and seller (travel agent) have alternative plans to satisfy their needs. For example, the academic started with his most preferred plan, but may later change to a less preferred plan where he pays \$250. The same holds for the travel agent. This will lead to the traditional bargaining game which results in a deal if the parties reach a price that is acceptable for both of them.

Information and Argument Exchange: I now illustrate how the IBN protocol allows agents to influence the negotiation outcome by influencing each other's plans. The seller could ask the buyer for the goal that motivates the need for hiring a car. The buyer asserts that he wants the car in order to travel to Sydney.

8. Seller: $\mathsf{REQ\text{-}PURPOSE}(S, B, \mathit{ford})$

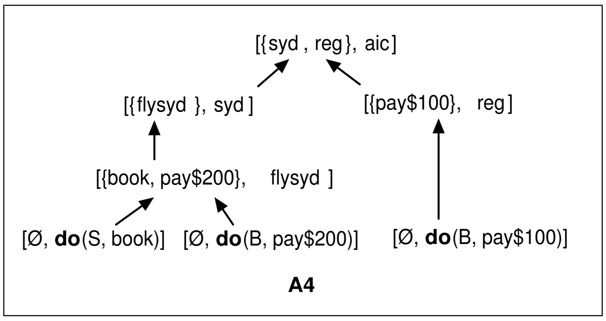


Figure 3.11: Agent B's new plan for attending the Sydney AI Conference

9. Buyer: ASSERT$(B, S, \texttt{instr}(\{\mathit{ford}\}, \mathit{syd}))$

By revealing this information, the buyer gives the opportunity to the seller to present, through a set of planning rules, an alternative means for travelling to Sydney via flight.

10. Seller: ASSERT$(S, B, \mathsf{prule}(\mathit{flysyd} \rightarrowtail \mathit{syd}))$

11. Buyer: PASS(B)

12. Seller: ASSERT$(S, B, \mathsf{prule}(\mathit{book} \wedge \mathit{pay\$200} \rightarrowtail \mathit{flysyd}))$

13. Buyer: PASS(B)

The buyer can now construct a new plan for getting to Sydney and ultimately for achieving the desire to attend a conference. The plan involves two actions: booking a ticket and going to the airport. This plan, call it A_4, is depicted in Figure 3.11. Because this plan also involves only one action pay\$200 on behalf of B, it has the same cost and hence utility, as the earlier plan A_1. Because A_1 could not be achieved because S rejected the associated plan, B will intend A_4.

Now the seller offers to book the ticket for \$200, cheaper than the car-renting contract she was willing to accept.

14. Seller: PROPOSE$(S, B, \mathbf{do}(S, \mathit{book}) \wedge \mathbf{do}(B, \mathit{pay\$200}))$

15. Buyer: ACCEPT$(B, S, \mathbf{do}(S, \mathit{book}) \wedge \mathbf{do}(B, \mathit{pay\$200}))$

It is clear how the protocol allows agents to request information about each other's plans, then exploit the public information available in the commitment stores in order to present persuasive arguments that may influence each other's needs. In the example above, the seller succeeded in making the buyer adopt a new intention (to get a ticket booked), which enabled an alternative deal.

3.8 Discussion and Conclusion

The main contribution of this chapter is the definition of a comprehensive framework for interest-based negotiation dialogues among autonomous computational agents. A conceptual framework is first presented, on which a particular formal framework was then based. The formal framework includes (i) a set of primitives and argumentation-theoretic mechanisms for allowing agents to generate desires and plans, and to modify these as a result of receiving new information; and (ii) a communication protocol that enables negotiation dialogues in which agents exploit the underlying argumentation theory in order to reach better agreements. The protocol allows agents to make proposals and requests, accept or reject proposals, challenge assertions and seek information about each others' plans during negotiation in a structured way. The argumentation theory describes how agents' desires, plans, and (consequently) preferences can change as a result of receiving new information from their negotiation counterparts. Therefore, the argumentation frameworks provide a kind of semantics of the dialogue protocol. By giving such semantics for IBN, the formalisation presented in this chapter lays the ground for further formal analysis of IBN dialogues and the properties of different strategies. This analysis provides fertile ground for further work beyond this thesis.

The notion of conflict I discussed is not exhaustive. Other types of conflicts may arise in multi-agent systems, such as those relating to temporal aspects of planning. To identify such conflicts, the planning language must capture temporal notions explicitly. Resolving these conflicts could then be achieved, for example, by reordering actions in different plan. The framework presented here provides building blocks and concepts that may be extended in that direction.

The IBN framework presented here also has limitations in relation to what agents can argue about. For example, while an agent can assert new planning rules to another agent, it cannot ask the other agent for the reason behind a planning rule. This means that if an agent believes he can go to Sydney by driving, there is no way for its counterpart to ask "why do you believe driving takes you to Sydney?" and possibly persuade him otherwise. This limitation is a consequence of the argumentation theory used. To enable such dialogue, we require a theory that enables agents to construct proofs for which the conclusions are planning rules. In this vein,

The PERSUADER [Sycara, 1992] is a negotiation system that relies on exploring goal structures. It relies on a trusted third-party –a *mediator*– which attempts to find compromise between two disputants. My framework differs from the PERSUADER since it enables agents to resolve conflicts in a distributed fashion through direct interaction. Other centralised approaches for merging multiple plans through a trusted mediator have been proposed by Barber et al. [1999] and by Clement and Durfee [1999].

IBN is not an isolated effort towards flexible negotiation dialogues. Various frameworks for argumentation-based negotiation have been explored in the literature [Rahwan et al., 2003b]. The framework presented in this chapter differs from those presented by Parsons et al. [1998] and Amgoud et al. [2000] in that it makes a clear distinction between preferences used in evaluating arguments over beliefs (reflecting objective truth) and preferences over negotiation outcomes (reflecting subjective gain). Both Parsons et al. and Amgoud et al. incorporate the latter implicitly within the former, since such preferences are required in order for the underlying argumentation logic to work. Since argumentation over beliefs is distinct from internal reasoning about plans and utilities, it becomes possible for agents to discuss objective truth (e.g. whether a conference has been cancelled, or whether there are flights to Sydney) separately from their subjective evaluation of outcomes (e.g. whether an agent prefers to fly or drive).

The framework presented by Sadri et al. [2002] bears some similarity to IBN. However, in Sadri's framework, the protocol is defined implicitly as part of the agents programs, whereas our protocol is externally defined as a dialogue game.[29] More importantly, Sadri's framework only allows agents to exchange a limited number of argument types. When a request is challenged, the challenged agent provides all relevant informa-

---

[29] Of course, in order to program agent strategies, one may *also* encode the protocol rules inside the individual agents. However, having an external representation of the protocol makes it easier to verify, modify and analyse.

tion: the complete plan for which the requested resource is needed and the goal which this plan is intended to achieve. IBN is more flexible since it allows agents to progressively reveal such information –or even withhold it– as part of their dialogue strategies.

Finally, a notable related work, developed more-or-less in parallel with this thesis, is that of Atkinson et al. [2004, 2005] on formalising dialogues for practical reasoning. One difference between my framework and theirs is that they aim at supporting persuasion over action, where one agent supports one action and the other may argue about the former's position. They also construct their framework around a different reasoning schema, founded on a theory of argumentation over action, that makes use of beliefs, actions, goals and values promoted by these goals. Their framework treats the relationship between actions, goals and values more abstractly than I do, which is sufficient for their purpose of identifying a comprehensive set of attacks that can be made against a position. In my framework, however, I encode justification and causal relationships using desire generation rules and planning rules and allow agents to argue about these on a more "micro-level." It would be interesting to further investigate how the two frameworks relate, and investigate how Atkinson et al's framework can be extended for negotiation in the sense described in this thesis.

Having specified an IBN framework, the rest of the thesis contains various elaborations on IBN. In the following chapter, I provide a simplified formalisation of IBN which enables me to draw some comparisons between IBN and bargaining. In Chapter 5 I present a generic methodology for designing negotiation strategies and show how it can be used to build strategies for agents participating in an IBN dialogue according to the protocol presented here. Finally, in Chapter 6 I discuss an implementation of a pilot application that makes use of IBN.

Chapter 4

On Interest-based Negotiation and Bargaining

> *"With, without;*
> *and who'll deny,*
> *it's what the fighting is all about."*
>
> Pink Floyd, 'Dark Side of the Moon'

In this chapter, I draw some comparisons between argumentation-based negotiation and bargaining.

4.1 Introduction

The interest-based negotiation framework I presented in Chapter 3 was motivated by an informal demonstration of the potential benefit of "argumentation" in improving negotiation when agents have incomplete and/or improper preferences. Such informal demonstration, which appeals more or less to intuition, has also been the driving force behind other framework of argumentation-based negotiation (ABN) surveyed in Chapter 2. A reasonable question to ask is:

> *Is argument-based negotiation really better than bargaining-based negotiation mechanisms?*

This chapter begins an investigation towards answering this question.

Recall that the majority of automated negotiation frameworks in the literature are focused on *bargaining* and *auctions*. I shall concentrate my analysis on bargaining, by which I refer to the family of protocols where the main form of interaction is the exchange of potential deals, i.e. potential allocations of the resources in question. Despite the advances made to date, the relationship between argument-based negotiation and bargaining frameworks has been rather informal [Jennings et al., 1998a]. This chapter presents a preliminary investigation into understanding this relationship.

A comparison between bargaining and argument-based negotiation framework can take a variety of forms. For example, one may analytically compare the complexities of dialogues generated using specific bargaining and argument-based negotiation protocols. Very little research has been done in this area, though Wooldridge and Parsons [2000] provide a starting point by studying the computational complexity of simple bargaining protocols, and (separately) present some complexity results for specific information-seeking, inquiry and persuasion dialogues [Parsons et al., 2003]. Another way to compare bargaining and ABN is through empirical investigation into performance issues (e.g. in terms of average outcome quality, dialogue length etc.) in different dialogues. Preliminary results of this kind have been presented by Karunatillake and Jennings [2004]. While these types of analyses are essential to enable us to understand specific argumentation frameworks, my aim in this chapter is more general; it is to understand when and whether argumentation can lead to better outcomes. In particular, I present a set of negotiation concepts through which I analytically study both bargaining and argument-based methods. I demonstrate that if agents have false beliefs, then they may make decisions during negotiation that lead them to suboptimal deals. I then describe a number of ways in which argument-based communication can cause changes in an agent's beliefs and, consequently, its preferences over contracts. This enables me to demonstrate how the argument-based approach can improve the likelihood and quality of deals, but might also lead to worse deals.

The chapter advances the state of the art in argumentation-based negotiation in two ways. First, it provides a step towards a more systematic comparison of argument-based and bargaining-based negotiation frameworks. Second, by making the link between belief change and preference change more explicit, the chapter paves the way for the study of negotiation strategies within argument-based frameworks.

This chapter is organised as follows. In the next section, I provide a conceptual framework which enables us to capture key negotiation concepts. I use these concepts in Section 4.3 to show how bargaining works and demonstrate how it can lead to suboptimal outcomes. In Section 4.4, I present an abstraction of a class of argument-based negotiation frameworks. Also, I show different ways in which preferences can change due to changes in beliefs, and draw some comparisons with bargaining. The chapter concludes in Section 4.5.

4.2 A Conceptual Framework for Negotiation

In this section, I set up the scene for the rest of the chapter by providing a particular formalisation of concepts involved in negotiation.

4.2.1 Agents and Plans

I will confine my discussion to two autonomous agents A and B sharing the same world, which is in some initial state $s \in \mathcal{S}$, where $\mathcal{S}$ is the set of all possible world states. Each agent might, however, believe it is in a different state from s, which can influence its decisions.

To get from one state, s_1 to another, s_2, agents execute actions. An action $\alpha \in \mathcal{A}$, where $\mathcal{A}$ is the set of all possible actions, moves the world from one state to another; hence it is a function $\alpha : \mathcal{S} \to \mathcal{S}$. I assume that actions are deterministic, and that the world changes only as a result of agents executing actions.[1]

Definition 26. (Plan) *A* one-agent plan *or simply* plan P *to move the world from state* s_1 *to* s_2 *is a finite list* $[\alpha_1, \dots, \alpha_n]$ *of actions such that* $s_2 = \alpha_n(\alpha_{n-1}(\dots \alpha_1(s_1) \dots))$

I denote by $\mathcal{P}$ the set of all possible plans. And I denote by $s_1 \models [P]s_2$ that if the world is in state s_1, then executing plan P moves the world to state s_2.

---

[1] Though this treatment of actions is rather naive, I made this choice deliberately in order to simplify the analysis.

What I have just defined is the *objective* action operators specification, i.e., how the world actually changes as a result of executing actions. Agents, however, might have possibly incomplete or incorrect beliefs about how the world changes as a result of executing actions. I therefore assume each agent i has its own mapping $\alpha^i : \mathcal{S} \cup \{?\} \rightarrow \mathcal{S} \cup \{?\}$ for each action, such that always $\alpha^i(?) = ?$. If $\alpha^i_x(s_1) = ?$, then we say that agent i does not know what state action α_x results in if executed in state s_1. The expression $s_1 \models^i [P]s_2$ means that agent i believes executing plan P in state s_1 results in state s_2.[2] Moreover, the expression $s_1 \models^i [P]?$ means that agent i does not know what state results from executing plan P in state s_1.

Agents can evaluate actions and plans based on their costs.

Definition 27. (Cost of Action) *The* cost of action α *for agent* $i \in \{A, B\}$ *is defined using an action cost function* $Cost : \{A, B\} \times \mathcal{A} \rightarrow \mathbb{R}^+$*, which assigns a number to each action.*

Definition 28. (Cost of Plan) *The* cost of plan $P \in \mathcal{P}$ *to agent* i *is defined using a plan cost function*

$$Cost : \{A, B\} \times \mathcal{P} \rightarrow \mathbb{R}^+ \text{ such that } Cost(i, P) = \sum_{\alpha \in P} Cost(i, \alpha)$$

Unlike the case with action operators, where agents can have incorrect beliefs about the results of actions, I assume each agent has accurate knowledge about how much each action costs him/her. However, an agent may not know how much an action would cost another agent (i.e., I only assume each agent i knows accurately what *Cost*(i, α) is for each α).

Each agent $i \in \{A, B\}$ has a set of desires $\mathcal{D}^i \subseteq \mathcal{D}$, where $\mathcal{D}$ is the set of all possible desires. These desires are formulae in propositional logic or closed formulae in first-order logic (i.e., with no free variables). We say that a world state s satisfies a desire d if $s \models d$, where $\models$ is an appropriate semantic entailment relation.

Definition 29. (Worth of Desire) *The* worth of desire d *for agent* i *is defined using a desire worth function* $Worth : \{A, B\} \times \mathcal{D} \rightarrow \mathbb{R}^+$*, which assigns a number to each desire.*

---

[2] Note that $\models^i$ is not a new "semantic operator." It is merely a syntactic abbreviation for distinguishing different agents' beliefs.

Definition 30. (Worth of State) *The* worth of state *$s \in \mathcal{S}$ to agent i is defined using a state worth function*

$$Worth : \{A, B\} \times \mathcal{S} \rightarrow \mathbb{R}^+ \text{ such that } Worth(i, s) = \sum_{s \models d} Worth(i, d)$$

As with costs, each agent knows precisely what each desire is worth to him/her. Also, an agent may not know how much a desire is worth to another agent (i.e., I only assume each agent i knows accurately what $Worth(i, s)$ is).

I can now define the *utility* of a plan for an agent given it is in a particular state. I distinguish between the *objective* and *perceived* utility. The objective utility denotes the *actual* gain achieved by the agent based on the actual resulting state (i.e., according to the objective action operators definition). The perceived utility, on the other hand, is the utility the agent *thinks* it would achieve from that plan, based on what it believes the resulting state is.[3]

Definition 31. (Utility of Plan) *The* utility of plan *P for agent i from state s_1 is defined as:*

$$Utility(i, P, s_1) = Worth(i, s_2) - Cost(i, P) \text{ where } s_1 \models [P]s_2$$

Definition 32. (Perceived Utility of Plan) *The* perceived utility of plan *P for agent i from state s_1 is defined as:*

$$Utility^i(i, P, s_1) = Worth(i, s_2) - Cost(i, P) \text{ where } s_1 \models^i [P]s_2$$

Definition 33. (Best Plan) *The* best plan *for agent i from state s_1 is a plan $P = BestP(i, s_1)$ such that $Utility(i, P, s_1) \geq Utility(i, P', s_1)$ for all $P' \neq P$*

Definition 34. (Perceived Best Plan) *The* perceived best plan *for agent i from state s_1 is a plan $P = BestP^i(i, s_1)$ such that $Utility^i(i, P, s_1) \geq Utility^i(i, P', s_1)$ for all $P' \neq P$*

4.2.2 Contracts and Deals

So far, I have outlined how an agent can individually achieve its desires through the execution of plans. An agent might also be able to achieve its desires by contracting

---

[3] The reader may argue that the notions of objective and perceived utilities are special cases of objective and subjective expected utilities in the decision-theoretic sense [von Neuman and Morgenstern, 1944]. I deliberately use the more simplistic notion because it makes the link to argumentation more trivial.

certain actions to other agents. Since agents are self-interested, they would only perform actions for one another if they receive something in return (i.e., if they get actions done for them, resulting in achieving their own desires). A specification of the terms of such exchange of services is a *contract*.

Definition 35. (Contract) *A* contract Ω *between agents A and B is a pair (P_A, P_B) of plans, and a schedule, such that P_i, $i \in \{A, B\}$ is the part of the contract to be executed by agent i according to the schedule.*

A schedule is a total order over the union of actions in the two one-agent plans. As with one-agent plans, I denote by $s_1 \models [\Omega]s_2$ that if the world is in state s_1, then executing the contract Ω moves the world to state s_2. Similarly, the perceived result of the contract by agent i is denoted by $s_1 \models^i [\Omega]s_2$. I denote by $\mathcal{C}$ the set of all possible contracts. I now define the cost of a contract to an agent.

Definition 36. (Cost of Contract) *The* cost of contract $\Omega = (P_A, P_B)$ *for agent $i \in \{A, B\}$ is the cost of i's part in that contract; i.e., $Cost(i, \Omega) = Cost(i, P_i)$*

I define the contract's objective and perceived utilities, denoted $Utility(i, \Omega, s_1)$ and $Utility^i(i, \Omega, s_1)$, and the best contract and best perceived contract, denoted $BestC(i, s_1)$ and $BestC^i(i, s_1)$, analogously to plans above.

I can now define the set of contracts acceptable to an agent.

Definition 37. (Individual Rational Contract) *A contract $\Omega = (P_A, P_B)$ is* individual rational, *or simply* acceptable, *for agent i in state s if and only if $Utility(i, \Omega, s) \geq Utility(i, BestP(i, s), s)$*

A *perceived individual rational contracts* is defined similarly using perceived utilities.

A rational agent[4] should only accept contracts that are individual rational. I denote by $IRC(i)$ the set of individual rational contracts for agent i, and by $IRC^i(i)$ the set of perceived individual rational contracts. On this basis, each agent can classify each possible contract into three sets: *acceptable*, *unacceptable*, and *suspended* contracts. Suspended contracts are contracts for which the agent does not know the result (i.e., for which

---

[4] I.e., rational in the economic sense, attempting to maximise expected utility.

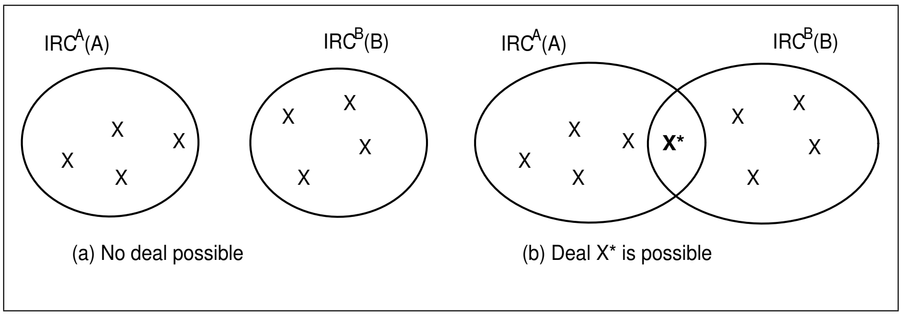


Figure 4.1: Possible and impossible deals

$s_1 \models^i [\Omega]?$), and is hence unable to assess the utilities. If $IRC(i) = \emptyset$, then it makes no sense for agent i to negotiate; i.e., the agent would be better-off doing things individually.

If agents do not change their beliefs, then the set $IRC^i(i) \cap IRC^j(j)$ is the set of possible deals: contracts that are individual rational from the points of view of both agents. Possible deals are those contracts that make both agents (as far as they know) better off than they would be working individually. If $IRC^i(i) \cap IRC^j(j) = \emptyset$, then agents will never reach a deal unless they change their preferences. Figure 4.1 exemplifies two cases. Each oval shows the set of individual rational contracts for an agent. If these sets intersect, then a deal is possible.

4.3 Searching for Deals Through Bargaining

In the previous section, I outlined the main concepts involved in the stage prior to negotiation. The questions that raises itself now is the following: *given two agents, each with a set of individual rational contracts, how can agents decide on a particular deal, if such deal is possible?* One way is to search for a deal by suggesting contracts to one another.

4.3.1 Elements of Bargaining

Negotiation can be seen as a process of joint search through the space of all contracts (i.e., through the set $\mathcal{C}$), in an attempt to find a mutually acceptable contract (i.e., one that belongs to $IRC^i(i) \cap IRC^j(j)$). Furthermore, agents may wish to find a contract that also satisfies some kind of "optimality" criteria. For example, agents may attempt to find a contract that is Pareto optimal, or one that maximises the sum or product of their individual utilities.[5]

One of the most widely studied mechanisms for searching for a deal is *bargaining* [Larson and Sandholm, 2002]. In bargaining, agents exchange *offers* – or *proposals*: contracts that represent potential deals. Of course, it would only make sense for each agent to propose contracts that are acceptable to itself.

Definition 38. (Offer) *An* offer *is a tuple* $\langle i, \Omega \rangle$*, where* $i \in \{A, B\}$ *and* $\Omega \in \mathcal{C}$*, and represents an announcement by agent* i *that* $\Omega \in IRC^i(i)$*.*

During negotiation, each agent may make multiple offers until agreement is reached. At any particular point in time, the offers made constitute the negotiation *position* of the agent: those contracts the agent has announced it is willing to accept as deals. I denote by $\mathcal{O}$ the set of all possible offers (by all agents).[6]

Definition 39. (Position) *The* position *of agent* i*, denoted* $Position(i)$*, is a set of contracts* i *has offered so far, such that at any time, we have* $Position(i) \subseteq IRC^i(i)$

Note that while the set $IRC^i(i)$ is static during bargaining, the set $Position(i)$ is dynamic, since it expands, within the confines of $IRC(i)$, as the agent makes new offers.

A question that raises itself now is: *how does an agent expand its position?* In other words, *given a set of offers made so far, what should an agent offer next?* The answer to this question is what constitutes the agent's bargaining strategy.

Definition 40. (Bargaining Strategy) *A* bargaining strategy *for agent* i*, denoted* Δ^i *is a function that takes the history of all proposals made so far, and returns a proposal to make next. Formally:* $\Delta^i : 2^{\mathcal{O}} \rightarrow \mathcal{O}$*, where* $2^{\mathcal{O}}$ *is the power set of the set of all possible offers* $\mathcal{O}$*.*

One of the key factors in influencing an agent's negotiation strategy is its preferences over contracts. It would make sense for an agent to begin by offering contracts most preferable to itself, then progressively "concede" to less preferred contracts if needed.[7]

---

[5] For more on outcome evaluation, refer to the book by Rosenschein and Zlotkin [Rosenschein and Zlotkin, 1994].

[6] Note that $\mathcal{O}$ is different from $\mathcal{C}$. While the latter denotes the set of all possible contracts, the former denotes the set of all possible agent/contract pairs.

[7] This is commonly known as the monotonic concession bargaining strategy.

Preference, however, is not the only factor that guides strategy. For example, an agent might have time constraints, making it wish to reach agreement quickly even if such agreement is not optimal. To reach a deal faster, the agent might make bigger concessions than it would otherwise. This issue becomes particularly relevant if the number of possible contracts is very large.

A variety of bargaining strategies have been studied in the literature. Such strategies might be specified in terms of a preprogrammed, fixed sequence of offers [Chavez and Maes, 1996] or be dependent on factors observed during negotiation itself, such as the offers made by the counterpart [Axelrod, 1984, Faratin et al., 2002, Zeuthen, 1930], or changes in the availability of resources [Faratin, 2000]. A thorough examination of these strategies is outside the scope of this study. Note, however, that strategies are highly dependent on the interaction protocol and on the information agents have. For example, following a risk-dependent strategy under the Monotonic Concession Protocol when agents have complete information can be guaranteed to lead to a Pareto-optimal agreement [Harsanyi, 1956]. Such a result could not be guaranteed if agents do not know each other's preferences.

4.3.2 Limitations of Bargaining

One of the main limitations of bargaining frameworks is that they usually assume agents have complete and accurate information about the current world state and the results of actions, and are consequently capable of providing a complete and accurate ranking of all possible contracts. If these assumptions are not satisfied, significant problems start to arise. In particular, bargaining could not be guaranteed to lead to agreements that truly maximise the participants' utilities.

To clarify the above point, consider the following example. Suppose a customer intending to purchase a car assigns a higher preference to Volv than Toyot because of their perceived safety of Volvs. Suppose also that this holds despite the customer's belief that Toyots have cheaper spare parts, because safety is more important to them. If this information is false (for example, if Toyots actually perform as good as Volvs on safety tests), then the actual utility received by purchasing a Volv is not maximised. This example is formalised below.

Example 15. *Suppose buyer agent B trying to purchase a car from seller A, such that:*

- *B believes they are in s_1*
- $\mathcal{D}^B = \{safety, cheapParts\}$
- $Worth(B, safety) = 18,\ Worth(B, cheapParts) = 12$
- $s_1 \models^B [do_A(giveVolv), do_B(pay\$10K)]s_2$ *where* $s_2 \models safety$
- $s_1 \models^B [do_A(giveToyot), do_B(pay\$10K)]s'_2$ *where* $s'_2 \models cheapParts$
- $Cost(B, pay\$10K) = 10$

Then B will assign the following utilities:

- $Utility^B(B, [do_A(giveVolv), do_B(pay\$10K)], s_1) = 18 - 10 = 8$
- $Utility^B(B, [do_A(giveToyot), do_B(pay\$10K)], s_1) = 12 - 10 = 2$

Consequently, B will attempt to purchase a Volv. However, suppose that the truth is that:

- $s_1 \models [do_A(giveToyot), do_B(pay\$10K)]s''_2$ *where* $s''_2 \models cheapParts \wedge safety$

In this case, the actual utility of the Toyot contract would be:

- $Utility(B, [do_A(giveToyot), do_B(pay\$10K)], s_1) = 12 + 18 - 10 = 20$

Hence, this lack in B's knowledge can lead to negotiation towards a suboptimal deal.

Another case based on the example above is when B does not know about the safety features of cars of make Honda. In this case, B would assign value "?" to Honda contracts, and would be unable to relate it preferentially to Toyots and Volvs. If Honda's were indeed cheaper, and offer both safety and good spare part prices, agent B would be missing out, again.

What I have just demonstrated is that if agent preferences remain fixed during negotiation and their beliefs are inaccurate, then they may fail to reach deals that maximise their utility. This can be generalised to the following trivial result.

Proposition 1. *In bargaining between agents i and j, the actual best reachable deal is the best deal acceptable to both agents according to their perceived preferences.*

Proof. Let us denote the actual best deal by *BEST*(i, j). This deal lies in the set $IRC(i) \cap IRC(j)$. But since agents make their decisions based on their perceived contract utilities, each contract $\Omega \notin IRC^i(i) \cap IRC^j(j)$ is unacceptable for at least one agent, and hence will never be selected as a deal. This means that the actual best reachable deal through bargaining is in the set:

$$IRC^i(i) \cap IRC^j(j) \cap IRC(i) \cap IRC(j)$$

Now, if

$$\mathit{BEST}(i, j) \in ((IRC(i) \cap IRC(j)) \backslash (IRC^i(i) \cap IRC^j(j)))$$

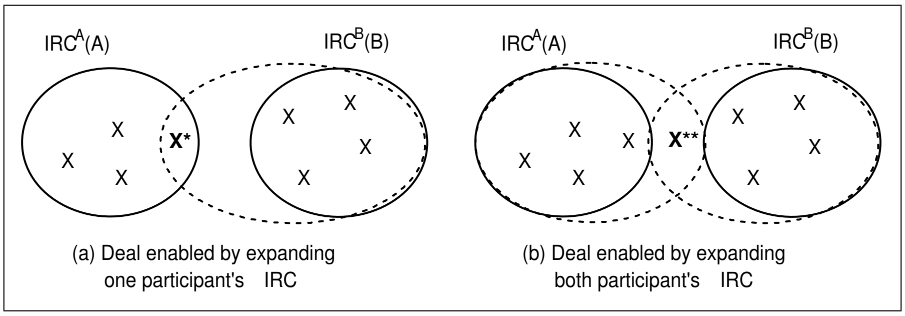
then the agents will never reach *BEST*(i, j). The same thing may apply for the actual second best deal, and so on, until a deal within $IRC^i(i) \cap IRC^j(j)$ is reached. □

This straightforward result demonstrates clearly that as long as agent preferences are inaccurate, they might miss out on better deals. Note, however, that this does not give us an indication of how good or bad the best perceived deal is.

4.4 Argument-based Negotiation

In the previous section, I explored how bargaining can be used to search for a deal on the basis of fixed agents' preferences over contracts. I showed that there are circumstances in which bargaining fails to achieve a deal, or leads to a suboptimal deal. In this section, I explore argument-based approaches to negotiation and relate it to bargaining.

As mentioned earlier, if $IRC^i(i) \cap IRC^j(j) = \emptyset$, then agents will never reach a deal unless at least one of them changes its perceived set of individually rational contracts. Figure 4.2 shows two cases where initially no deal was possible because the agents' individual rational contract sets did not intersect, but a deal is enabled by changes in the set of individual rational contracts. In Figure 4.2(a), a deal is enabled when agent B's perceived IRC set changes such that contract X^* becomes acceptable. In Figure 4.2(b), both agents IRCs change, making deal X^{**} (which initially was not acceptable to either agents) mutually acceptable. Changing $IRC^i(i)$ requires changing agent i's preferences, which in fact requires change in i's beliefs. *Argumentation* is a way to enable agents to rationally influence such beliefs through rational dialogues.



(a) Deal enabled by expanding one participant's IRC

(b) Deal enabled by expanding both participant's IRC

Figure 4.2: Changes in perceived individual rational contracts

The benefit of argumentation is apparent in human negotiations. Humans form their preferences based on information available to them. As a result, they acquire and modify their preferences as a result of interaction with the environment and other consumers [Lilien et al., 1992]. Advertising capitalises on this idea, and can be seen a process of argumentation in which marketers attempt to persuade consumers to change their preferences over products [Slade, 2002]. In negotiation, participants are encouraged to argue with one another and discuss each other's interests. This enables them to jointly discover new possibilities and correct misconceptions, which increases both the likelihood and quality of agreement [Fisher et al., 1991]. Computational agents may realise a similar benefit if they are able to conduct dialogues over interests during negotiation.

4.4.1 Elements of Argument-Based Negotiation

Argument-based negotiation (ABN) extends bargaining-based protocols. Therefore, concepts such as offers and positions are also part of ABN. In addition, agents can exchange information in order to influence each others' beliefs. As a result, they influence each others' negotiation positions and set of acceptable contracts. Therefore, the first step towards understanding how preferences over contracts change is to understand the different ways influence on beliefs may take place, and how such influence affects the utility an agent assigns to a contract.

Recall that the utility of contracts and plans are calculated by agent i based on the following definition, which merges the definitions of plan and contract utility.

Definition 41. (Utility of Plan or Contract) *The utility of contract or plan* X *for agent* i *from state* s_1 *is defined as:* $\mathit{Utility}^i(i, X, s_1) = \mathit{Worth}(i, s_2) - \mathit{Cost}(i, X)$ *where*

$s_1 \models^i [X]s_2$

From the definition, it is clear that the utility of a contract or plan (a) increases as the the perceived worth of the resulting state increases, and (b) decreases as the perceived cost of carrying out that contract or plan increases. Since I assume that perceived costs are accurate, I concentrate on how changes in perceived worth of state s_2 affect the utility. According to definition 30, the worth of state s_2 depends on the set of desires from $\mathcal{D}^i$ that are satisfied in s_2.

Based on this understanding, I can now enumerate how changes in beliefs can influence the perceived utility of a contract or plan. I dub these changes **C1**, **C2** etc.

C1 *Learn that in s_1, X results in a state other than s_2*:

Description: Agent i learns that $s_1 \models^i [P]s_2'$ where $s_2' \neq s_2$.

Effect: This may trigger a change in the worth of X's result, which then influences the utility of X, as follows:

1. If $\mathit{Worth}(i, s_2') = \mathit{Worth}^i(s_2)$, then the utility of X remains the same;
2. If $\mathit{Worth}(i, s_2') \geq \mathit{Worth}(i, s_2)$, then the utility of X increases;
3. If $\mathit{Worth}(i, s_2') \leq \mathit{Worth}(i, s_2)$, then the utility of X decreases;

Example: A traveller who knew it was possible to travel to Sydney by train learns that by doing so, he also gets free accommodation with the booking. As a result, his preference for train travel increases. Hence, this is an example of the second effect described above.

C2 *Learn that it is in a different state*:

Description: The agent learns that it is not in state s_1 as initially thought, but rather in state s_1', where $s_1' \neq s_1$.

Effect: Two things might happen:

1. If the agent believes that in this new state, X has the same result, i.e. that $s_1' \models^i [X]s_2'$, then the perceived utility of X remains the same.
2. If the agent believes X now results in a different state, i.e. that $s_1' \models^i [X]s_2$ where $s_2' \neq s_2$, then the utility of X changes as in the three cases described in **C1** above.

Example: A traveller who was planning a conference trip learns that the conference has been cancelled. Now, flying to Sydney will no longer achieve his desire to present a research paper.

C3 *Learn a new plan*:

Description: Agent i, which did not know what plan X results in, i.e., $s_1 \models^i [X]?$, now learns that $s_1 \models^i [X]s_2$.

Effect: X moves from being suspended to having a precise utility. If X is a contract, it gets classified as either acceptable or unacceptable.

Example: A car buyer did not know whether a car of make Honda has airbags. After learning that they do, he can now calculate the utility of this car.

C4 *Unlearn an existing plan*:

Description: Agent i discovers that some X actually does not achieve the expected resulting state, i.e., that $s_1 \models^i [X]?$.

Effect: The utility of X becomes undefined, and X becomes suspended.

Example: A traveller might find out that merely booking a ticket does not achieve the state of being in Sydney.

As a result of a perceived utility change, the relative preferences among various plans and contracts *may* change. A change in preference change may not take place if the agent's perceived utilities of contracts does not change at all, or if utilities do not change sufficiently to cause a reordering of preferences.

Note that what I described above is the effect of a belief change on the utility of a single contract. In fact, each belief change may trigger changes in the utilities of a large number of contracts, resulting in quite complex changes in the agent's preference relation. This adds another complication to strategic reasoning in ABN, since agents must reason about such unexpected changes in preferences.

4.4.2 Embedded Dialogues as Means for Utility Change

One might ask: *how does argument-based negotiation enable the above changes in belief and perceived utilities during negotiation?* The interest-based negotiation protocol I presented in Chapter 3 aimed to provide an answer to this question. However, it is worth providing a more abstract informal discussion. Since a rational agent should only change its preferences in light of new information, there must be a way to exchange other information during negotiation. This exchange of information can be seen as an *embedding* of other types of dialogues within bargaining. The idea of embedding has been discussed by philosophers of argumentation. For example, Walton and Krabbe [Walton and Krabbe, 1995, pages 66] provide a classification of main dialogue types, namely: *persuasion*, *negotiation*, *inquiry*, *deliberation*, *information seeking*, and *eristic* dialogues. Embedding is a type of dialogical shift where interacting agents move from one dialogue to another [Walton and Krabbe, 1995, pages 100–102]. During negotiation between two participants, the following shifts to embedded dialogues may take place:

- *Information seeking in negotiation*: one participant seeks information from its counterpart in order to find out more (e.g., a customer asks a car seller about the safety record of a particular vehicle make);

- *Persuasion in negotiation*: one participant enters a persuasion dialogue in an attempt to change the counterpart's beliefs (e.g., a car salesperson tries to persuade a customer of the value of airbags for safety);

- *Inquiry in negotiation*: both participants initiate an inquiry dialogue in order to find out whether a particular statement is true, or in order to establish the utility of a particular contract; a precondition to enquiry is that neither agent knows the answer a priori (e.g., a customer and car seller jointly attempt to establish whether a particular car meets the customer's safety criteria);

- *Deliberation in negotiation*: both participants enter a deliberation dialogue in order to establish the best course of individual or joint action (i.e., the best plan or joint plan), potentially changing their initial preferences (e.g., a customer and car seller jointly attempt to find out the best way to achieve the customer's safety and budget requirements).

In computer science, formal frameworks for dialogue embedding have been presented by Reed [1998] and McBurney and Parsons [2002], and they could be applied to embed the above types of dialogues within negotiation dialogues. In the context of interest-based negotiation, the protocol I presented in Chapter 3 enables agents to "express" the above dialogue embeddings, but more work is required to integrate this protocol with a full-fledged dialogue embedding framework such as that of McBurney and Parsons [2002].

4.4.3 Some ABN Examples

I now list a number of examples, building on example 15, which demonstrate some ways in which preference can change as a result of belief change.

Example 16. *Car selling agent A initiates the following persuasion dialogue in order to get the buyer B to choose the Toyot:*

A: *Don't you know that Toyots actually perform as good as Volvs on major road safety tests?*

B: *Oh really? And it costs the same right?*

A: *True.*

B: *Well, I would rather purchase the Toyot then!*

As a result of argumentation, B now believes that

$$s_1 \models^B [do_A(giveToyot), do_B(pay\$10K)]s_2'' \text{ where } s_2'' \models cheapParts \wedge safety$$

As discussed in example 15, this leads to a more accurate preference. Note that this example involves a belief change of type **C1**, where B changes his expectation about the result of the Toyot contract.

Example 17. *Suppose B did not initially know about the safety features of cars of make Honda. In this case, B would have the following belief:*

$$s_1 \models^B [do_A(giveHonda), do_B(pay\$10K)]?$$

As a result, B would be unable to relate it preferentially to Toyots and Volvs. Suppose B then initiates the information seeking dialogue:

B: *How about that Honda over there?*

A: *Actually Hondas satisfy both your criteria. They are safe, and also have cheap parts. In fact, this one is available for $8K.*

A: *Seems better than both. I'll go for the Honda then!*

In this example, if we have $Cost(B, pay\$8K) = 8$, then as a result of the above dialogue, B can now give a utility valuation for contract $[do_A(giveHonda), do_B(pay\$8K)]$. This will be $12 + 18 - 8 = 22$, which will rank the Honda higher than both Toyots and Volvs. Note that this example involves a belief change of type **C3** for the Honda contract.

Example 18. *Suppose that the seller would still rather sell the Toyot than the Honda, because she wants to get rid of the old Toyot stock. Consider the following dialogue:*

B: *From what you said, I like this Honda. It offers the same features as the Toyot, but is cheaper.*

A: *But did you consider its registration cost?*

B: *It's the same for all cars, so I think it's irrelevant.*

A: *Actually, the government recently introduced a new tax cut of $3K for purchasing locally manufactured cars. This is aimed at encouraging the national car industry.*

B: *Wow! This would indirectly reduce the cost of Toyots because they are manufactured in Australia. I presume this does not apply to the imported Hondas.*

A: *That's correct.*

B: *Aha! Toyot is definitely the way to go then.*

Before the dialogue, B knew that if there was a tax cut for local cars, i.e., if it is in $s_1' \models^i localTaxCut$, then purchasing a Toyot results in an additional worth of 3, i.e., that:

$$s_1' \models^i [do_A(giveToyot), do_B(pay\$10K)]cheapParts \wedge safety \wedge get\$3K$$

But because B initially thought that there is no such tax cut, i.e., that it is in $s_1 \models^i \neg localTaxCut$, the resulting state was not thought to contain $get\$3K$. During the dialogue B finds out that it is in s_1' rather than s_1. As a result, the utility of the Toyot contract becomes $12+18+3-10 = 23$, whereas the utility of the Honda remains $12+18-8 = 22$. Note that this dialogue involves a belief change of type **C2**.

4.4.4 Position and Negotiation Set Dynamics

The examples presented in the previous subsection demonstrate how preferences can change during negotiation as a result of belief and utility changes. Now, the question is: *how can such preference change influence the likelihood and quality of agreement?*

Proposition 2. *Argumentation can influence a negotiator i's set of individually rational contracts.*

This is because changes in utilities may cause existing contracts to leave the set $\mathit{IRC}^i(i)$, or new contracts to enter this set.

Recall from Proposition 1 that the quality of reachable deals depends on the contents of the sets $\mathit{IRC}^i(i)$ (or more specifically, on their intersection) and how they differ from their actual counterparts $\mathit{IRC}(i)$. Hence, changes to $\mathit{IRC}^i(i)$ caused by argumentation could influence the quality of reachable deals. Moreover, argumentation can enable a deal in an otherwise failed negotiation. This happens when the sets of individual rational contracts did not initially intersect.

Proposition 3. *Argumentation can improve the actual quality of the deal reached.*

Proof. Let A and B be two agents negotiating over two mutually acceptable contracts, Ω and Ω'. And suppose that for each agent $i \in \{A, B\}$, the perceived utilities are such that $\mathit{Utility}^i(i, \Omega, s_1^i) \geq \mathit{Utility}^i(i, \Omega', s_1^i)$ whereas actual utilities are such that $\mathit{Utility}(i, \Omega, s_1^i) \leq \mathit{Utility}(i, \Omega', s_1^i)$. This means that contract Ω Pareto dominates[8] Ω' from the point of view of both agents, whereas based on the actual objective utilities, Ω' Pareto dominates Ω. If the agents were bargaining, they would choose Ω. Through argumentation, the beliefs of participants may change such that the perceived utility of Ω' becomes higher than that of Ω for both agents. In this case, Ω' would be chosen, resulting in an objectively better outcome. □

A popular example that demonstrates the above proposition has been presented by Parsons et al. [1998]. The example concerns two home-improvement agents – one trying to hang a mirror, the other trying to hang a painting. They each have some but not all of the resources needed. Even though a deal was possible, the agents could not reach a deal

---

[8] i.e., makes one agent better off without making the other worse off.

because one agent knew only one way to achieve his goals. By engaging in argument, that agent was able to learn that he could achieve his goals in a different way, by using a different set of resources. Thus, the information exchanged in the course of the interaction resulted in the agent learning a new way to achieve his goal (i.e., learning some new beliefs), and so changed his preferences across the set of possible contracts. As much as the above result seems promising, there is a potential downside.

Proposition 4. *Agents can be worse off as a result of argumentation.*

Proof. Similar to Proposition 3 above, except that the agents begin *correctly* preferring Ω', and end up preferring Ω. □

The above proposition states that argumentation can lead to worse outcomes. This may happen if an agent's preference ordering after receiving an argument becomes less accurate. For example, an agent may initially prefer Volv's to Toyot's because he believe Volv's are safer cars. After receiving an argument from the seller about the safety of Toyot's, he changes his mind and prefers Toyot's. However, the seller may be wrong (e.g. she used outdated crash statistics, or did not consider other safety aspects) or may deliberately lie to the buyer. In such case, choosing the Toyot does not lead to the best outcome. Argument-based negotiation does not guarantee that arguments will lead to "better" preferences. Whether arguments are useful would depend on the efficiency of the agents' argumentative abilities, their reasoning capabilities, any time constraints, and whether or not they attempt to deceive each other.

4.5 Conclusions

In this chapter, I initiated an investigation into understanding the relationship between bargaining and argumentation-based negotiation frameworks. I described both types of frameworks using a uniform "vocabulary", and made some intuitions about their differences more precise. In particular, I provided a precise account of how certain types of arguments can influence preferences over contracts. I then showed how the ability to exchange such arguments can help overcome some problems with bargaining. In particular, I have demonstrated that:

- Rational agents *may* change their preferences in the light of new information;
- Rational agents should *only* change their preferences in the light of new information;
- Negotiation involving the exchange of arguments provides the capability for agents to change their preferences;
- Such negotiations could increase the likelihood and quality of a deal, compared to bargaining, particularly in situations where agents have incomplete and/or inaccurate beliefs;
- Such negotiations could also lead to worse outcomes compared to bargaining.

This study paves the way for a more systematic study of strategies in argument-based negotiation. Understanding the possible effects of different types of embedded dialogues can help an agent make decisions about how to argue during negotiation. This also enables studying more complex strategies that result in multiple interrelated changes in utility. For example, a car seller may first attempt to persuade a customer of adopting a new desire towards safety, then attempt to convince him that his current preferred contract does not achieve this desire. Indeed, I capitalise on this compositional nature of strategies (informed by the understanding of argument effects) when I present a methodology for designing negotiation strategies in the next chapter.

Chapter 5

A Methodology for Designing Negotiation Strategies

> *"Strategy without tactics is the slowest route to victory . . .*
> *Tactics without strategy is the noise before defeat."*
>
> Sun Tzu, Chinese Military General (544-496 B.C.)

In this chapter, I introduce STRATUM, an informal methodology for designing strategies for negotiating agents. Then I apply the methodology to characterise some strategies for interest-based negotiation and the Trading Agent Competition.

5.1 Introduction

A central feature of all negotiation mechanisms, be they game-theoretic, heuristic, or argument-based, is that agents have some choice of what they may utter, and possibly when they may make these utterances. Open-cry auction participants, for example, choose both the content of their utterances (within the constraints of the particular auction protocol) and the timing of their utterances; participants in a sealed-bid, single-round auction may only choose the content. In argumentation-based approaches, participants have greater freedom in their choice of the content and timing of utterances. A participant in a negotiation framework therefore faces an important question:

What should an agent say, and when to say it, in a particular negotiation interaction?

A *negotiation strategy* may be defined as a rule or algorithm which provides an answer to this question.

It has been consistently argued that game-theoretic approaches are insufficient for designing strategies in complex domains [Jennings et al., 2001]. To this end, I present a methodology which could guide the design and selection of strategies for agents engaged in negotiation interactions in which game-theoretic approaches cannot be applied.

The rest of the chapter is organised as follows. In the next section, I motivate the need for a strategy design methodology. In Section 5.3, I present the STRATUM methodology for strategy design. Then, in Section 5.4, I demonstrate how the methodology can be applied in the context of interest-based negotiation and the Trading Agent Competition [Eriksson and Janson, 2002]. I conclude the chapter in Section 5.5.

5.2 Rationale: Why We Need a Methodology?

5.2.1 Insufficiency of the Game-Theoretic Approach

Recall, from Chapter 2, that the analysis of negotiation strategies in the automated negotiation literature has been conducted using methods based on game-theoretic analysis, heuristic experimentation and argumentation-based studies.

In game-theoretic analysis, researchers usually attempt to determine the optimal strategy by analyzing the interaction as a game between identical participants and seeking its equilibrium [Harsanyi, 1956, Rosenschein and Zlotkin, 1994, von Stengel, 2002]. The strategy determined by these methods is optimal for a participant, given the game rules, the assumed payoffs and the preferences of participants, and assuming that agents have common knowledge that they are all rational and that participants have no knowledge of one another not provided by introspection. On a further assumption that participants behave according to the assumptions of *rational choice theory*, then this approach can guide the design of the interaction mechanism itself, and thus force economically rational agents to behave in certain ways [Varian, 1995]. Hence, game theory provides mathematically rigorous ways of analysing strategic encounters under relatively controlled settings, and very specific protocols. The results that game theory produces (e.g. about the optimal strategies) are only valid under the assumptions of game theory. In the real world, agents may be resource-constrained, malicious or whimsical, or simply badly-coded, so that participant behaviour may not conform to the assumptions of rational choice theory. In such cases, where the game-theoretic assumptions are not guaranteed, the "methodology" of game theory cannot be applied.[1]

In cases where the tools of game theory cannot be applied (because the assumptions of game theory are not satisfied), heuristics are often devised. Heuristics are rules of thumb that produce "good enough" outcomes, and are mainly based on empirical testing and evaluation. In heuristic-based frameworks, strategies have been proposed which are based on, for example, the underlying utility model, decay functions of factors such as utility and time [Faratin, 2000, Kraus, 2001], or fuzzy modelling of the environment [He et al., 2003]. Very specific bargaining strategies have been analyzed, for example, by investigating their optimality [Fatima et al., 2001], their performance in multiple negotiation rounds [Faratin, 2000], or the resulting social welfare [Harsanyi, 1956]. There has been little work, to my knowledge, which looks at strategies in negotiation interactions in a generic way.

Finally, argument-based frameworks are negotiation frameworks that allow agents to exchange, in addition to proposals and indications of their acceptance or rejection, meta-information about them, such as the reasons for their proposal, and for accepting or rejecting them. Strategies have been presented for some of these frameworks also; for example, Sierra and colleagues [Sierra et al., 1998] consider the execution of an authoritarian strategy in which an agent makes an appeal to its authority over others in order to exert pressure on its negotiating counterpart. Similarly, Sadri and colleagues [Sadri et al., 2001b] describe agents negotiating over scarce resources who are always willing to share resources which are not currently needed for their own goals, a rule which partly determines the utterances each agent may make in the interaction.

While the game-theoretic approach can produce precise advice to negotiators in controlled settings, there is no "alternative" methodology for guiding designers of strategy

---

[1] Research on bounded-rationality in game theory has started to address some of these issues [Rubinstein, 1997].

in heuristic-based or argumentation-based negotiation encounters. It seems that agent designers have made choices based more or less on their intuition, rather than on a more systematic methodology. It would be of value, therefore, to provide a methodology that gives generic guidelines for strategy designers operating under such protocols. To this end, this chapter takes the first step towards characterising a methodology for designing strategies under complex protocols and relaxed assumptions.

5.2.2 Insufficiency of AOSE Methodologies

In recent years, there has been an increase in research on Agent-Oriented Software Engineering (AOSE) methodologies. A number of AOSE methodologies have been proposed for guiding the design and construction of multi-agent systems, such as GAIA [Zambonelli et al., 2003, Wooldridge et al., 2000], ROADMAP [Juan et al., 2002], Prometheus [Padgham and Winikoff, 2004] and OperA [Dignum, 2004].

One might ask whether AOSE methodologies could provide the answer to the problem of strategy design for protocols that cannot be studied using game-theoretic tools. In my opinion, none of the existing AOSE methodologies can do the job. This is because such methodologies are typically concerned with the process of capturing domain requirements *before* the agent system exists, and then transforming these requirements to multi-agent system specifications. During this process, system designers can follow the methodology as they describe the environment, agent capabilities, roles, relationship structures, interaction protocols, and so on. With respect to multi-agent interaction in self-interested settings, however, users of these methodologies can usually either (i) produce rigid interaction protocols that *prescribe* how agents should behave, by building the system themselves, by assuming that programmers will follow their specification, or by using mechanism design concepts to make it irrational for programmers to deviate from the prescribed behaviour; or (ii) provide flexible protocol descriptions but leave the design of individual agents unguided. In other words, existing AOSE methodologies do not offer guidance to designers of self-interested agents that can negotiate within an existing multi-agent system.

There have been some attempts to use or produce AOSE methodologies in engineer negotiation protocols and/or agents. Fritschi and Dorer [2002] use an AOSE methodology to construct an agent that participated in the Tracing Agent Competition. However, the methodology was used for engineering the overall agent design, while the strategies were still constructed in an ad-hoc manner. Bartolini et al. [2002] present a modular framework for constructing negotiation protocols using declarative rules. However, this time the focus is not on the description of agent behaviour, but rather on the rules of the protocol. Finally, Dumas et al. [2002] propose a formal language for specifying negotiating agent strategies, which combines State Charts [Harel and Naamad, 1996] and Defeasible Logic Programs [Billington, 1993]. However, they still do not provide guidance for *what* strategies to build using this formal language.

The methodology I present here is complementary to existing AOSE methodologies, since it assumes a multi-agent system specification is given (implicitly or explicitly) using some formal language, and provides guidance to an agent designer in programming strategies which operate *within* the given specification. Intuitively, though, the methodology can only be useful if the MAS specification is reasonably flexible to leave room for the programmer's choice of strategy design. This is typical in open multi-agent systems, such as those involving agents designed by different programmers and operating in a flexible electronic market, such as that described in the experimental Trading Agent Competition [TAC, 2003, Greenwald, 2003].

5.2.3 Value Proposition and Scope

A reasonable question to ask now is: Why would a methodology for strategy design be useful? In addition to potentially providing an understanding of the nature of strategy in complex domains, a generic methodology for strategy design should provide assistance to agents (or their designers) in designing new strategies for new domains. Moreover, the methodology could provide a common framework for comparison of strategies across negotiation frameworks, hence guiding designers in selecting between strategies based on the characteristics of a given domain. This is a direct response to the following call presented in the seminal paper by Jennings et al. [2001]:

> "...there are also a number of broader issues, which, to date, have received comparatively little attention. These include ...the development of a best practice repository for negotiation techniques. That is, a coherent resource

> that describes which negotiation techniques are best suited to a given type of problem or domain (much like the way design patterns function in object-oriented analysis and design)." [Jennings et al., 2001, page 212]

To properly address the above call, agent designers need a framework through which they can answer questions such as: Why should an agent adopt an exponential decay function for conceding on its utility rather than a linear concession strategy? Why should an agent choose to include time, trust, or the history of the interaction in the design of strategies? Why should an argumentative agent resort to making threats rather than seeking to learn more about its negotiating counterpart? Should an agent share resources it does not currently need? It is hoped that a general methodology of strategy design (and, eventually, assessment and selection) would provide a means to answer these questions.

To precisely define the scope of the methodology presented in this chapter, I require a characterisation of the various components of a negotiation framework. A negotiation framework can be seen to involve the following [Bartolini et al., 2002]:

1. a **negotiation locale**, which is a communication platform through which agents interact;
2. a **host** (or **monitor**) that manages the negotiation locale and facilitates the negotiation, making sure participants abide by the protocol rules, pay violation fines etc.;
3. a **negotiation template**, which is essentially a language for describing deals;
4. a set of **negotiation rules**, which include:
    - (a) *rules for participant admission*;
    - (b) *rules for checking proposal validity*;
    - (c) *rules for protocol enforcement*, which include rules that specify when agents can make proposals, accept proposals, make other utterances etc.;
    - (d) *rules for updating status and informing participants*, which specify what agents have access to what information and how the overall state of the negotiation (e.g. current highest bid) is updated;

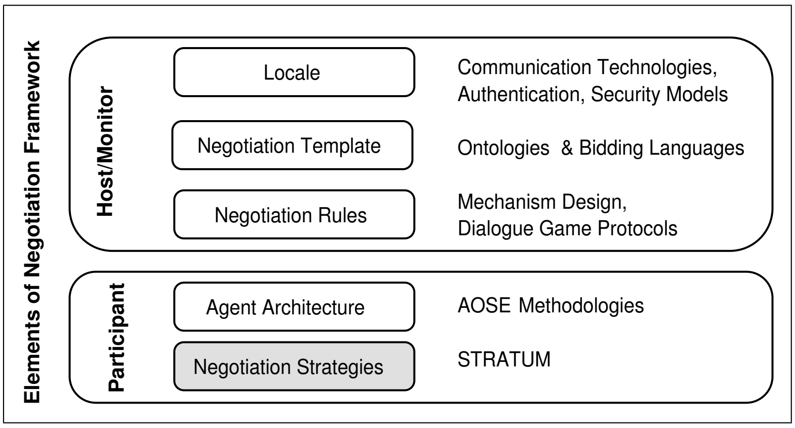


Figure 5.1: Scope of the STRATUM methodology

(e) *rules for agreement formation* specify when an agreement is reached (e.g. an English auction ends with agreement if an acceptable bid is not topped by another bidder for a certain amount of time);

(f) *termination rules* specify under what conditions the negotiation terminates;

5. a number of **participants**, with their internal decision-making components (e.g. their architectures, proposal evaluation logic etc.) and their negotiation strategies;

This chapter is specifically concerned with the design of the *participants* in a negotiation framework, and in particular with their *strategies*. Figure 5.1 depicts the scope of our methodology, which has the specific purpose of helping designers of participating agents specify their agents' negotiation strategies. I am hence not concerned with guiding software engineers in designing the rules of negotiation or specifying languages for describing agreements etc. In fact, I assume a negotiation framework or mechanisms is *given* and specified, formally or informally, in some form. However, since the effectiveness of strategies is highly dependant on the nature of the underlying negotiation framework (rules, templates etc.), the methodology also guides agent strategy designers in distilling framework characteristics that are relevant and useful for strategy design.

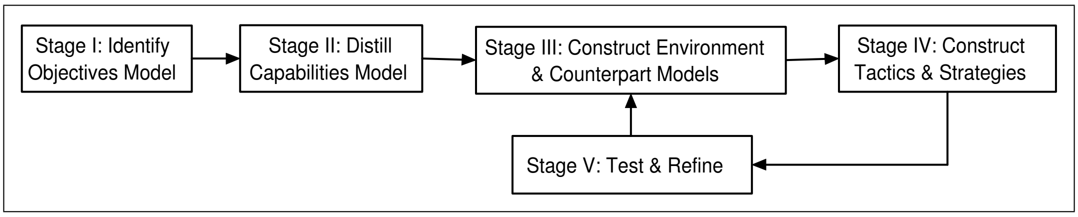


Figure 5.2: Stages of the methodology

5.3 A Methodology for Strategy Design

In this section, I introduce the methodology **STRATUM** (for **STRAT**egy via **U**northodox **M**ethodology)[2] for designing negotiation strategies.

The main function of the STRATUM methodology is to guide the designer of negotiation strategies through the process of acquiring domain information, through to producing modular specifications of tactics and strategies. These specifications should be generic enough to provide flexibility in the underlying implementation details and specific enough to provide thorough and useful guidance for programming these strategies.

STRATUM consists of a number of stages, described in Figure 5.2. In the next subsections, I discuss each of these stages in more detail.

5.3.1 Stage I: Specify Objectives Model

The Objectives Model (OM) model specifies the objectives of the agent. This may be as simple as “maximise own expected utility,” or involve a more complex specification of the types of goals the agent needs to achieve in the world (e.g. to win a battle), the safety conditions it has to maintain (e.g. maintain minimum casualties of one’s own army), the constraints imposed on these objectives (e.g. if there are civilians, then avoid attack) and so on.

Since we are dealing only with purposeful agents, then each party to a negotiation may be assumed to have some objectives which lead it to enter into the negotiation interaction itself. These objectives may be at the highest level of an agent’s stack of goals or they may not be, in which case they may support some other, even higher, goals. Entering

---

[2] The name illustrates that this methodology does not follow the traditional game-theoretic approach. The word *stratum* is the singular of *strata* (a set of layers) and refers to the compositional construction of strategies from simple tactics and capabilities.

into a particular negotiation interaction over certain resources with particular agents at a particular time, will, the agent believes, assist it in seeking to achieve these goals.

An agent's negotiation objectives may be a particular agreement to divide the scarce resources under discussion. Such an outcome, however, is not the only objective an agent may have. Indeed, an agent may enter into an interaction with no intention of seeking a division of the resources in question, but merely to confuse, distract, or otherwise delude the other participants, or even non-participants. An agent may also engage in a negotiation interaction to acquire information about a new domain, as when potential house-buyers participate in auctions in order, not to purchase a house, but to learn about prevailing house prices, or even to learn about the auction process itself. Similarly, an agent may enter into a particular negotiation interaction in order to establish or maintain a larger relationship with the other agents concerned, or to gain knowledge about such agents and their negotiating behaviours. As an example, John Lukacs [Lukacs, 1999] argues that, during May 1940, Prime Minister Winston Churchill of Britain pretended to entertain the possibility of a negotiated peace deal with Nazi-led Germany in order to strengthen his base of political support with key members of his own Conservative Party; Lukacs argues that Churchill was not serious about these possibilities, but in the early days of his Premiership he needed political support from people who were.

Agents may even enter into a negotiation interaction with one counterpart in order to have a stronger negotiation position relative to another counterpart in a separate interaction; business-to-business negotiations often involve such parallel, competitive negotiations [Lilien et al., 1992, Chapter 3]. These objectives are all valid — and, by any definition, rational — objectives from a negotiation interaction. I make no judgment, though, on their wisdom, feasibility, or ethical content.

5.3.2 Stage II: Specify Capabilities Model

A key influence on strategy design will be the interaction capabilities of the agent negotiator — what is the agent capable of doing in the interaction.[3] The second stage of the STRATUM methodology is to distill information about what the agent is capable of doing within the system. This results in the agent's Capabilities Model (CM). Two main types

---

[3] Here we are not referring to the agent's internal capabilities, such as its ability to evaluate an offer.

of capabilities may exist:

1. *Dialogic abilities:* These specify what the agent is capable of uttering to other agents. Such capabilities may be specified in terms of the communication language used and the protocol rules that govern the use of this language. Capabilities may also be constrained through some system of value, such as a system for tracking agents' commitments to check their consistency [Maudet and Chaib-draa, 2003], or a system of reputation or trust that removes agents that behave in a deceptive or misleading manner.

2. *Relevant physical abilities:* An agent may have physical capabilities which may impact its dialogic abilities. Consider, for example, an agent who wants to promise another agent to deliver a fuel tank at a certain time. If the mechanism prohibits agents from lying or decommitting on their promises, then the agent must actually have access to a fuel tank and be capable of moving it to the specified address at the specified time. In other words, the agent can only make the promise (a dialogic ability) if it is capable of following through (a physical ability).

My interest here is in the dialogic abilities, and implicitly in the physical abilities –only insofar as they contribute to these dialogic abilities.

At the bottom level, an agent engaged in a negotiation interaction must be able to make utterances which are legal according to the rules of the protocol. Above this level are some higher-order capabilities, which may, depending on the specific protocol, require utterance of a sequence of locutions to be effected. Typical types of dialogic capabilities needed in a negotiation encounter are listed in Table 5.1. A specific protocol may enable only a subset of these capabilities. Moreover, a specific protocol may enable only certain sub-types of these capabilities. For example, a protocol may enable an agent to provide information proactively (capability C4) about its preferences, but not its beliefs. In any case, the agent designer must have a clear picture of the agent's dialogic capabilities.

Of course, an agent may be said to also have capabilities which are complex combinations of these. For example, the ability to prevaricate [Dunne, 2003] may be constructed from abilities to: request irrelevant information; provide irrelevant, misleading or confusing information; or repeat previous questions or statements. Such capabilities may

	Capability	**Explanation**
C1	*Make proposals*	Proposing potential deals to the counterpart
C2	*Accept proposals*	–
C3	*Reject proposals*	–
C4	*Present information proactively*	An agent may present information in order to influence a counterparty's beliefs, preferences, intentions etc.
C5	*Seek information from a counterpart*	Participants may have varying abilities to extract information from one another, for example, due to differing levels of authority in a social structure.
C6	*Provide information reactively*	Agents may have differing capabilities to provide information to one another; some agents may not be able to lie, or to answer evasively, for example.
C7	*Seek to exert pressure on counterpart*	An agent might be able to threaten or reward other participants for accepting certain deals [Kraus et al., 1998, Sierra et al., 1998], e.g. using authority.
C8	*Retract Commitments*	Agents may have ability to retract commitments or proposals they have made previously. Retraction has been studied, for example, in argumentation theory [Walton and Krabbe, 1995] and in bargaining [Sandholm and Lesser, 2002].
C9	*Withdraw*	Truly autonomous agents should have the ability to withdraw from any interaction at any stage. Agents may also have the ability to threaten to withdraw.
C10	*Re-enter*	Some auction protocols allow agents to withdraw form auctions and re-enter them at a later stage
C11	*Do nothing*	Be passive and wait until conditions change; E.g. until market prices go down, or until counterpart concedes because of his/her time constraints

Table 5.1: Types of Dialogic Capabilities in a Negotiation Encounter

constitute negotiation *tactics* and will be discussed in section 5.3.4 below.

When identifying the agent's capabilities, the designer also needs to take account of constraints on the exercise of any potential capabilities. Such constraints could include:

1. *Interaction Protocol*: The rules of the negotiation interaction protocol may preclude or require certain utterances or certain types of utterances by agents at particular times in an interaction. The FIPA Agent Communications Language, FIPA ACL, for example, requires agent sincerity: only statements which are believed by an agent may be uttered using the *inform* locution [FIPA, 2001]. In principle, such a condition must severely limit the use of FIPA ACL for negotiations.

2. *Values*: As mentioned earlier, the agent's values may preclude or require certain behaviours and so constrain the potential capabilities of the agent. Young [2001], for example, argues that the strategies of human negotiators are significantly influ-

enced by the value of maintaining their social identity.

3. *Resource Constraints*: Time, memory or processing limitations on an agent may limit its capabilities in a negotiation interaction.

5.3.3 Stage III: Construct Environment and Counterpart Model

This stage involves attributing meaningful abstractions to the negotiation environment, including the negotiation counterparts, for the sake of reasoning about them. The result constitutes what I shall refer to as the Environment and Counterpart Model (EM) model. Reasoning about this model, either by the strategy designer at design-time, or by the automated agent at run-time, would provide indispensable guidance to the choice of strategy.

Examples of environment and counterpart models exist in the automated negotiation literature. Gal and Pfeffer [2003], for example, present a technique for *opponent modeling* in order to aid the selection of strategies in a Rock-Paper-Scissors domain. In the context of continuous double auction protocols, He et al. [2003] enable agents to model the "state of the marketplace"[4] and adjust their bidding strategies accordingly using a meta-strategy based on pre-programmed fuzzy rules.

5.3.4 Stage IV: Construct Tactics and Strategies

With the objectives identified, capabilities clarified and environment understood, we can proceed to designing actual strategies. This process constitutes the Tactic and Strategy Design Stage.

Composing Capabilities into Tactics

As stated earlier, we assume an agent enters a particular negotiation interaction over particular resources using a particular interaction protocol with particular counterpart agents at a particular time, in order to achieve its negotiation objectives. In order to achieve these objectives, the agent may attempt to achieve some *sub-objectives*, with the belief that such sub-objectives assist in realizing the overall negotiation objectives. Sub-objectives themselves may be further decomposed into lower-level sub-sub-objectives and so on. We can

---

[4] The state of the marketplace is inferred from observations of supply and demand.

therefore see the process of strategy design in a compositional fashion.[5]

For example, a potential buyer entering into a negotiation with a car-dealer aiming to buy a car may seek to achieve this negotiation objective by realizing each of the following sub-objectives (in sequence):

X. Learning about the alternative models available from the dealer;

Y. Establishing a preference ordering over some or all of these models; and

Z. Getting the cheapest price for the most-preferred model.

The buyer might achieve the first sub-objective by posing a series of questions to the car dealer. The second sub-objective may be achieved by introspection, perhaps involving a process of comparison of the expected utilities of different models [Roberts and Lattin, 1991]. To achieve the third sub-objective, the buyer may seek to achieve two lower-level objectives:

Z.1 Informing the dealer about an offer made by a competing dealer; and

Z.2 Bargaining with the dealer through an exchange of offers.

Each of these sub-sub-objectives may be achieved directly by making a series of utterances, or through decomposition into further sub-objectives and so on.[6] This process guides the design of tactics and strategies. While negotiation objectives are decomposed in a top-down fashion, capabilities are composed in a bottom-up fashion in order to construct tactics and strategies that achieve these objectives. These two processes are abstractly depicted in Figure 5.3. The arrow between the two processes denotes that capability description is informed by the reasonable decomposition of objectives and, vice versa, objectives decomposition is informed by the capabilities available.

---

[5] When describing the negotiation objectives and strategies, I use the terms “objective” and “sub-objective” deliberately in order to distinguish them from “goals” and “sub-goals” described in Chapter 3.

[6] The hierarchical “objectives structure” I have outlined here has a structural similarity to the landmarks theory of conversation protocols of Sanjeev Kumar and colleagues [Kumar et al., 2002]. However, my approach concerns only the objectives of an individual agent and not the joint goals of all participants to an interaction.

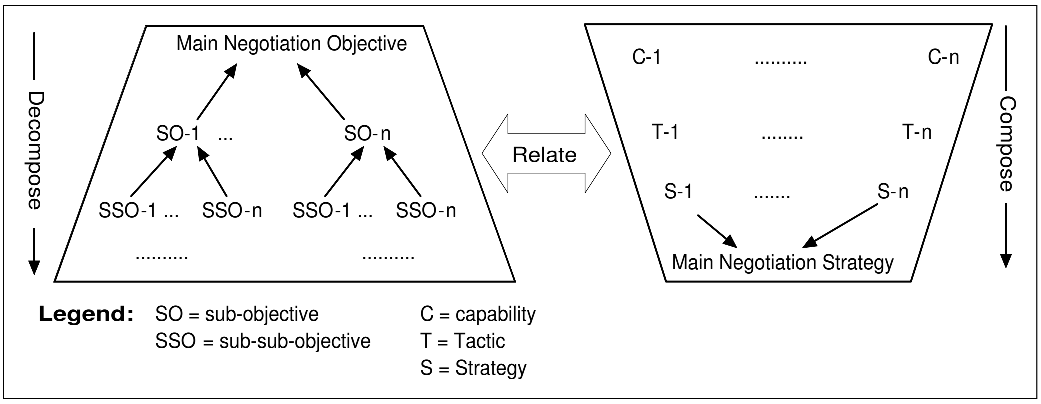


Figure 5.3: Decomposing objectives top-down; composing capabilities bottom-up

I now list a number of possible low-level *tactics* (or equivalently, low-level sub-objectives), which may contribute to an agent's achievement of its negotiation objectives. To my knowledge, no comprehensive list of all possible applicable negotiation tactics is available in the multi-agent literature or in the literature on human negotiation. Therefore, I list those tactics inspired by my experience with the multi-agent negotiation literature as well as informal advice to human negotiation participants [Fisher et al., 1991, Young, 2001, Lewicki et al., 2003].

1. **Seek to change a counterpart's mental state.** One participant in a negotiation may judge it to be in its interests to have other participants change their beliefs (about the environment or about the first participant), or their preferences, goals etc.

2. **Gain a better understanding of a counterpart.** Counterparts may be seeking to mislead a participant about their beliefs, intentions, preferences, constraints etc., or about the domain. An agent may then seek to gain a better understanding of its counterparts' true mental states or constraints. It has been argued [Fisher et al., 1991] that agreement is more likely in negotiation interactions when participants understand each others' interests (desires, preferences, goals etc.) rather than their current positions.

3. **Seek to discuss a particular issue.** By moving the interaction towards particular issues, a participant may be able to frame the problem in certain ways and thus

influence the mental states of its counterparts. A seller of a particular make of car, for example, may seek to turn the topic of discussions with potential buyers towards attributes on which this make of car scores highly.

4. **Seek to avoid discussion.** For the same reasons, a participant may wish to steer discussion away from particular issues.
5. **Seek fast termination.** An agent with time or processing resource constraints might seek a fast resolution or termination of the negotiation.
6. **Seek to delay.** An agent who believes it has greater time or other resources than other participants may seek to delay resolution of the interaction beyond the perceived resource limits of its counter-party [Dunne, 2003].
7. **Resist a counterpart.** An agent may resist attempts by a counterpart to achieve one of the above tactics.

With these main types of low-level tactics in mind, the strategy designer can start specifying specific tactics to be used by the software agent. Specification of these tactics must take into account the actual capabilities of the agent (as discussed above). Tactic description can also be compositional, so a tactic can make use of other tactics already specified. The methodology requires that the designer fill in a *tactic template*, described in Table 5.2, to specify each tactic. A particular instantiation of this template is called a *tactic description*.

Tactic Name	–
Dialogic Objective	–
Method	–
Capabilities & Sub-Tactics Used	–
Rationality Condition	–
Risk	–

Table 5.2: Template for tactic and strategy description

Each tactic description specifies the dialogic objective it is aimed at achieving. The *method* cell contains an informal description of the different steps the tactic involves in order to achieve its dialogic objective. The capabilities and/or sub-tactics used to execute these steps are specified in the following cell. Finally, the rationality condition specifies any constraints on the rational use of the tactic. For example, one may specify that it is only rational to make an offer to a counterpart if this offer is more preferred to the counterpart than any previously made offer. Finally, the designer can list the potential risks of enacting the strategy. This would enable the programmer of the strategy to take these risks into account and provide suitable solutions or precautions.

Composing Tactics into Strategies

The process of designing strategies using tactics is similar to designing tactics using primitive capabilities. We follow the intuitive distinction between *strategies*, which govern an entire interaction or large parts of it, and *tactics*, which govern just a small number of utterances in an interaction. Hence, one tactic may support multiple or competing strategies. For example, asking a direct question may implement a strategy to gather information from another participant or it may implement a strategy to delay resolution of the negotiation, or both. The methodology leaves to the designer the decision about whether to refer to something as a "tactic" or a "strategy." One might argue that this renders the distinction between tactics and strategies redundant. I concede that the boundary between the two is rather vague. Nevertheless, this conceptual distinction may be useful from the strategy designer's point of view. This distinction has proved useful, for example, in the work on natural language generation [McKeown, 1985].

5.3.5 Stage V: Testing and Refinement

The final stage of the methodology is to test the strategy in terms of some appropriate criteria or metrics. This testing may also lead to an iterative process of refinement, based on the new findings.

One way to perform testing is through *empirical* analysis. This would involve running simulations of different dialogues using a variety of strategies. Simulation parameters could be varied in a number of ways: among different strategies operating against a fixed type of counterpart, among different types of low level tactics within a particular generic strategy, in terms of the types of counterparts faced, the information available to agents, or the agents' time constraints. Simulations have been used to analyse bargaining-based

[Faratin, 2000], auction-based [He et al., 2003] and (to a limited extent) argument-based strategies [Ramchurn et al., 2003].

Another way to test strategies is through theoretical analysis. In game-theory, this is normally done using equilibrium analysis, though it is usually aimed at designing the mechanism, not the strategy. For strategies that cannot be studied using traditional game-theoretic techniques, it is often hard to analyse strategies theoretically. Hybrid theoretical/empirical approaches have been used, where the outcomes under different strategies are first generated empirically, then studied analytically using game-theoretic concepts. This approach has been used, for example, by Fatima et al. [2004b] to study bargaining in incomplete information settings.

5.3.6 Summary of the Methodology

The STRATUM methodology is summarised in Figure 5.4. Stages I, II and III first produce the OM, CM and EM. Then Stage IV involves the simultaneous decomposition of objectives and composition of capabilities in order to produce tactics that eventually form sophisticated strategies. Finally, the testing and refinement stage leads to either (i) adjusting the mental-models that the agent or the designer has about the environment and counterparts, or (ii) into repeating Stage IV with different decompositions and compositions.

It is worth noting that the OM, CM and EM can be specified in the system either explicitly or implicitly. When a model is specified explicitly, there is little work to do by the designer. This is the case, for example, when the protocol is specified clearly in terms of declarative rules, or where the counterparts are fully predictable (e.g., in complete information settings). In case the system is not thoroughly specified, however, the strategy designer needs to make these implicit models explicit.

5.4 Applying the Methodology

In this section, I demonstrate, through two case studies, how the STRATUM methodology can be used to aid the design of tactics and strategies in particular negotiation frameworks. The first case study addresses the interest-based negotiation (IBN) framework I presented

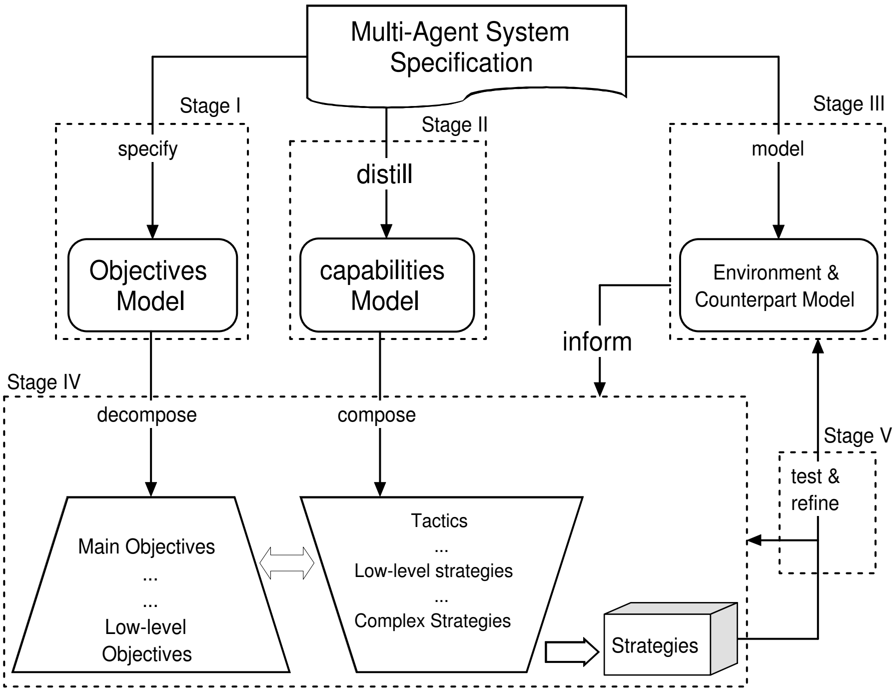


Figure 5.4: Abstract view of the methodology

in Chapter 3. Devising strategies for such argumentation-based negotiation frameworks is a non-trivial challenge, which partly explains the limited research in this area. In the second case study, I explore an existing strategy used in the Trading Agent Competition [TAC, 2003]. I demonstrate that STRATUM captures notions that the designers seem to have included in the design of their strategy.

It is worth noting that I do not undertake a thorough and complete application of the STRATUM methodology in the case studies below. Instead, my aim is to give the reader a feel for how the different stages of the methodology can be applied in particular, relatively rich, negotiation scenarios.

5.4.1 Case Study 1: Interest Based Negotiation

Stage I of the STRATUM methodology requires identifying the agent's objectives in the negotiation. Objectives are specified explicitly in the IBN framework, making the OM easy to document. Each agent attempts to achieve a set of desires and tries to reach a deal that maximises the utility it receives (calculated as the difference between the worth of desires achieved and the cost of the actions needed).

Stage II of the methodology requires defining the agent's capabilities, which constitute the CM. These are also explicitly specified already. In terms of the typology in table 5.1, the IBN protocol presented in Chapter 3 enables, in some form or another, all capabilities except C7 (exerting pressure on a counterpart). Agents can propose, accept and reject deals; they can also make assertions, retract commitments, ask each other for information about each others' beliefs, or ask questions about how they may achieve certain goals, or what a particular goal is useful for. They can also do nothing by uttering the PASS locution. Note, however, that the exact type of capability is somewhat restricted. For example, while an agent can request information about the counterpart's higher-level and lower-level goals –through the REQ-PURPOSE(.) and REQ-ACHIEVE(.) locutions– and about the counterpart's beliefs –using the QUESTION(.) and CHALLENGE(.) locutions– it is not able to ask the counterpart about the reason for believing a certain planning rule.

Moving to **Stage III** (constructing environment and counterpart model EM), we first know that both agents use the same underlying reasoning mechanisms. This is an important assumption, since if the counterpart does not have an explicit representation of underlying goals, for example, then it does not make sense to ask that counterpart of the purpose of a particular request. In IBN, an explicit model of the counterpart is available at any time through its commitment stores. One may also attempt to *guess* other information about the counterpart by inferring its beliefs, goals etc. from its utterances. This would benefit from the assumption that all agents have symmetric reasoning abilities. But for the purpose of this discussion, commitment stores are sufficient. Figure 5.5 summarises the mapping between the IBN framework and the first three models of the STRATUM methodology.

Stage IV involves defining tactics and strategies from capabilities, in light of some decomposition of objectives. I now give an example of the kinds of analysis involved in Stage IV. Suppose I attempt to design a strategy for my seller agent A_1 to negotiate with another buyer agent A_2. As a strategy designer, I would like to design a particular strategy that contributes to A_1's objective, and I want this strategy to be suitable for a situation where A_2 has already proposed some deal D to my agent, and that D is not acceptable to A_1. One way to achieve A_1's negotiation objective is to get A_2 to accept another deal D'

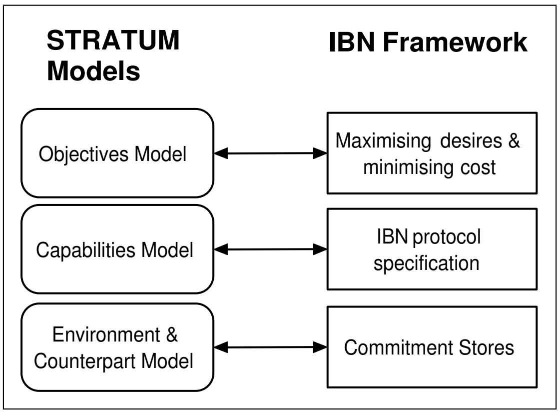


Figure 5.5: Methodology models in the IBN framework

which is preferred by A_1. The most trivial strategy to achieve this is to offer D' to A_2. This is clarified in the following typical bargaining dialogue:[7]

A2: $\mathsf{PROPOSE}(A_2, A_1, D)$
A1: $\mathsf{REJECT}(A_1, A_2, D)$
A2: $\mathsf{PASS}(A_2)$
A1: $\mathsf{PROPOSE}(A_1, A_2, D')$

If the above attempt fails, the only potential solution is to propose other alternative deals D'', D''' etc. However, IBN enables alternative ways to reach a solution, by providing agents with a wider set of dialogic abilities. Reasoning about how these dialogic capabilities may influence the outcome must requires an understanding of the effect of these capabilities on preferences. The analysis presented in Chapter 4 contributes to this by making the link between belief change and preference change more explicit, and hence gives guidance for the design of IBN strategies. Knowing that in IBN, both agents are assumed to have identical reasoning models, we can make the following observations about the counterpart:

– A_2 might have rejected D' because it has some goal g and it believes that D' does not achieve g;

– If A_2 is persuaded that D' does actually achieve g, then it might accept D' (subject to other issues, such as the costs involved);

---

[7] Note that this dialogue is expressed using the IBN protocol described in Chapter 3.

Hence, based on the above understanding of the counterpart, we can construct a strategy that attempts to persuade A_2 that D' does indeed achieve its goal(s). But first, we need to explore how the IBN agent capabilities can be composed to achieve such persuasion. Given that A_1 has the ability C5 to seek information about the purpose of a resource requested by A_2, and to present new information about planning rules, we can construct the following pattern to persuade A_2 to accept D':

- A_1 finds out about a goal g that A_2 wants to achieve using resource r, which is part of deal D that A_2 requested;
- A_1 then argues that D' can also achieve goal g;
- A_1 offers D' to A_2;

This strategy can be described in STRATUM as shown in table 5.3. The following dialogue sequence demonstrates how this strategy can be encoded in the IBN protocol:

Strategy Name	S1
Dialogic Objective	Cause the counterpart to intend a deal that is more preferable to me
Method	- Find out what deal the counterpart currently wants, call this deal D - Find out what goal D is intended to achieve; - Assert that D' also achieves that goal; - Offer deal D'
Capabilities & Sub-Tactics Used	- Capability C5 (seek information) - Capability C4 (present information proactively) - Capability C1 (make a proposal)
Rationality Condition	- Deal D' must be acceptable to me, and more preferred to me than deal D
Risk	–

Table 5.3: Description of an example IBN strategy

A2: $\mathsf{PROPOSE}(A_2, A_1, D)$

this leads to inserting $\mathsf{int}(r)$ for each $r \in \textit{Resources}(D)$ to A_2's commitment store $\mathcal{CS}(A_2)$

A1: $\mathsf{REJECT}(A_1, A_2, D)$

A2: $\mathsf{PASS}(A_2)$

A1: REQ-PURPOSE(A_1, A_2, r)
where $r \in D$
A2: ASSERT$(A_2, A_1,$ instr$(r, g))$
this leads to int(g) being inserted to $\mathcal{CS}(A_2)$
A1: ASSERT$(A_1, A_2,$ prule$(r_1 \wedge \cdots \wedge r_n \rightarrowtail g))$
A2: PASS(A_2)
A1: PROPOSE(A_1, A_2, D')
where $r_1, \ldots, r_n \in D'$

If the strategy works, the above dialogue will be followed by an acceptance from agent A_2 using the utterance ACCEPT(A_2, A_1, D'). The following is a natural language description of a dialogue that uses the above strategy.

A1: *Why do you need the car?*
A2: *To travel to Sydney.*
A1: *You can also go to Sydney by flying there.*
A2: *But this would be even more expensive.*
A1: *We're in the low-season, so I can book you a flight for only $250.*[8]
A2: *That's great. Go ahead and book it please.*

In the above example, agent A_1 simply presents information about an alternative way of achieving the goal of going to Sydney. Whether A_2 accepts this alternative might depend on other issues, such as whether A_2 has petrol vouchers, or whether A_2 wishes to visit a friend who lives on the way to Sydney. Therefore, a more aggressive strategy would be to try to make the driving alternative itself less appealing. As a result, this might make the flying option more preferable, hence increasing the likelihood of its acceptance. This new new approach can be described by the following pattern:

- A_1 finds out about a goal g that A_2 wants to achieve using resource r, which is part of deal D that A_2 requested;
- A_1 then argues that D' can also achieve goal g;
- A_1 persuades A_2 that its previous deal D does not achieve goal g;

---

[8] Of course, this dialogue assumes that A_1 prefers to sell a ticket for $250 than hire out a car for $400.

- A_1 offers D' to A_2;

Note that this strategy is similar to strategy S1, except that it requires A_1 to persuade A_2 to abandon D. So before we specify the new strategy, we need to specify a tactic for achieving such persuasion. One way to do so is to show a "side effect" of D that was not known to A_1 before. This tactic is described in table 5.4, and is self explanatory. We can now describe the new strategy by referring to tactic T1. This strategy is described in table 5.5.

Tactic Name	T1
Dialogic Objective	Cause counterpart to abandon some intended deal D
Method	- Ask the counterpart whether it intends some goal g'; - Argue that D has the undesirable side-effect of precluding g'
Capabilities & Sub-Tactics Used	- Ability to provide information about consequences
Rationality Condition	–
Risk	–

Table 5.4: Tactic description of an IBN tactic

Strategy Name	S2
Dialogic Objective	Cause the counterpart to intend a deal that is more preferable to me
Method	- Find out what deal the counterpart currently wants, call this deal D; - Find out what goal D is intended to achieve, call this goal g; - Assert that D' also achieves that goal; - Cause counterpart to no-longer intend D; - Offer deal D'
Capabilities & Sub-Tactics Used	- Capability C5 (seek information) - Capability C4 (present information proactively) - Tactic T1 to cause counterpart to abandon D - Capability C1 (make a proposal)
Rationality Condition	- Deal D' must be acceptable to me, and more preferred to me than deal D
Risk	- After abandoning deal D, the counterpart may still not accept deal D', for some other reason. As a result, if my agent prefers accepting D to nothing, and it is not possible to persuade the counterpart to intend D again, then my agent is worse off

Table 5.5: Description of an example IBN strategy

The following is an illustration of how strategy S2 can be encoded in the IBN protocol.

A2: $\mathsf{PROPOSE}(A_2, A_1, D)$

this leads to inserting $\mathsf{int}(r)$ for each $r \in \textit{Resources}(D)$ to A_2's commitment store $\mathcal{CS}(A_2)$

A1: $\mathsf{REJECT}(A_1, A_2, D)$

A2: $\mathsf{PASS}(A_2)$

A1: $\mathsf{REQ\text{-}PURPOSE}(A_1, A_2, r)$

where $r \in D$

A2: $\mathsf{ASSERT}(A_2, A_1, \mathsf{instr}(r, g))$

this leads to $\mathsf{int}(g)$ being inserted to $\mathcal{CS}(A_2)$

A1: $\mathsf{QUESTION}(A_1, A_2, \mathsf{int}(g'))$

A2: $\mathsf{ASSERT}(A_2, A_1, \mathsf{int}(g'))$

A1: $\mathsf{ASSERT}(A_1, A_2, \mathsf{prule}(r_1 \wedge \cdots \wedge r_n \rightarrowtail \neg g'))$

where $r_1 \wedge \cdots \wedge r_n \subseteq \textit{Resources}(D)$

A2: $\mathsf{PASS}(A_2)$

A1: $\mathsf{PROPOSE}(A_1, A_2, D')$

where $r_1, \ldots, r_n \subsetneq \textit{Resources}(D')$

The following follow-up natural language dialogue illustrates the usage of the strategy:

A2: *I still prefer to drive to Sydney.*

A1: *Are you collecting frequent flyer points?*

A2: *Yes!*

A1: *Because you are dropping the car in another state, you will not be able to get frequent flyer points by hiring this car.*

A2: *Oh! I thought I would. In that case, I prefer to fly and get the points.*

The risk associated with strategy S2, as shown in table 5.5, is clarified in the following variant of the above dialogue:

A2: *I still prefer to drive to Sydney.*

A1: *Are you collecting frequent flyer points?*

A2: *Yes!*

A1: *Because you are dropping the car in another state, you will not be able to get frequent flyer points by hiring this car.*

A2: *Oh! I thought I would. In that case, I don't want to hire a car, and I'd rather stay in Melbourne for my holiday; I hate flying.*

Finally, testing and refinement of the strategies is performed in **State V**. This stage on its own may require a separate study, because of the richness of the protocol and the complexity of possible dialogue sequences. However, I shall briefly discuss some possible directions.

One option is to perform *empirical testing through simulation*. In argument-based negotiation, such types of studies have only been used for very simplistic dialogues [Jung et al., 2001, Karunatillake and Jennings, 2004, Ramchurn et al., 2003], where variations in parameters are relatively easy to enumerate (e.g. by varying the strength of threats and promises based on trust). The lack of extensive research in this area is largely due to the complexity of the protocol, and the fact that no generic formal theory of argument-based interaction protocols exists. The STRATUM methodology has the potential to enable designers of multi-agent systems to follow a more systematic approach when enumerating variations of strategy within richer protocols.

The other option for analysing IBN strategies is to perform *formal analysis* by appeal to the machinerly of the underlying logic. This approach has been used to study simple strategies in persuasion dialogues [Amgoud and Maudet, 2002], as well as in inquiry and information seeking dialogues [Parsons et al., 2002], but not in negotiation dialogues as yet. Torroni [2002] study whether certain strategies lead to dialogue termination. The formal analysis, in this case, is facilitated by studying the way dialogues control the underlying Abductive Logic Program proof theory [Sadri et al., 2002]. This type of formal analysis may not be as simple for the IBN framework, since I did not provide a proof theory for the protocol.[9]

5.4.2 Case Study 2: Trading Agent Competition

In the previous subsection, I demonstrated the use of the STRATUM methodology to *construct* strategies in a particular negotiation framework. In this subsection, I do the

---

[9] This may be a subject of future investigation.

opposite: I *deconstruct* an existing strategy used by an agent participating in the Trading Agent Competition (TAC) [TAC, 2003]. Through this rational deconstruction, my aim is to demonstrate that STRATUM captures notions that the designers of this strategy seem to have reasoned about as they designed their strategy. I hope that this complementary deconstructive exercise would provide further support of the feasibility of the methodology and its consistency with current actual practice. The analysis is based on SouthamptonTAC [He and Jennings, 2003], a trading agent developed at the University of Southampton, and one of the most successful agents in TAC2002.

I now start with a brief description of TAC games.[10] In a TAC game, there are eight software agents that compete against each other in a variety of auctions to assemble travel packages for 64 customers (8 customers each). A package consists of (i) a round trip flight during a 5-day period between TACtown and Tampa; and (ii) a stay at a particular hotel for every night between their arrival and departure. Customer satisfaction is measured in terms of *utility*, and each agent attempts to maximise its customer's utility, as measured against customer preference data that is randomly generated at the start of the game. Agents can also bid for the *optional* entertainment package, which can provide additional utility. There are different types of entertainment options. An individual game lasts 12 minutes and involves 28 auctions. Each component in a package is traded in a different type of auction:

– *Flights* are sold in single seller auctions, for which the ask price is updated randomly every 24 to 32 seconds.

– *Hotels* are traded in 16th price multi-unit English auctions (that is, the winner pays the 16th highest bid). There are eight hotel auctions that close in random order at the end of every minute after the 4th minute. When a hotel auction clears, it allocates 16 rooms to agents that bid the 16th highest prices.

– *Entertainment* tickets are randomly provided to agents, 12 tickets each, at the beginning of the game. Then agents can trade their tickets in a continuous double auction (CDA), where agents can buy and sell at any time before the game closes. Tickets are only useful to a customer if they are for different events, on different

---

[10] For the interested reader, Greenwald [2003] provide a comprehensive description.

dates, and for nights when the customer is in town.

Designing bidding strategies for TAC auctions has proved to be a challenging problem. There are many interdependencies between different kinds of auctions (e.g. flights will be useless if the hotel rooms are not available) and within the same auction (e.g. having an extra ticket for the same entertainment is useless). This means that agents have to solve combinatorial optimisation problems in a very short time [Eriksson and Janson, 2002]. Moreover, there is an inherent uncertainty in the domain (e.g. since flight prices change randomly) and there is no way to predict precisely how other agents will behave.

Let us deconstruct SouthamptonTAC-02 based on STRATUM. **Stage I** (specifying objectives) is straightforward, since the objectives are explicitly stated in the form of the utility formula. Each agent attempts to maximise the overall utility based on the given customers preferences.

Stage II (specifying capabilities) is significantly simpler than that of the IBN framework presented earlier. This is because the protocol allows fewer dialogic capabilities. Agents have the ability C1 of making proposals in the form of bids according to the different auction protocol rules, the ability C2 to accept proposals in the entertainment CDAs, and the ability C5 to seek information from counterparts *indirectly* by observing their bidding behaviour. Finally, agents have the ability C11 (do nothing) by simply not making any bids. Agents are not allowed to leave the game before it is over, nor enter the game after it starts. Moreover, agents have no external physical abilities which may influence their dialogic abilities.

The authors of SouthamptonTAC-02 invested a significant effort into **Stage III** (constructing environment and counterpart model). In the authors own words:

> "Our post hoc analysis of the TAC2001 competition shows that an agent's performance depends heavily on the risk attitude of its opponents." [He and Jennings, 2003, page 221]

This marks an explicit recognition by the authors of the importance of modeling the negotiation environment. To this end, the authors identify three types of environments, based on the prices of hotels: (i) *competitive environments* where the prices of the hotels are very high, (ii) *noncompetitive environments* where agents can get cheap hotel rooms, and

(iii) *semicompetitive environments*. Environment type recognition is performed by monitoring the hotel prices during the game. Recognition is based on *fuzzy pattern recognition* techniques, which classify the environment based on the degrees of membership of the asking price in the fuzzy sets representing the three environment types mentioned above.

It is unclear why the authors did not use the prices of entertainment tickets or flight tickets as a measure of competition. It may be that the authors believe, perhaps based on experience, that hotel bidding behaviour gives a good estimate of the overall sentiment in the environment. Alternatively, the authors may be assuming that hotel closing prices has the most significant effect on the outcome of the game, and so there is no need to worry much about bidding behaviour in the travel and entertainment auctions.

Let us now move to **Stage IV** (constructing tactics and strategies). Given the limited number of dialogic abilities, tactics are mainly characterised by the following two families: *seeking to delay* and *seeking fast termination*. In other words, tactics are mainly time-dependent. The precise way in which agents delay or speed up their buying and selling is what constitutes TAC strategies.

The authors characterise a *risk-averse* (RA) strategy as one where the agent buys a small number of flight tickets at the beginning of the game and bids for hotels according to the situation as the game progresses. A *risk-seeking* (RS) strategy, on the other hand, is one where the agent buys a large number of flight tickets at the beginning of the game, and therefore does not change its customers' travel plans often during the game. The authors state the following observation about the risk-averse strategy, based on their experience in TAC2001:

> "a *risk-averse* agent ... is highly flexible and copes well when there is a significant degree of competition and the hotel prices are high ... In this way, it avoids buying extra hotels which cost extra money. Also, the agent can receive optimal utility by not shortening the stay of its customers." [He and Jennings, 2003, pages 221, 226]

The authors also make the following observation about the risk-seeking strategy:

> "a *risk-seeking* agent ... does well in environments in which hotels are cheap. For example, when a hotel price goes up sharply, a risk-averse agent would

> stop bidding on that hotel (changing the stay to a counterpart hotel or reducing the trip period). In contrast, a risk-seeking agent will insist on bidding for that hotel, although the price is very high. In so doing, it hopes that the price will eventually stabilise (hence the risk) ... (It) is highly effective in noncompetitive environments ... because there is little competition in hotel bidding and the agent can always obtain what it wants." [He and Jennings, 2003, page 222, 226]

It is possible to characterise the above observations, which guided the design of the SouthamptonTAC-02 strategy, using STRATUM tactic templates. The risk-averse and risk-seeking tactics are encoded in tables 5.6 and 5.7 respectively.

Tactic Name	Risk-Averse Tactic (RA)
Dialogic Objective	- Avoid buying extra expensive hotel bookings - Optimise utility by not shortening customers' visits
Method	- Buy small number of flight tickets at the beginning - Bid on hotels as the game progresses
Capabilities & Sub-Tactics Used	- Ability C1 (making proposals/bids) - ability C5 (seek information) by observing other agents' bidding behaviour - ability C11 (do nothing)
Rationality Condition	Environment is competitive
Risk	If the environment is noncompetitive, the agent misses out on opportunities to optimise length of trips

Table 5.6: Encoding risk-averse tactic using the STRATUM template

Stage IV of strategy design does not stop here. The authors make the following additional observation:

> "After our experiences in TAC-01, we came to believe that there is no single best strategy that can deal with all the different types of TAC environment. For example, a risk-seeking agent ... is highly effective in noncompetitive environments. This is because there is little competition in hotel bidding and the agent can always obtain what it wants. On the other hand, delaying buying flights and shortening the stay of customers[11] works well in competitive games. For this reason, SouthamptonTAC dynamically varies its bidding

---

[11] I.e. a risk-averse behaviour

Tactic Name	Risk-Seeking Tactic (RS)
Dialogic Objective	Optimise holiday allocation, and take advantage of lack of competition in order to get the hotels it wants
Method	- Buy large number of flight tickets at the start - Purchase suitable hotels later, without changing flight bookings
Capabilities & Sub-Tactics Used	- Ability C1 (making proposals/bids) - ability C5 (seek information) by observing other agents' bidding behaviour - ability C11 (do nothing)
Rationality Condition	- Environment is noncompetitive - If hotel prices rise, they eventually stabilise
Risk	- If environment gets competitive, hotel prices rise significantly; as a result, one must either pay high hotel prices, or change travel plans and shorten stay, hence wasting travel tickets already bought

Table 5.7: Encoding risk-seeking tactic using the STRATUM template

> strategy according to its assessment of the environment type." [He and Jennings, 2003, page 226]

Based on this observation, SouthamptonTAC-02 uses a *composed* strategy which makes use of the risk-seeking and risk-averse tactics, as well as a *medium-risk tactic* in semi-competitive environments.[12] This can be characterised in a STRATUM strategy template.

Tactic Name	SouthamptonTAC-02 Adaptive Strategy
Dialogic Objective	Maximise utility by adapting to changes in the environment
Method	- If environment is competitive, use the risk-averse tactic - If environment is noncompetitive, use risk-seeking tactic - If environment is semicompetitive, use medium-risk tactic
Capabilities & Sub-Tactics Used	- Risk-Averse Tactic - Risk-Seeking Tactic - Medium-Risk Tactic
Rationality Condition	–
Risk	–

Table 5.8: Encoding the adaptive strategy of SouthamptonTAC-02 using STRATUM

Finally, the real test in **Stage V** (testing and refinement) is the empirical result of the actual TAC2002 competition. TAC2002 results showed that SouthamptonTAC-02 was

---

[12] In this tactic, the agent buys most of the flights earlier and will only change travel plans if a significant improvement can be obtained.

ranked second, with a difference of 0.8% from the top scoring agents.

Since each game involves a wide variety of agents, it is difficult to draw strong comparative results, such as whether the adaptive strategy made a real difference. To get more precise results, the authors set up a number of more controlled experiments in their lab. In particular, they simulated games in different combinations of SouthamptonTAC-02, RA and RS agents played (among the 8 participants). They showed that SouthamptonTAC-02 does best in competitive games, where the number of RS agents is big. The agent also does well in noncompetitive environments, where there are many RA-agents. It turns out that the worst situation for SouthamptonTAC-02 is when all players are like itself, i.e. they are all adaptive. This is due to the fact that all agents switch their strategies (i.e. adapt) at the same time, causing competition to be instantaneously shifted, or for prices to fluctuate constantly.

The authors provide no discussion of their *strategy refinement* part of Stage IV. Presumably, the authors strategically withheld information about their future strategy refinement plans from their competitors. A possible refinement could be to attempt to deal with the problem that arises when multiple identical adaptive agents populate the game. In such cases, adding some randomness to the timing of agents' adaptation (e.g. by allowing an agent to switch its strategy before others do) might enable an agent to overcome the problem mentioned above. Another refinement suggestion is to include other factors when modeling the environment. For example, instead of considering hotel prices as the only measure of environment competitiveness, the strategy designers may take account of competitors' bidding behaviour in entertainment and travel auctions.

5.4.3 Discussion

The two exercises above demonstrate how STRATUM can provide guidance to the process of strategy design in reasonably complex negotiation environments. The examples provide a hands-on feel for how the methodology can be applied, and shows that it is consistent with current practice.

It is worth noting that it may be possible to exploit the hierarchical structure of objectives and strategies in order enable agents to compose strategies at run-time. In fact, this is exactly what SouthamptonTAC-02 does, since it varies its underlying tactic based on observations of the environment. In more complex dialogues, this would require the dependencies between objectives and tactics on different levels to be represented formally in a way that can be processed by a computer program. Then the reasoning required may be made possible using a hierarchical task planner [Erol et al., 1994]. This would enable agents to modify their strategies, or even their objectives, during negotiation. For example, an agent might abandon an objective or a tactic if the agent perceives that this objective or tactic is not currently achievable or is counterproductive because the negotiation counterpart is resistant to it. As with any other intentions, the defeasibility of objectives in a computational agent requires some mechanism for intention-reconsideration [Schut and Wooldridge, 2001].

5.5 Conclusions

In this chapter, I presented the STRATUM methodology for guiding the design of negotiation strategy. STRATUM addresses domains where the nature of the interaction protocol and the (limited) information available to agents makes it impossible to deduce optimal strategies using game-theoretic techniques. To my knowledge, STRATUM is the first attempt to structure the process of strategy construction in non-game-theoretic domains. I demonstrated how the methodology can be used through a case study of strategy design for interest-based negotiation. I also illustrated that the strategy is consistent with a reconstruction of a strategy used in the Trading Agent Competition.

I have only attempted to use STRATUM in two particular negotiation domains. Therefore, it is not yet clear whether the methodology is suitable for *any* negotiation domain under a variety of protocols. Certainly, a number of additional attempts must be performed before it is possible to reach a confident assessment of STRATUM's applicability. And it is no doubt that these attempts will provide insights that help refine the methodology and found it in both negotiation theory and practice. Eventually, a theoretically well-founded and empirically well-tested methodology for studying negotiation strategies would provide clues similar to Ariel's Rubinstein's view of the usefulness of modeling to economics: *"Models of economic theory are meant to establish "linkages" between the concepts and statements that appear in our daily thinking on economic situations"*

[Rubinstein, 1997, page 191].

Finally, it is also unclear whether STRATUM would integrate easily with AOSE methodologies such as OPERA [Dignum, 2004] or ROADMAP [Juan et al., 2002]. If this is possible, then STRATUM may provide a complementary layer to these methodologies.

Chapter 6

Negotiation for Mobile Collaboration Support[1]

> *"Take time to deliberate;*
>
> *but when the time for action has arrived,*
>
> *stop thinking and go in."*
>
> Napoleon Bonaparte, 1769-1821

In this chapter, I present an application of automated interest-based negotiation to mobile user coordination.

6.1 INTRODUCTION

In Chapter 3, I presented a framework for interest-based negotiation (IBN), and then elaborated on this framework by comparing it to bargaining (in Chapter 4) and by exploring the process of constructing strategies within it (in Chapter 5). In this chapter, I present yet another elaboration through a pilot application that makes use of the IBN framework.

In summary, the application presented in this chapter involves the use of concepts of IBN in order to support collaborative activities among mobile users. The scenario involves two users, each carrying a mobile device, such as a personal digital assistant

---

[1] The narrative development and prototype implementation presented in this chapter are a result of work done jointly with Connor Graham, Anton Kattan, and Fernando Koch.

(PDA). The user specifies a certain task to achieve, and the agent may negotiate collaborative activity with agents representing other users. Agents take advantage of the hierarchical task representation used in IBN in order to guide their negotiation towards successful coordination.

The demonstration reported here is not aimed to be a comprehensive application of IBN. Rather, it serves two purposes. First, it serves as a preliminary investigation of the application of automated negotiation technology to mobile coordination. To my knowledge, there has been no earlier attempt to use automated negotiation techniques in this context. Second, and more importantly, the application demonstrates that the IBN specification is comprehensive enough to form a basis for implementing flexible negotiation applications. This provides a form of verification of the operational adequacy of the IBN framework presented in Chapter 3.

Having summarised the motivation behind the chapter, I take a more traditional approach to telling the story in the remainder of the chapter. I begin by discussing the characteristics of mobile coordination in the next section. This helps clarify the features required by potential supporting technology. In Section 6.3, I argue that IBN provides a set of mechanisms that provide some of these features, and I present a conceptual framework and prototype implementation for using IBN to mobile coordination. The chapter concludes in Section 6.4.

6.2 Problem: Impromptu Coordination while Mobile

The use of mobile computing devices in everyday life is increasing largely due to the advancement of enabling technologies [Sadeh, 2003] as well as increasing efforts to make the technology more usable [Weiss, 2002]. In order to cope with the dynamism resulting from user mobility during task execution, providing context-sensitive support becomes crucial. Dey [2001] describes context as “any information that can be used to characterize the situation of an entity. An entity is a person, place or object that is considered relevant to the interaction between a user and an application, including the user and the application themselves.” For the purpose at hand, context-sensitive support includes performing background computation to identify the situation, in order to take action on behalf of the user, and to represent appropriately to the user the state of any such external interaction.

Mobile users do not operate in isolation of other users. Hence, users often need to interact in order to manage their activities. For example, a foreman on a construction site might need to coordinate with engineers and those responsible for gangs of workers [Luff and Heath, 1998], or a student might contact a colleague to organise a study group. Interaction may be aimed at seeking information, coordination, and so on. In this chapter, I am concerned with forms of interaction in which multiple users, with differing agendas and interests, may realise opportunities for useful synchronisation of their activities. This interaction occurs in a dynamic environment and hence automated support for this interaction has to be context-sensitive, taking into account location, time, and the task(s) at hand. As a result, the coordination is often unplanned, and therefore takes place "impromptu" as users happen to be in a context where coordination opportunities arise.

I begin with some informal observations on the role of technology in facilitating interactions through which multiple users, with differing agendas and interests, may realise opportunities for useful coordination of their activities. To better understand the opportunities for technology intervention, I take the advice of Luff and Heath [1998] and "examine activities in which people engage with others when they are 'mobile' and how various tools and artefacts feature in those activities." To this end, I analyse an informal narrative to distill essential characteristics of mobile use.

Scenario: Supporting Impromptu Coordination

In the settings of interest, the user is mobile, connected, and engaged in complex interactions. This creates an opportunity for technology to support the user. In Table 6.1, I list different levels of support that technology could provide, and compare the extent to which different technologies go. The most basic approach would be to provide connectivity, for example, using mobile telephones. However, when support only takes the form of communication facilitation, users would need to keep track of all changes to their context "in their heads" and manage the complexity of identifying opportunities as events unfold, deal with multiple interaction partners, and so on. This places great cognitive load on mobile users, and it is precisely for this reason that support software such as calendar applications are appropriate tools.

Table 6.1: Levels of support for impromptu coordination

Feature / Technology	Connected while mobile	Represent tasks	Manual task manipulation		Auto task manipulation	
			Individual tasks	Group tasks	Individual tasks	Group tasks
Phone	✓					
Ph/Calendar	✓	✓	✓			
MS Outlook		✓	✓	✓		
All above + automated negotiation	✓	✓	✓	✓	✓	✓

When a mobile phone is endowed with a calendar functionality, the user can *outsource* the storage of large amounts of information about activities (meetings, special occasions etc.) to his/her device. This representation of *individual* activities can then be used to help a user coordinate with others. Applications allowing for group task representation (e.g. Microsoft Outlook with the Exchange Sever) go a step further by providing stationary users with representations of multiple users activities in a globally accessible manner.

One could envisage device support not only through *representation* of individual and group activities, but also *automation* to support the cognitive processes that exploit and manipulate those representations. Such automatic processes would use the available information about the user's context as well as information available about other users in order to automatically negotiate agreements over collaboration and coordination of activities. Through more elaborate examples, in the following section I demonstrate that making explicit and available a representation of users' goals and task structures and some ability to view and configure these, through communication or automatically, can better support impromptu coordination.[2]

I now turn to look in more detail at types of situations where such support could be

---

[2] This representation and configuration, to be successful, would of course have to account for individual and group privacy concerns, along the lines of the notions already adopted in calendar applications, but such issues are outside my current focus.

advantageous - with a view to illustrating the settings in which negotiation can effectively facilitate impromptu coordination. The "mobility" setting naturally creates such opportunities as we shall see. This analysis emerged from discussions in a multi-disciplinary focus group and of a narrative based on a diary of a PhD student renamed Fred, generated over a period of three days. The narrative approach has been used in order to understand individual mobile activities in other projects, such as ActiveCampus [Griswold et al., 2002]. An approach grounded in broader and more systematic data collection would be desirable in the future, c.f. [Isaacs et al., 1996].

Narrative 1. *I realized I had not set up a lift home so I called my wife. I couldn't get through, so I left her a message and asked her to call me when she was close. While waiting for her to reply, I continued work. Then Jack gave me a call to discuss our Wednesday meeting. Jack asked if I could get him a book from the university library, which he needs for an assignment. I declined because I needed more time to finish my work. But Jack happened to be planning to head home to study at the same time I wanted to leave the University. I managed to get myself a lift home by offering to help him out with his assignment, in which case he no longer needed the book.*

Fluidity

Kakihara and Sørensen [2002] describe mobility as having three dimensions: spatial, temporal and contextual. Spatial mobility captures the nomadic nature of mobility, encompassing the mobility of objects, symbols and space itself. Temporal mobility encompasses an objective sense of clock time against an interpretative, individualistic sense of time. Contextual mobility captures other relevant aspects of interaction pertinent to mobility, including people and events. The resultant interaction experienced by mobile individuals is "fluid." Thus "human interaction is becoming ambiguous and transitory. The patterns of social interaction are dynamically reshaped and renegotiated through our everyday activities significantly freed from spatial, temporal and contextual constraints" (ibid, page 5).

Fluidity in mobility suggests that interaction can be rather occasional in mobile use scenarios, since the context in which these portable devices operate changes more frequently than with stationary computers. Thus, well-established, long-term relationships, in which task structures are well-defined and agreed upon, are less likely to be achieved in a mobile world. This is particularly due to the dynamism in resource availability as a result of mobility. Negotiation is one way of reaching temporary agreement in such dynamic settings. In the narrative above, Fred negotiated a lift after realising the opportunity.

Impromptu Coordination

For mobile users, opportunities for collaboration arise more frequently than with static users due to the more diverse forms of context change, such as change in the user's location or the proximity of multiple users. Such opportunities usually cannot be anticipated a priori. In narrative 1 above, Fred being connected to Jack was critical to him being able to capitalise on the opportunity presented by Jack's proximity. The phone did not allow him to predict the possible chances of the success of this opportunistic interaction through a representation of Jack's goals or tasks. Negotiation is a way of dynamically realising and taking advantage of such opportunities.[3]

This theme also relates to the findings of Perry et al. [2001], who build on the study by Luff and Heath [1998] through the examination of 17 mobile workers in the United Kingdom. Specifically, they recommend that technologies supporting mobile workers should "allow more effective planning of activities and flexible allocation of resources."

Heterogeneity

When the modelling of context is to take into account varying location, time, user profiles, tasks, interaction history etc., we are confronted with a much greater variety of agent (and user) types. Each individual agent may achieve tasks in a different way. It is unlikely that information about this heterogeneity will be available a priori. Negotiation is a natural way to exchange information and reach useful agreement or compromise with collaborators (or in collaboration settings) not known before. In the above narrative, Fred's coordination could have been made easier by accessing a representation of Jack's activities or, at the very least, his availability.

---

[3] This characteristic stresses the contrast between the focus of this chapter and the objectives of intelligent scheduling applications.

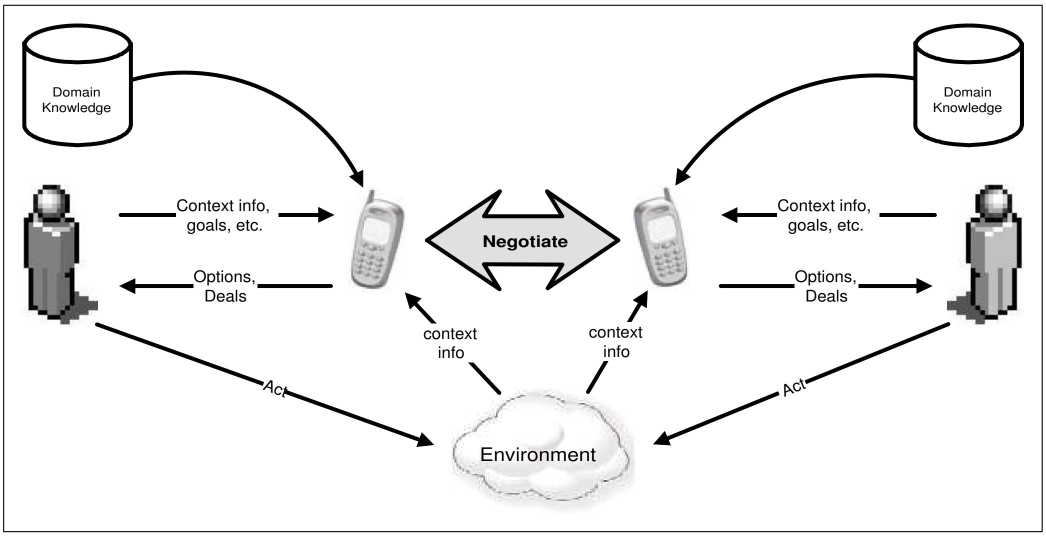


Figure 6.1: Conceptual framework for automatically negotiated mobile coordination

Privacy and Connectivity

Mobile users are constantly confronted with different interaction partners that want to obtain information about them. Users may be unwilling to disclose all the information required to run a centralised algorithm for coordinating joint activity. They may be willing to do so only when interacting with particular partners, or when the they realise the potential benefit of exchanging such information. Negotiation is a natural way to reconcile one's own wish to protect private information with the potential benefit of interacting with others.

6.3 Solution: Negotiating Impromptu Coordination

In the previous section, I argued that impromptu mobile coordination requires the ability to represent information about the tasks of different users, and the ability to interactively process this information. In this section, I illustrate that IBN has the potential to fulfil some of these requirements, and then present a prototype implementation that demonstrates its usage.

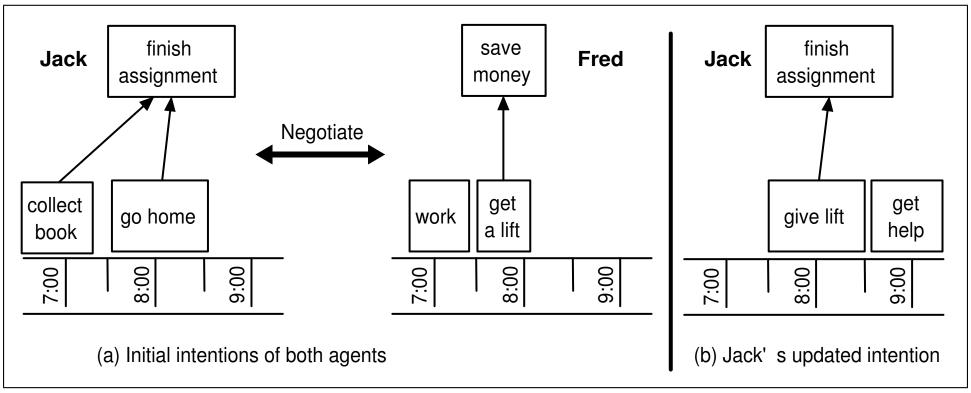


Figure 6.2: An abstract view of negotiation

6.3.1 Conceptual Overview

The conceptual framework for mobile user coordination through automated negotiation is illustrated in Figure 6.1. An agent running on a user's mobile device acts as an intermediary between the user and other potential collaborators. The agent gathers contextual information from the environment (e.g., lecture times, location of user and colleagues) and from the user (e.g., availability, goals). The agent then uses this information, as well as domain-specific knowledge (e.g., procedures for borrowing books from the library) in order to negotiate with agents representing other users. Negotiations are motivated by the user's goals and aim at achieving “deals” with other users. If negotiation results in useful potential deals (e.g., appointment, lunch, lift home), these are proposed to the respective users, who might accept, reject, or modify these deals as they see suitable.

Note that agents may have incomplete information about each others' plans and desires, and about the environment and the appropriate planning procedures within it.[4] In order to address this issue, the automated negotiation framework must enable agents to exchange information about these underlying notions. Since the negotiation mechanism required must exploit representations of users' tasks and goals, interest-based negotiation seems an appropriate choice.

6.3.2 Illustrative Example

In this section, I revisit the narrative introduced in section 6.2 and illustrate how the interest-based negotiation framework presented in Chapter 3 can be used to engineer automated support for the narrative scenario.[5] Recall the situation where Fred fails to get in contact with his wife to secure a lift home. Suppose Fred has a permanent desire for saving money whenever possible, and this is stored in his device in the form of a desire generation rule for which the rule body is always true. One way of saving money is by getting a lift home rather than taking the train or a taxi. This is represented on the device as a planning rule. The two rules are encoded as follows.[6]

$$\mathcal{B}_d^{Fred} = \{\mathsf{True} \Rightarrow \mathsf{d}(save(fred))\}$$
$$\mathcal{B}_p^{Fred} = \{GetLift(X,Y) \rightarrowtail save(Y)\}$$

Recall also that Jack would like to finish his assignment, and in order to achieve that, he believes he needs to collect a book and go home.[7]

$$\mathcal{B}_d^{Jack} = \{\mathsf{True} \Rightarrow \mathsf{d}(finish(assignment))\}$$
$$\mathcal{B}_p^{Jack} = \{Collect(book) \wedge Go(home) \rightarrowtail finish(assignment)\}$$

Now, suppose Jack is unable to collect the book himself (say because he has other tasks to perform) or that he simply prefers someone else to do it for him. Jack's device, which is equipped with interest-based negotiation ability, could automatically attempt to find alternative ways to get a lift home by searching for nearby friends and checking (with their devices) for potential collaboration. When Fred's and Jack's devices detect one another, they initiate a negotiation process.

---

[4] This characteristic corresponds to "heterogeneity" in mobile coordination.

[5] At this point, the reader may wish to refresh some of the concepts presented in Chapter 3.

[6] For clarity, I use predicates instead of propositions. Rules with variables can be seen as placeholders of all their grounded instantiations. Moreover, I use literals beginning with capital letters to denote physical actions (i.e. capabilities).

[7] Note that I do not have an explicit representation of time and temporal constraints. This is mainly to simplify implementation. A realistic implementation would of course need to account for temporal aspects, integrate tasks in a proper calendar application etc.

As soon as Jack's device detects that Fred is in a nearby area, it checks whether Fred is willing to collect the book for Jack. Upon inspection of the request, Fred's device discovers there is not enough time to go to the library to pick the book, say because he has to do some work, as part of another task; i.e., that there is some form of conflict between the two actions. Therefore, Fred's device would reject Jack's request. However, Fred's device could attempt to find out the reason behind the request, with the objective of exploring other potential solutions. After finding that Jack wants the book in order to finish the assignment, Fred's agent could propose an alternative way to achieve the objective, by offering Fred's help. This is not for free, though, as it is in exchange for getting a lift home.

The following is the dialogue sequence just described, encoded using the IBN protocol presented in Chapter 3, between Fred's and Jack's negotiation-enabled mobile devices.

JACK: $\mathsf{PROPOSE}(jack, fred, \mathbf{do}(fred, Collect(book)))$

FRED: $\mathsf{REJECT}(fred, jack, \mathbf{do}(fred, Collect(book)))$

JACK: $\mathsf{PASS}(jack)$

FRED: $\mathsf{REQ\text{-}PURPOSE}(fred, jack, Collect(book))$

JACK: $\mathsf{ASSERT}(jack, fred, \mathsf{instr}(Collect(book), finish(assignment)))$

FRED: $\mathsf{ASSERT}(fred, jack, \mathsf{prule}(Help(assignment) \wedge GiveLift(jack, fred)$
$\rightarrowtail finish(assignment)))$

JACK: $\mathsf{PASS}(jack)$

FRED: $\mathsf{PROPOSE}(fred, jack, \mathbf{do}(jack, GiveLift(jack, fred)),$
$\mathbf{do}(fred, Help(assignment)))$

JACK: $\mathsf{ACCEPT}(jack, fred, \mathbf{do}(jack, GiveLift(jack, fred)),$
$\mathbf{do}(fred, Help(assignment)))$

FRED: $\mathsf{OK}(fred, jack, \mathbf{do}(jack, GiveLift(jack, fred)),$
$\mathbf{do}(fred, Help(assignment)))$

Part (a) in Figure 6.2 shows a sketch of (parts of) the plan structures for Fred and Jack from the narrative 1 above. Part (b) shows Jack's modified plan, which can achieve his goals while also helping Fred.

There are other types of arguments that Fred and Jack could exchange. For example, Jack could ask Fred why he needs the lift, and after finding out that is is purely for saving, he could offer to take him out for dinner instead, in exchange for collecting the book. Not all such arguments seem realistic, and it would be interesting to use the methodology described in Chapter 5 to design strategies informed by theories of human interaction and cooperation in context mobility [Luff and Heath, 1998]. A sensible approach might be to encode a set of typical "attitudes" towards cooperation (represented as negotiation strategies), from which the user could choose for his/her agent, or which could be automatically learned by observing the user's response over a period of time.

6.3.3 Prototype Implementation

The prototype has been implemented based on *3APL-M*, an implementation of *3APL* [Hindriks et al., 1999]. 3APL is a logic-based agent programming language which provides programming constructs for implementing agents' beliefs, goals, and capabilities. It also uses practical reasoning rules in order to generate plans for achieving agents' goals and for updating or revising these plans.[8] Each 3APL program is executed by means of an interpreter that deliberates on the cognitive attitudes of that agent. 3APL-M is implemented using Java Micro Edition (J2ME) [Java 2 Micro Edition], which is a scaled down version of the Java programming language suitable for a variety of mobile devices.

In the initial 3APL version presented by Hindriks et al. [1999], the deliberation cycle is fixed and agents generate their plans by choosing the first applicable rule that matches a particular goal/desire. This means that an agent generates only one plan for each goal, and only generates other plans if the initial plan fails. This approach is not suitable for IBN, since IBN requires the agent to generate all possible plans and then rank them according to their utilities, as has been mentioned in Chapter 3. For this reason, the implementation of 3APL-M is based on an extended 3APL deliberation cycle in which the agent intends the best plan after generating all possible plans (using all applicable practical reasoning rules). This *Deliberative 3APL* implementation is based on an abstract modified reasoning cycle presented by Dastani et al. [2003a].

Figure 6.3 summarises the different languages (and different 3APL tiers) used for the

---

[8] The syntax of 3APL rules is slightly different from those presented in Chapter 3. For planning rules, translation by hand was done. 3APL does not have an equivalent for desire generation rules. This problem was avoided in the current implementation since the desires in the scenario are not conditional.

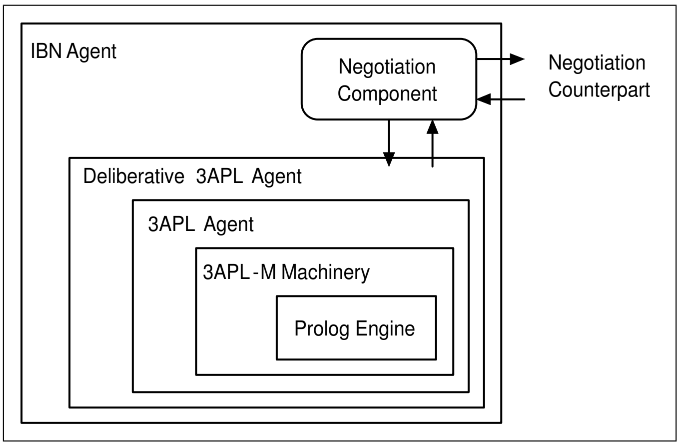


Figure 6.3: IBN agent and the underlying 3APL-M component

implementation. At the core, the 3APL-M basic functionality, such as plan generation, belief manipulation, and so on. These basic building blocks make use of a lightweight Prolog engine in order to perform low-level variable unification. Next, using 3APL-M, a 3APL Agent is built, which is then extended to the Deliberative 3APL Agent by modifying the deliberation cycle to enable thorough plan generation and ranking. Finally, the IBN agent is an extension of the Deliberative 3APL agent. At this stage, the negotiation component, and hence the negotiation behaviour and strategies, are implemented as a Java component *external* to the 3APL reasoning engine. This means that the interaction strategies themselves are not encoded using 3APL structures. This approach significantly simplifies implementation. However, using 3APL at all levels would be advantageous since it would be possible to encode the agent's negotiation strategies themselves as part of its deliberation cycle, and hence enable it to generate *dialogue plans* in the same way as it generates plans for achieving its desires. Such extension would require more theoretical work on encoding complex communication strategies using 3APL, and would likely build on some preliminary work by Dastani et al. [2003b].

Figure 6.4 shows sample screen shots of the implemented system. On the left-hand side is the interface of Jack's mobile device, showing the list of active tasks. The user can add new tasks or edit existing tasks. The right hand side of the figure shows Fred's mobile device, with an instant messaging dialogue with another user named "Bill."

The "Deliberate" button on Jack's device triggers the deliberation process, which is described in figure 6.5. In this figure, the normal arrows denote messages that imme-

You have 3 active tasks

1) Assignment 240
2) Pay bill
3) Arrange airport pickup

Deliberation required 03:35:31

2 buddies are connected

* Fred
* Bill

Bill: Yeh, how did it go yesterday
Jack: Not bad, was good
Bill: Tell me more

lets meet at lunch and will tell you everything

Deliberation required 03:36:31

Figure 6.4: Screen shot of the prototype implementation

diately follow an action by the user. For example, after Fred clicks "Yes" on the top-right screen, a REJECT(.) message is sent. Dotted lines, on the other hand, denote messages exchanged by the agents without an explicit user action. For example, the REQ-PURPOSE(.) message from Fred's agent to Jack's agent did not require a confirmation or explicit request by the user, but is rather triggered automatically as part of Fred's agent's built-in coordination strategy.

In this particular implementation, Jack's agent selects Jack's task to finish the assignment, and by applying the planning rule mentioned above, creates a plan that involves collecting a book and then going home. After searching for nearby friends, Jack's agent discovers Fred's presence and suggests to Jack to ask Fred to collect the book on his behalf. If Jack clicks on "Yes" (part (a) in figure 6.5), his agent initiates a negotiation dialogue with Fred's agent by proposing that Fred pick the book. Fred's device presents this proposal, which Fred rejects by clicking "No" (part (b)). However, Fred's agent attempts to discover the reason behind Jack's request, and based on that information, presents an alternative. Then Fred's agent asks Fred whether he would be interested in getting a lift home (part (c)). Fred clicks “Yes,” after which Fred’s agent sends a proposal to Jack’s agent. Upon receipt, Jack’s agent presents the proposal to Jack, who accepts the proposal by clicking “Yes” (part (d)). This approval is sent to Fred’s agent, which informs Fred of the deal and sends a confirmation back to Jack (part (e)).

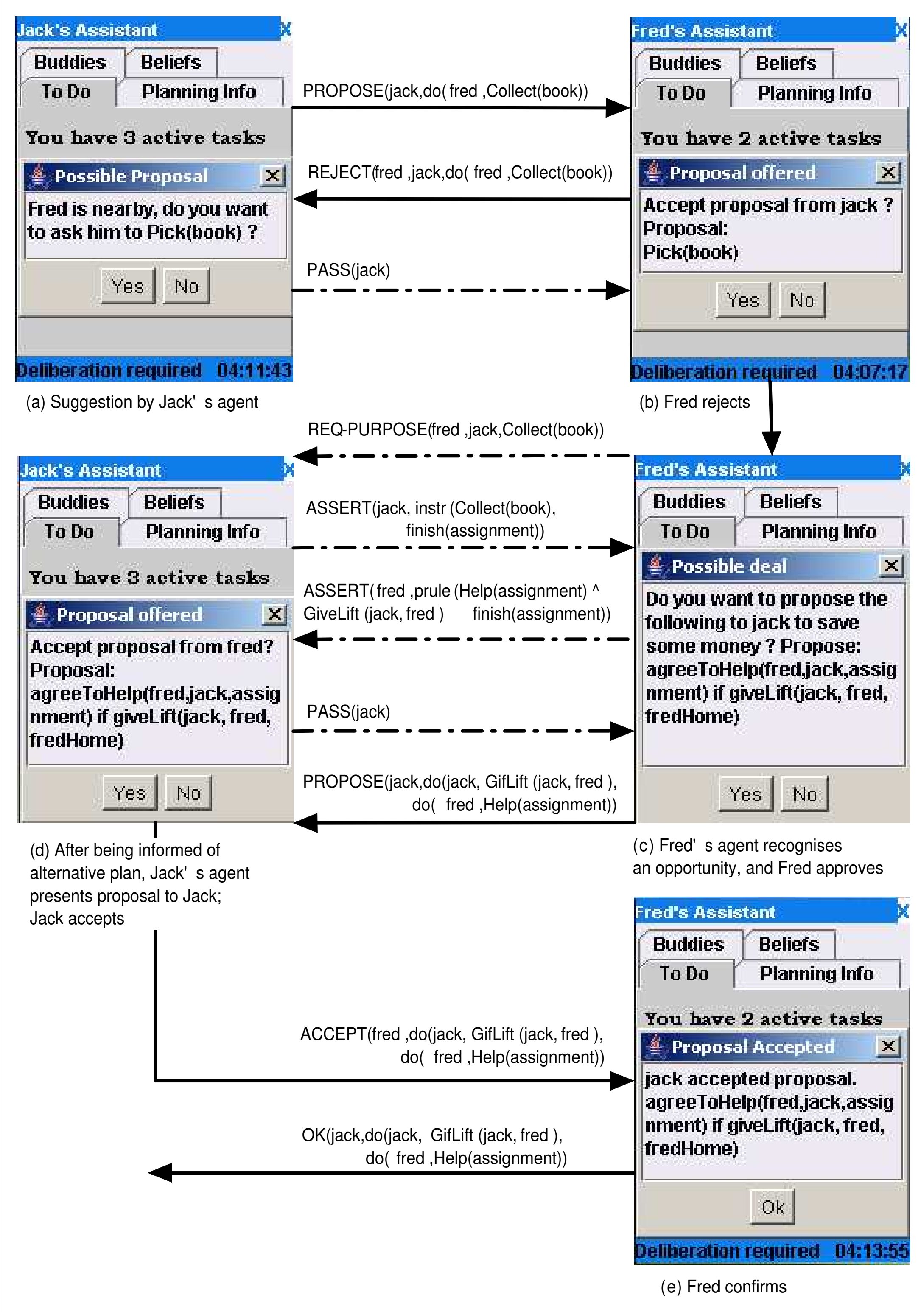
6.3.4 Characteristics Revisited

The framework I presented offers the features required to deal with the characteristics of impromptu mobile coordination discussed in section 6.2 above. In particular, the framework caters for the fluidity encountered in mobile use contexts, since coordination does not assume predetermined and pre-negotiated task structures. Moreover, the focus on tasks and their underlying goals also enables impromptu realisation of opportunities for coordinating activities. By expressing the resources and objectives explicitly, it becomes possible to build technology that processes this information in order to “allow more effective planning and flexible allocation of resources” [Perry et al., 2001].

6.4 Conclusions

In this chapter, I demonstrated a potential application of interest-based negotiation. My aim was to both: introduce a novel application of negotiation technology to mobile coordination, and to demonstrate that the IBN specification given in Chapter 3 is comprehensive enough to form a basis for implementation.

In order to justify the choice of IBN for mobile coordination, I grounded my discussion in current views of mobility in the literature. Through a narrative, I identified key issues of mobile coordination and showed how they may be addressed using negotiation technologies. In particular, I argued for the suitability of negotiation frameworks that represent and manipulate users’ goals. This is because negotiation allows coordination to be task focused, and so no long term coordination structures were necessary (as is required, for example, in the Electric Elves project [Chalupsky et al., 2001]). I then illustrated how IBN, as an automated negotiation that exploits a hierarchical representation of tasks and goals, can provide such support. I then demonstrated a prototype application that demonstrates this.



You have 3 active tasks

Possible Proposal

Fred is nearby, do you want to ask him to Pick(book) ?

Deliberation required 04:11:43

(a) Suggestion by Jack' s agent

You have 2 active tasks

Proposal offered

Accept proposal from jack ?
Proposal:
Pick(book)

Deliberation required 04:07:17

(b) Fred rejects

Possible deal

Do you want to propose the following to jack to save some money ? Propose: agreeToHelp(fred,jack,assignment) if giveLift(jack, fred, fredHome)

(c) Fred' s agent recognises an opportunity, and Fred approves

You have 3 active tasks

Proposal offered

Accept proposal from fred?
Proposal:
agreeToHelp(fred,jack,assignment) if giveLift(jack, fred, fredHome)

(d) After being informed of alternative plan, Jack' s agent presents proposal to Jack; Jack accepts

You have 2 active tasks

Proposal Accepted

jack accepted proposal.
agreeToHelp(fred,jack,assignment) if giveLift(jack, fred, fredHome)

Deliberation required 04:13:55

(e) Fred confirms

Figure 6.5: Screen shots of the scenario run

To my knowledge, the work presented in this chapter constitutes the first application of automated negotiation technology to the problem of mobile user coordination. In particular, it is the first attempt at using negotiation techniques to support non-routine coordination of mobile users. Most existing work on agents for mobile devices focuses on supporting single users [Rahwan et al., 2004d] or collaborative teams executing routine tasks [Chalupsky et al., 2001]. This novel coordination architecture integrates context-aware networked devices, agent-based reasoning, and automated negotiation.

A very important issue worth mentioning is "knowledge elicitation." The scenario implemented here was based on a set of hard-coded rules, based on a relatively naive representation of user tasks. In a realistic 'real-world' implementation, it would be unreasonable to expect users to pre-program their task structures into their devices. Moreover, even if users did manage to encode their task representation, an 'ontology' problem arises, since different users may represent the same tasks differently, and adequate translation between these description is likely to be nontrivial. However, in relatively well-structured domains, such as truck delivery, pre-programming typical task representations may be practical.[9] In less structured domains, it may be possible to use knowledge acquisition techniques based on machine learning [Chipman and Meyrowitz, 1993] to enable agents to automatically build task structures by observing user behaviour. Once these structures are available, IBN may be used to automate interaction that exploits them. After all, much of artificial intelligence is about providing means to engineer knowledge into computer systems and means to computationally manipulate the resulting representations towards useful ends.

---

[9] Some work in knowledge elicitation to initialise automated negotiating agents has been presented by Castro-Schez et al. [2004].

Chapter 7

Conclusion

> *"You seek problems because you need their gifts."*
>
> Richard Bach, 'Illusions'

Recall from Chapter 1 that this thesis began as an attempt to answer the following question:

> *How can we specify intelligent agents capable of negotiating effectively, despite having incomplete and/or inaccurate individual preferences over negotiation outcomes?*

The thesis provides a number of contributions towards answering this question. The contributions revolve around the following specific questions, which I first presented at the introduction to the thesis.

1. What are the appropriate abstractions and decision-making primitives that need to be specified to enable automated agents to conduct rational discussion over their preferences?
2. Can a suitable communication protocol be defined to enable rational discussion over these primitives, leading to flexible, automated negotiation?
3. How would negotiation under such types of protocols compare to more basic "bargaining," where agents simply exchange offers?

4. How may software agent designers reason about the process of designing negotiation strategies for agents participating under flexible negotiation protocols?

Chapter 3 provided an answer to the first two questions. I argued that since agents' preferences are based on their underlying interests, influencing preferences through dialogue should be based on a manipulation of these interests. I then described a generic conceptual framework for interest-based negotiation among autonomous computational agents. This conceptual framework was then used to build a formal framework, grounded in argumentation theory, through which it is possible to specify agents that can modify their preferences over negotiation outcomes in light of new information. I complemented the formal framework with a communication language that can be used by agents to exchange information relating to their interests during negotiation. Then I was able to demonstrate that the communication language and underlying formal framework can express relatively complex negotiation dialogues in which agents change their preferences.

A reasonable question that follows from developing the IBN framework is whether IBN actually does lead to better negotiation outcomes. To answer this question, I compared IBN to bargaining-based approaches to negotiation in Chapter 4 and showed that IBN can either improve or worsen the outcomes depending on how accurate the preferences are after evaluating arguments. This result is based on specific types of arguments; more generic results are likely to require an abstraction of *all* types of possible arguments in negotiation. Such analytical study could benefit from a strong grounding of argumentation-based negotiation in game-theory, a significant challenge in its own right.

The framework presented in Chapter 3 gives a precise account of *what* preference change follows the receipt of different types of arguments, but says nothing about *how* agents decide what arguments to exchange. In other words, the agents' dialogue strategies are not specified. Strategies under rich communication protocols pose non-trivial challenges, and agent designers would benefit from a structured approach to designing their strategies. Hence, to answer the fourth question, I presented a methodology for guiding the design of agent negotiation strategies in Chapter 5. I argued that the methodology is not only applicable to IBN, but also to other complex negotiation domains, such as the Trading Agent Competition. More empirical and conceptual work is required, however, before the scope of applicability of the methodology and its usefulness in different settings, can be established.

Finally, in Chapter 6 I presented an implementation that makes use of the concepts developed throughout the thesis. This implementation highlighted the potential of automated negotiation techniques in providing support to collaborating users. The implementation also served as a proof-of-concept demonstration for IBN, which complements the analytical results presented in Chapter 4. Further validation of IBN may benefit from empirical simulations of different negotiation scenarios. Such evaluation, however, was outside the scope of the thesis.

Recall from Chapter 2 that negotiation frameworks can be classified based on whether they focus on game-theoretic, heuristic-based, or argumentation-based analysis. The contributions of this thesis vary in depth and scope in relation to this classification. The conceptual and formal frameworks for IBN presented in Chapter 3 represent a contribution specific to argumentation-based negotiation and builds on the emerging literature in the field. Chapter 4, on the other hand, provides a comparison of argument-based and game-theoretic approaches. The methodology I presented in Chapter 5 gives a similar broad-based contribution, spanning argument-based and heuristic-based approaches.

It must be noted that IBN is not the only way to implement flexible negotiating agents. IBN uses beliefs, desires, goals and planning rules as the primitives upon which argumentation is based. It may well be that other concepts, such as agent roles, relationships or trust are also appropriate to include. Certainly, in domains such as electronic commerce, the notions of trust and reputation play a crucial role and one may conceive of arguments that make use of these concepts to influence preferences of other agents. Nevertheless, IBN provides a starting point based on a simplified theory of choice and an explicit argumentation-theoretic account of preference update. This attempt may provide a basis for exploring how the primitives of more sophisticated formalisations of decision-making can be exploited through dialogue.

Appendix A

Argumentation for Plan Generation

In this Appendix, I demonstrate that the fixed-point semantics for argumentation is insufficient for expressing a planning process that generates a set of plans that is maximal with respect to the desire worths and action costs.

I first present some formal preliminaries. Finding the acceptable arguments requires reasoning about how arguments "defeat" one another. We will use the argumentation framework presented by [Amgoud and Cayrol, 2002], which extends that of Dung [1995]. This framework defines an abstract argumentation framework as a pair consisting of a set of arguments and a binary relation representing defeat among these arguments. The framework is 'abstract' in the sense that it is not concerned with the internal structure of the individual arguments, but rather in the interplay between different arguments.

Definition 42. *([Amgoud and Cayrol, 2002])*
An argumentation framework *is a tuple* $\langle \mathcal{A}, \mathcal{R}, \gg \rangle$ *where* $\mathcal{A}$ *is a set of arguments and* $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ *is a binary relation representing the defeat relation among arguments, and* $\gg \subseteq \mathcal{A} \times \mathcal{A}$ *is a strict preorder on* $\mathcal{A}$.[1]

Definition 43. *([Amgoud and Cayrol, 2002])*
Let $\langle \mathcal{A}, \mathcal{R}, \gg \rangle$ *be an argumentation framework, and let* $A_1, A_2 \in \mathcal{A}$. *We say that* A_2 attacks A_1 *if* $A_2 \mathcal{R} A_1$ *and not* $A_1 \gg A_2$.

Definition 44. *([Amgoud and Cayrol, 2002])*
Let $\langle \mathcal{A}, \mathcal{R}, \gg \rangle$ *be an argumentation framework, and let* $A_1, A_2 \in \mathcal{A}$ *such that* $A_2 \mathcal{R} A_1$.

---

[1] A relation is a strict preorder if it is irreflexive and transitive.

We say that A_1 defends *itself against* A_2 *if* $A_1 \gg A_2$.

Definition 45. *([Amgoud and Cayrol, 2002])*
Let $\langle \mathcal{A}, \mathcal{R}, \gg \rangle$ *be an argumentation framework, and let* $A, B, C \in \mathcal{A}$ *such that* $B\mathcal{R}A$*. We say that* A *is* defended *by a set of arguments* $\mathcal{S} \subseteq \mathcal{A}$ *if* $\forall B \in \mathcal{A}$*, if* $B\mathcal{R}A$ *and not* $A \gg B$*, then* $\exists C \in \mathcal{S}$ *such that* $C\mathcal{R}B$ *and not* $B \gg C$*.*

We can characterise the set of arguments of $\mathcal{A}$ that are defended by another set of arguments $\mathcal{S}$ through the following characteristic function:

Definition 46. *([Amgoud and Cayrol, 2002, Dung, 1995])*
Let $AF = \langle \mathcal{A}, \mathcal{R}, \gg \rangle$ *be an argumentation framework, and let* $\mathcal{S} \subseteq \mathcal{A}$*. Then the set of arguments* acceptable with respect to $\mathcal{S}$ *can be obtained by applying the function:*

$$\mathcal{F}(\mathcal{S}) = \{A \in \mathcal{A} : \ A \textit{ is defended by } \mathcal{S}\}$$

Note that the set of arguments acceptable with respect to the empty set includes only those that defend themselves against their attackers. We call these *self-defending arguments* and refer to them as $\mathit{SelfDefend}(AF) = \mathcal{F}(\emptyset)$. However, the complete set of acceptable arguments includes those that defend themselves as well as those that are defended by other acceptable arguments. We can obtain these (semantically) by finding the least fixed point of the function $\mathcal{F}$:

Definition 47. *([Amgoud and Cayrol, 2002])*
The set of acceptable arguments *of a finite*[2] *argumentation framework* $AF = \langle \mathcal{A}, \mathcal{R}, \gg \rangle$ *is:*

$$\mathit{Acceptable}(AF) = \bigcup \mathcal{F}^{i \geq 0}(\emptyset) = \mathit{SelfDefend}(AF) \cup [\bigcup \mathcal{F}^{i \geq 1}(\mathit{SelfDefend}(AF))]$$

The above definition corresponds to a so-called *skeptical* (or *well-founded*) semantics [Dung, 1995]. A skeptical agent restricts itself to either defend $\mathit{Acceptable}(AF)$ without making any assumptions at all, or to defending it with the aid of assumptions already justified without assuming $\mathit{Acceptable}(AF)$ to start with. This differs from the semantics defined in terms of *preferred extensions*, which gathers maximal sets of nonconflicting arguments. Hence, each preferred extension reflects a particular *consistent point of view*.

---

[2] Finite in the sense that each argument is defeated by finitely many arguments.

An agent reasoning using preferred-extension semantics would have to select a particular extension to define its beliefs.

In Section 3.4 in Chapter 3, I described the frameworks for arguing about beliefs and desires using skeptical semantics. Skeptical semantics are “cautious” semantics for beliefs, and are hence reasonable. For desires, it also makes sense to use skeptical semantics because we would like the agent to generate desires based on solid justification. It does not make sense for an agent to desire adopt any arbitrary consistent set of desires.

When it came to argumentation-based planning, the skeptical semantics are not suitable, and hence I did not define acceptable sets of instrumental arguments using the fixed-point semantics. The reason behind this is that we would skeptical semantics are *monotonic* in the sense that the agent adds new beliefs or desires on the basis of *already established* beliefs or desires. Hence, there exists a unique extension. In planning, on the other hand, we need nonmonotonicity. Suppose we have four possible instrumental arguments (i.e. plans) P_1, P_2, P_3 and P_4 for achieving different desires. Suppose we first select P_1 for achieving desire d_1. Then to add another plan P_2 for achieving desire d_2, P_2 must be consistent with P_1. Then, to add plan P_3, this plan must be consistent with both P_1 and P_2, and so on. However, we may discover that while P_1 and P_2 are consistent, they conflict with both P_3 and P_4. It may also be the case that P_3 and P_4 are consistent and (together) achieve a higher total *utility* to the agent than the pair P_1 and P_2. Once we discover this, we need to remove P_1 and P_2 and replace them with P_3 and P_4, hence the nonmonotonicity. This is why preferred extensions are more suitable for planning than skeptical fixed-point extensions.

Bibliography

L. Amgoud. A formal framework for handling conflicting desires. In T. D. Nielsen and N. L. Zhang, editors, *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, 7th European Conference (ECSQARU 2003)*, volume 2711 of *Lecture Notes in Computer Science*, pages 552–563. Springer Verlag, Berlin, Germany, 2003. ISBN 3-540-40494-5.

L. Amgoud and C. Cayrol. On the acceptability of arguments in preference-based argumentation. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI 1998)*, pages 1–7, San Francisco CA, USA, 1998. Morgan Kaufmann Publishers.

L. Amgoud and C. Cayrol. A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, 34(1–3): 197–215, 2002.

L. Amgoud and S. Kaci. On the generation of bipolar goals in argumentation-based negotiation. In *Proceedings of the 1st International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2004)*, Lecture Notes in Computer Science. Springer Verlag, Heidelberg, Germany, 2004. (to appear).

L. Amgoud and N. Maudet. Strategical considerations for argumentative agents (preliminary report). In S. Benferhat and E. Giunchiglia, editors, *Proceedings of the 9th International Workshop on Non-Monotonic Reasoning (NMR 2002): Special session on Argument, Dialogue and Decision*, pages 399–407, 2002.

L. Amgoud and S. Parsons. Agent dialogues with conflicting preferences. In J.-J. Meyer and M. Tambe, editors, *Proceedings of the 8th International Workshop on*

Agent Theories Architectures and Languages (ATAL 2001), pages 190–205, Seattle, 2001.

L. Amgoud, N. Maudet, and S. Parsons. Modelling dialogues using argumentation. In E. Durfee, editor, *Proceedings of the 4th International Conference on Multi-Agent Systems (ICMAS 1998)*, pages 31–38, Boston MA, USA, 1998. IEEE Press.

L. Amgoud, S. Parsons, and N. Maudet. Arguments, dialogue, and negotiation. In W. Horn, editor, *Proceedings of the European Conference on Artificial Intelligence (ECAI 2000)*, pages 338–342, Amsterdam, Netherlands, 2000. IOS Press.

Aristotle. *Topics*. Clarendon, Oxford, UK, 1928. (W. D. Ross, Editor).

R. Ashri, I. Rahwan, and M. Luck. Architectures for negotiating agents. In V. Marik, J. Müller, and M. Pechoucek, editors, *Multi-Agent Systems and Applications III: Proceedings of the 3rd International Central and Eastern European Conference on Multi-Agent Systems (CEEMAS 2003)*, volume 2691 of *Lecture Notes in Artificial Intelligence*, pages 136–146. Springer Verlag, Berlin, Germany, 2003.

K. Atkinson, T. Bench-Capon, and P. McBurney. Justifying practical reasoning. In C. R. F. Grasso and G. Carenini, editors, *Proceedings of the Fourth Workshop on Computational Models of Natural Argument (CMNA 2004)*, pages 87–90, 2004.

K. Atkinson, T. Bench-Capon, and P. McBurney. A dialogue game protocol for multi-agent argument over proposals for action. In I. Rahwan, P. Moraitis, and C. Reed, editors, *Argumentation in Multi-Agent Systems: (Proceedings of the First International Workshop (ArgMAS'04): Expanded and Invited Contributions)*, volume 3366 of *Lecture Notes in Computer Science*. Springer Verlag, Berlin, Germany, 2005.

J. L. Austin. *How To Do Things with Words*. Oxford University Press, Oxford, UK, 1962. Originally delivered as the William James Lectures at Harvard University in 1955.

R. Axelrod. *The Evolution of Cooperation*. Basic Books, Inc. Publishers, New York NY, USA, 1984.

K. S. Barber, T. H. Liu, A. Goel, and C. E. Martin. Conflict representation and classification in a domain-independent conflict management framework. In *Proceedings of the third annual conference on Autonomous Agents*, pages 346–347. ACM Press, 1999. ISBN 1-58113-066-X. doi: http://doi.acm.org/10.1145/301136.301227.

C. Bartolini, C. Preist, and N. R. Jennings. Architecting for reuse: A software framework for automated negotiation. In F. Giunchiglia, J. Odell, and G. Weiss, editors, *Proceedings of the 3rd International Workshop on Agent-Oriented Software Engineering (AOSE-2002)*, Berlin, Germany, 2002. Springer Verlag.

T. J. M. Bench-Capon, P. E. S. Dunne, and P. H. Leng. Interacting with knowledge based systems through dialogue games. In *Proceedings of 11th International Conference on Expert Systems and their Applications, Avignon, France*, volume 1, pages 123–130, 1991.

D. Billington. Defeasible logic is stable. *Journal of logic and computation*, 3: 370–400, 1993.

K. Binmore and N. Vulkan. Applying game theory to automated negotiation. In *Proceedings of the DIMACS workshop on economics, game theory and the Internet*, New Brunswick NJ, USA, April 1997. Rutgers University.

C. Boutilier. Toward a logic for qualitative decision theory. In J. Doyle, E. Sandewall, and P. Torasso, editors, *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning (KR-1994)*, pages 75–86, Bonn, Germany, May 1994. Morgan Kaufmann.

J. J. Castro-Schez, N. R. Jennings, X. Luo, and N. Shadbolt. Acquiring domain knowledge for negotiating agents: a case study. *International Journal of Human-Computer Studies*, 61(1):3–31, 2004.

H. Chalupsky, Y. Gil, C. A. Knoblock, K. Lerman, J. Oh, D. V. Pynadath, T. A. Russ, and M. Tambe. Electric elves: Applying agent technology to support human organizations. In H. Hirsh and S. Chien, editors, *Proceedings of the 13th Interna-*

tional Conference of Innovative Application of Artificial Intelligence (IAAI-2001). AAAI Press, 2001. ISBN 0-1-57735-134-7.

A. Chavez and P. Maes. Kasbah: An agent marketplace for buying and selling goods. In *First International Conference on the Practical Appication of Intelligent Agents and Multi-Agent Technology*, 1996.

S. F. Chipman and A. L. Meyrowitz. *Foundations of Knowledge Acquisition: Machine Learning*. Kluwer Academic Publishers, 1993.

A. Ciampolini, E. Lamma, P. Mello, F. Toni, and P. Torroni. Co-operation and competition in alias: a logic framework for agents that negotiate. *Annals of Mathematics and Artificial Intelligence: Special Issue on Computational Logic in Multi-Agent Systems*, 37:28–37, 2003.

B. J. Clement and E. H. Durfee. Identifying and resolving conflicts among agents with hierarchical plans. In *AAAI Workshop on Negotiation: Settling Conflicts and Identifying Opportunities*, number WS-99-12 in Technical Report, pages 6–11. AAAI Press, 1999.

P. Cohen and H. Levesque. Teamwork. *Nous*, 25:487–512, 1991.

J. S. Coleman. *Foundations of Social Theory*. Harvard University Press, Cambridge MA, USA, 1990.

M. Colombetti. A commitment-based approach to agent speech acts and conversations. In *Proceedings of the Workshop on Agent Languages and Conversational Policies*, pages 21–29, 2000.

V. Conitzer and T. Sandholm. Complexity of mechanism design. In A. Darwiche and N. Friedman, editors, *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence (UAI-2002)*, pages 103–110, August 2002.

R. K. Dash, D. C. Parkes, and N. R. Jennings. Computational mechanism design: A call to arms. *IEEE Intelligent Systems*, 18(6):40–47, 2003.

M. Dastani, Z. Huang, and L. van der Torre. Dynamic desires. In S. Parsons, P. Gymtrasiewicz, and M. J. Wooldridge, editors, *Game Theory and Decision Theory in Agent-Based Systems*. Kluwer Academic Publishers, Dordrecht, Netherlands, 2002.

M. Dastani, F. de Boer, F. Dignum, and J.-J. Meyer. Programming agent deliberation: an approach illustrated using the 3APL language. In *Proceedings of the second international joint conference on Autonomous agents and multi-agent systems*, pages 97–104. ACM Press, 2003a. ISBN 1-58113-683-8. doi: http://doi.acm.org.mate.lib.unimelb.edu.au/10.1145/860575.860592.

M. Dastani, J. van der Ham, and F. Dignum. Communication for goal directed agents. In M.-P. Huget, editor, *Communication in Multiagent Systems*, volume 2650 of *Lecture Notes in Computer Science*, pages 239–252. Springer Verlag, Berlin, Germany, 2003b.

A. K. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, 5(1):4–7, 2001. ISSN 1617-4909. doi: http://dx.doi.org/10.1007/s007790170019.

F. Dignum, B. Dunin-Kęplicz, and R. Berbrugge. Agent theory for team formation by dialogue. In C. Castelfranchi and Y. Lespérance, editors, *Intelligent Agents VII: Proceedings of the 7th International Workshop on Agent Theories, Architectures, and Languages (ATAL-2000)*, volume 1986 of *Lecture Notes in Computer Science*, pages 150–166. Springer Verlag, Berlin, Germany, 2000.

V. Dignum. *A model for organizational interaction: based on agents, founded in logic*. PhD thesis, Institute of Information and Computing Sciences, Utrecht University, The Netherlands, 2004.

M. Dumas, G. Governatory, A. H. M. ter Hofstede, and P. Oaks. A formal approach to negotiating agent development. *Electronic commerce research and applications*, 1(2), 2002.

P. M. Dung. On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–358, 1995.

P. E. Dunne. Prevarication in dispute protocols. In G. Sartor, editor, *Proceedings of the 9th Internatonal Conference on AI and Law (ICAIL)*, New York, NY, USA, 2003. ACM Press.

M. Elvang-Gøransson, P. Krause, and J. Fox. Acceptability of arguments as 'logical uncertainty'. In M. Clarke, R. Kruse, and S. Moral, editors, *Proceedings of the European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU 1993)*, number 747 in Lecture Notes in Computer Science, pages 85–90. Springer Verlag, Berlin, Germany, 1993a.

M. Elvang-Gøransson, P. Krause, and J. Fox. Dialectic reasoning with inconsistent information. In D. Heckerman and A. Mamdani, editors, *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, pages 114–121, Washington D. C., USA, 1993b. Morgan Kaufmann.

J. Eriksson and S. Janson. The trading agent competition – TAC 2002. *ERCIM News*, (51):57–58, October 2002.

K. Erol, J. Hendler, and D. Nau. Semantics for hierarchical task network planning. Technical Report CS-TR-3239, UMIACS-TR-94-31, Department of Computer Science, University of Maryland, 1994.

M. Esteva, J. A. Rodríguez, C. Sierra, P. Garcia, and J. L. Arcos. On the formal specifications of electronic institutions. In F. Dignum and C. Sierra, editors, *Agent Mediated Electronic Commerce (The European AgentLink Perspective)*, volume 1991 of *Lecture Notes in Artificial Intelligence*, pages 126–147. Springer Verlag, Berlin, Germany, 2001.

P. Faratin. *Automated Service Negotiation Between Autonomous Computational Agents*. PhD thesis, University of London, Queen Mary and Westfield College, Department of Electronic Engineering, 2000.

P. Faratin, C. Sierra, and N. R. Jennings. Negotiation decision functions for autonomous agents. *International Journal of Robotics and Autonomous Systems*, 24 (3–4):159–182, 1998.

P. Faratin, C. Sierra, and N. R. Jennings. Using similarity criteria to make tradeoffs in automated negotiations. *Artificial Intelligence*, 142(2):205–237, 2002.

S. Fatima, M. J. Wooldridge, and N. R. Jennings. Optimal negotiation strategies for agents with incomplete information. In J.-J. Meyer and M. Tambe, editors, *Intelligent Agent series VIII: Proceedings of the 8th International Workshop on Agent Theories, Architectures, and Languages (ATAL 2001)*, volume 2333 of *Lecture Notes in Computer Science*, pages 53–68. Springer Verlag, Berlin, Germany, 2001.

S. Fatima, M. J. Wooldridge, and N. R. Jennings. Multi-issue negotiation under time constraints. In C. Castelfranchi and L. Johnson, editors, *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2002)*, pages 143–150, New York, USA, 2002. ACM Press.

S. Fatima, M. J. Wooldridge, and N. R. Jennings. An agenda based framework for multi-issues negotiation. *Artificial Intelligence Journal*, 152(1):1–45, 2004a.

S. Fatima, M. J. Wooldridge, and N. R. Jennings. Bargaining with incomplete information. *Annals of Mathematics and Artificial Intelligence*, 2004b. (to appear).

FIPA. Communicative Act Library Specification. Technical Report XC00037H, Foundation for Intelligent Physical Agents, 10 August 2001.

R. Fisher, W. Ury, and B. Patton. *Getting to Yes: Negotiating Agreement Without Giving In.* Penguin Books, New York, USA, second edition, 1991.

J. Fox and S. Parsons. Arguing about beliefs and actions. In A. Hunter and S. Parsons, editors, *Applications of Uncertainty Formalisms*, number 1455 in Lecture Notes in Computer Science, pages 266–302. Springer Verlag, Berlin, Germany, 1998.

J. Fox, P. Krause, , and S. Ambler. Arguments, contradictions and practical reasoning. In B. Neumann, editor, *Proceedings of the 10th European Conference on Artificial Intelligence (ECAI-1992)*, pages 623–627, Vienna, Austria, 1992.

C. Fritschi and K. Dorer. Agent-oriented software engineering for successful tac participation. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 45–46. ACM Press, 2002. ISBN 1-58113-480-0. doi: http://doi.acm.org/10.1145/544741.544753.

T. Fung and R. Kowalski. The iff proof procedure for abductive logic programming. *Journal of Logic Programming*, 33(1):151–165, 1997.

Y. Gal and A. Pfeffer. A language for opponent modeling in repeated games. In S. Parsons and P. Gmytrasiewicz, editors, *Proceedings of the 5th Workshop on Game Theoretic and Decision Theoretic Agents (GTDT-2003)*, 2003.

M. Georgeff. Planning. *Annual Review of Computer Science*, 2:359–400, 1987.

F. Giunchiglia and L. Serafini. Multilanguage hierarchical logics or: How we can do without modal logics. *Artificial Intelligence*, 65(1):29–70, 1994.

T. F. Gordon. The pleadings game: formalizing procedural justice. In *Proceedings of the 4th international conference on artificial intelligence and law*, pages 10–19. ACM Press, 1993.

T. F. Gordon and N. Karacapilidis. The Zeno argumentation framework. In *Proceedings of the Sixth International Conference on AI and Law*, pages 10–18, New York, NY, USA, 1997. ACM Press.

A. Greenwald. The 2002 trading agent competition: An overview of agent strategies. *AI Magazine*, 24(1):83–91, 2003.

W. G. Griswold, R. Boyer, S. W. Brown, T. M. Truong, E. Bhasker, G. R. Jay, and R. B. Shapiro. Using mobile technology to create opportunitistic interactions on a university campus. In *UbiComp 2002 Workshop on Supporting Spontaneous Interaction in Ubiquitous Computing Settings*, September 2002.

B. J. Grosz and S. Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 82(2):269–357, 1996.

C. L. Hamblin. *Fallacies*. Methuen, London, UK, 1970.

D. Harel and A. Naamad. The STATEMATE semantics of statecharts. *ACM transactions on software engineering and methodology*, 5(4):293–333, October 1996.

J. C. Harsanyi. Approaches to the bargaining problem before and after the theory of games: a critical discussion of Zeuthen's, Hicks', and Nash's theories. *Econometrica*, 24:144–157, 1956.

M. He and N. R. Jennings. SouthamptonTAC: An adaptive autonomous trading agent. *ACM Trans on Internet Technology*, 3(3):218–235, 2003.

M. He, H. Leung, and N. R. Jennings. A fuzzy logic based bidding strategy for autonomous agents in continuous double auctions. *IEEE Transaction on Knowledge and Data Engineering*, 15(6):1345–1363, 2003.

K. V. Hindriks, F. S. de Boer, W. van der Hoek, and J.-J. Meyer. Agent programming in 3apl. *Autonomous Agents and Multi-Agent Systems*, 2(4):357–401, 1999.

D. Hitchcock. Aristotle's theory of argument evaluation. In K. Boudouris and J. Poulakos, editors, *The Philosophy of Communication*, volume 1. Ionia Press, Athens, Greece, 2002.

E. J. Horvitz, J. S. Breese, and M. Henrion. Decision theory in expert systems and artificial intelligence. *International Journal of Approximate Reasoning*, 2: 247–302, 1988.

J. Hulstijn. *Dialogue models for enquiry and transaction*. PhD thesis, Universiteit Twente, Enschede, The Netherlands, 2000.

J. Hulstijn and L. van der Torre. Combining goal generation and planning in an argumentation framework. In A. Hunter and J. Lang, editors, *Proceedings of the Workshop on Argument, Dialogue and Decision at the International Workshop on Non-monotonic Reasoning (NMR 2004)*, Whistler, Canada, June 2004.

E. A. Isaacs, J. C. Tang, and T. Morris. Piazza: a desktop environment supporting impromptu and planned interactions. In *Proceedings of the 1996 ACM conference on Computer supported cooperative work*, pages 315–324. ACM Press, 1996. ISBN 0-89791-765-0. doi: http://doi.acm.org/10.1145/240080.240316.

Java 2 Micro Edition. http://java.sun.com/j2me/.

N. R. Jennings. Controlling cooperative problem solving in industrial multi-agent systems using joint intention. *Artificial Intelligence*, 75(2):195–240, 1995.

N. R. Jennings. An agent-based approach for building complex software systems. *Communications of the ACM*, 44(4):35–41, 2001.

N. R. Jennings and M. J. Wooldridge. Applications of intelligent agents. In N. R. Jennings and M. J. Wooldridge, editors, *Agent Technology: Foundations, Applications, and Markets*, pages 3–28. Springer-Verlag, Heidelberg, Germany, 1998.

N. R. Jennings, S. Parsons, P. Noriega, and C. Sierra. On argumentation-based negotiation. In *Proceedings of the International Workshop on Multi-Agent Systems*, pages 1–7, Boston, USA, 1998a.

N. R. Jennings, K. Sycara, and M. J. Wooldridge. A roadmap of agent research and development. *Journal of Autonomous Agents and Multi-Agent Systems*, 1(1):7–38, 1998b. URL `citeseer.ist.psu.edu/jennings98roadmap.html`.

N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra, and M. J. Wooldridge. Automated negotiation: prospects, methods and challenges. *International Journal of Group Decision and Negotiation*, 10(2):199–215, 2001.

R. Johnson. *Manifest rationality: a pragmatic theory of argument*. Lawrence Erlbaum Associates, Mahwah NJ, USA, 2000.

T. Juan, A. Pearce, and L. Sterling. ROADMAP: Extending the gaia methodology for complex open systems. In C. Castelfranchi and W. L. Johnson, editors, *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2002), Bologna, Italy*, pages 3–10, New York City NY, USA, 2002. ACM Press.

H. Jung, M. Tambe, and S. Kulkarni. Argumentation as distributed constraint satisfaction: applications and results. In J. P. Müller, E. Andre, S. Sen, and C. Frasson, editors, *Proceedings of the Fifth International Conference on Autonomous Agents*, pages 324–331, Montreal, Canada, 2001. ACM Press.

A. Kakas and P. Moraitis. Argumentation based decision making for autonomous agents. In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2003)*, pages 883–890, Melbourne, Australia, 2003.

M. Kakihara and C. Sørensen. Mobility: An extended perspective. In R. S. Jr, editor, *Proceedings of the 35th Hawaii International Conference on Systems Sciences*, Big Island, Hawaii, USA, 2002. IEEE Press.

S. Kambhampati. A comparative analysis of partial order planning and task reduction planning. *SIGART Bulletin*, 6(1):16–25, 1995.

M. Karlins and H. I. Abelson. *Persuasion: How Opinions and Attitudes are Changed*. Lockwood, second edition, 1970.

N. C. Karunatillake and N. R. Jennings. Is it worth arguing? In *Proceedings of the 1st International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2004)*, Lecture Notes in Computer Science. Springer Verlag, Heidelberg, Germany, 2004. (to appear).

F. L. Koch and I. Rahwan. The role of agents in intelligent mobile services. In *Proceedings of the 7th Pacific Rim International Workshop on Multi-Agents (PRIMA)*, Auckland, New Zealand, 2004a.

F. L. Koch and I. Rahwan. Classification of agents-based mobile assistants. In *Proceedings of the AAMAS Workshop on Agents for Ubiquitous Computing (Ubi-Agents)*, New York NY, USA, 2004b.

R. Kowalczyk. On negotiation as a distributed fuzzy constraint satisfaction problem. In *Proceedings of the 3rd International Symposium on Soft Computing for Industry*, World Automation Congress, pages 631–637, 2000.

R. Kowalczyk and V. Bui. On constraint-based reasoning in e-negotiation agents. In F. Dighum and U. Cortés, editors, *Agent-Mediated Electronic Commerce III*, volume 2003 of *Lecture Notes in Computer Science*, pages 31–46. Springer Verlag, Berlin, Germany, 2001.

R. Kowalski and F. Sadri. From logic programming to multi-agent systems. *Annals of Mathematics and Artificial Intelligence*, 25:391–419, 1999.

S. Kraus. *Strategic Negotiation in Multi-Agent Environments*. MIT Press, Cambridge MA, USA, 2001.

S. Kraus, K. Sycara, and A. Evenchik. Reaching agreements through argumentation: A logical model and implementation. *Artificial Intelligence*, 104(1–2):1–69, 1998.

P. Krause, S. Ambler, M. Elvang-Gøransson, and J. Fox. A logic of argumentation for reasoning under uncertainty. *Computational Intelligence*, 11:113–131, 1995.

S. Kumar, M. J. Huber, and P. R. Cohen. Representing and executing protocols as joint actions. In C. Castelfranchi and W. L. Johnson, editors, *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2002), Bologna, Italy*, pages 543–550, New York City NY, USA, 2002. ACM Press.

Y. Labrou, T. Finin, and Y. Peng. Agent communication languages: The current landscape. *Intelligent Systems*, 14(2):45–52, 1999.

K. Larson and T. Sandholm. An alternating offers bargaining model for computationally limited agents. In C. Castelfranchi and L. Johnson, editors, *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-2002)*, pages 135–142, 2002.

R. J. Lewicki, D. M. Saunders, and J. W. Minton, editors. *Essentials of Negotiation*. McGraw-Hill, Boston, MA, 2003.

G. L. Lilien, P. Kotler, and S. K. Moorthy. *Marketing Models*. Prentice-Hall Press, USA, 1992.

P. Luff and C. Heath. Mobility in collaboration. In *Proceedings of the 1998 ACM conference on Computer Supported Cooperative Work*, pages 305–314. ACM Press, 1998. ISBN 1-58113-009-0. doi: http://doi.acm.org/10.1145/289444.289505.

J. Lukacs. *Five Days in London: May 1940.* Yale University Press, New Haven, CT, USA, 1999.

X. Luo, N. R. Jennings, and N. Shadbolt. Acquiring tradeoff preferences for automated negotiations: A case study. In *Proceedings of the 5th International Workshop on Agent-Mediated E-Commerce*, pages 41–48, Melbourne, Australia, 2003a.

X. Luo, N. R. Jennings, N. Shadbolt, H.-F. Leung, and J. H.-M. Lee. A fuzzy constraint-based knowledge model for bilateral, multi-issue negotiations in competitive environments. *Artificial Intelligence*, 148(1–2):53–102, 2003b.

S. Marsh. *Formalising Trust as a Computational Concept.* PhD thesis, Department of Mathematics and Computer Science, University of Stirling, 1994.

A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic Theory.* Oxford University Press, New York NY, USA, 1995.

N. Maudet and B. Chaib-draa. Commitment-based and dialogue-game based protocols–news trends in agent communication language. *Knowledge Engineering Review*, 17(2):157–179, 2003.

J. Mayfield, Y. Labrou, and T. Finin. Evaluating kqml as an agent communication language. In M. J. Wooldridge, J. P. Müller, and M. Tambe, editors, *Intelligent Agents II*, volume 1039 of *Lecture Notes in Computer Science*, pages 347–360. Springer Verlag, Berlin, Germany, 1996.

P. McBurney. *Rational Interaction.* PhD thesis, University of Liverpool, 2002.

P. McBurney and S. Parsons. Games that agents play: A formal framework for dialogues between autonomous agents. *Journal of Logic, Language and Information*, 11(3):315–334, 2002.

P. McBurney and S. Parsons. Dialogue game protocols. In M.-P. Huget, editor, *Communication in Multiagent Systems*, volume 2650 of *Lecture Notes in Computer Science*, pages 269–283. Springer Verlag, Berlin, Germany, 2003.

P. McBurney, R. M. van Eijk, S. Parsons, and L. Amgoud. A dialogue-game protocol for agent purchase negotiations. *Journal of Autonomous Agents and Multi-Agent Systems*, 7(3):235–273, 2003.

K. R. McKeown. *Text generation: using discourse strategies and focus constraints to generate natural language text*. Cambridge University Press, 1985. ISBN 0-521-30116-5.

D. J. O'Keefe. *Persuasion: Theory and Research.* SAGE Publications, 1990.

M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, Cambridge MA, USA, 1994.

L. Padgham and M. Winikoff. *Developing Intelligent Agent Systems: A Practical Guide*. John Wiley and Sons, 2004.

V. Pareto. *Manual of Political Economy*. Augustus M. Kelley Publishers, New York NY, USA, 1906. Reprint edition (June 1, 1969).

S. Parsons and N. R. Jennings. Negotiation through argumentation—a preliminary report. In *Proceedings of the 2nd International Conference on Multi Agent Systems*, pages 267–274, 1996.

S. Parsons, C. Sierra, and N. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.

S. Parsons, M. J. Wooldridge, and L. Amgoud. An analysis of formal inter-agent dialogues. In C. Castelfranchi and L. Johnson, editors, *Proceedings of the 1st International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS-2002)*, pages 394–401, New York, USA, 2002. ACM Press.

S. Parsons, M. J. Wooldridge, and L. Amgoud. Properties and complexity of formal inter-agent dialogues. *Journal of Logic and Computation*, 13(3):347–376, 2003.

H. V. D. P. Parunak. Industrial and practical applications of dai. In G. Weiss, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 377–421. MIT Press, Cambridge MA, USA, 1999.

M. Perry, K. O'Hara, A. Sellen, B. Brown, and R. Harper. Dealing with mobility: understanding access anytime, anywhere. *ACM Transactions on Computer-Human Interaction*, 8(4):323–347, 2001.

J. L. Pollock. The logical foundations of goal-regression planning in autonomous agents. *Artificial Intelligence*, 106(2):267–334, 1998.

W. Poundstone. *Prisoner's Dilemma*. Anchor, New York, USA, 1993.

H. Prakken and G. Sartor. The role of logic in computational models of legal argument: a criticial survey. In A. Kakas and F. Sadri, editors, *Computational Logic: From Logic Programming into the Future (In honour of Bob Kowalski)*, volume 2048 of *Lecture Notes in Computer Science*, pages 342–3. Springer Verlag, Berlin, Germany, 2001.

D. G. Pruitt. *Negotiation Behavior*. Academic Press, New York, USA, 1981.

I. Rahwan, R. Kowalczyk, and H. H. Pham. Intelligent agents for automated one-to-many e-commerce negotiation. In M. Oudshoorn, editor, *Proceedings of the 25th Australasian conference on Computer science*, pages 197–204. Australian Computer Society Press, 2002.

I. Rahwan, P. McBurney, and L. Sonenberg. Towards a theory of negotiation strategy (a preliminary report). In S. Parsons and P. Gmytrasiewicz, editors, *Proceedings of the 5th Workshop on Game Theoretic and Decision Theoretic Agents (GTDT-2003)*, pages 73–80, 2003a.

I. Rahwan, S. D. Ramchurn, N. R. Jennings, P. McBurney, S. Parsons, and L. Sonenberg. Argumentation based negotiation. *Knowledge Engineering Review (to appear)*, 18(4):343–375, 2003b.

I. Rahwan, L. Sonenberg, and F. Dignum. Towards interest-based negotiation. In J. Rosenschein, T. Sandholm, M. J. Wooldridge, and M. Yokoo, editors, *Pro-*

ceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2003), pages 773–780. ACM Press, 2003c.

I. Rahwan, C. Graham, and L. Sonenberg. Supporting impromptu coordination using automated negotiation. In *Proceedings of the Pacific Rim International Workshop on MultiAgent Systems (PRIMA)*, Auckland, New Zealand, 2004a.

I. Rahwan, P. McBurney, and L. Sonenberg. Bargaining and argument-based negotiation: Some preliminary comparisons. In P. M. I. Rahwan and C. Reed, editors, *Proceedings of the AAMAS First International Workshop on Argumentation in Multi-Agent Systems (ArgMAS)*, New York NY, USA, 2004b.

I. Rahwan, L. Sonenberg, and F. Dignum. On interest-based negotiation. In F. Dignum, editor, *Advances in Agent Communication*, volume 2922 of *Lecture Notes in Artificial Intelligence*, pages 383–401. Springer Verlag, Berlin, Germany, 2004c.

T. Rahwan, T. Rahwan, I. Rahwan, and R. Ashri. Agent-based support for mobile users using AgentSpeak(L). In P. Giorgini, B. Hederson-Sellers, and M. Winikoff, editors, *Agent Oriented Information Systems*, volume 3030 of *Lecture Notes in Artificial Intelligence*, pages 47–62. Springer Verlag, Berlin, Germany, 2004d.

S. D. Ramchurn, N. R. Jennings, and C. Sierra. Persuasive negotiation for autonomous agents: a rhetorical approach. In C. Reed, F. Grasso, and G. Carenini, editors, *Proceedings of the IJCAI Workshop on Computational Models of Natural Argument*, pages 9–17. AAAI Press, 2003.

A. S. Rao and M. P. Georgeff. BDI-agents: from theory to practice. In *Proceedings of the First International Conference on Multiagent Systems*, San Francisco, USA, 1995.

C. Reed. Dialogue frames in agent communication. In E. Durfee, editor, *Proceedings of the 4th International Conference on Multi-Agent Systems (ICMAS 1998)*, pages 246–253, Boston MA, USA, 1998. IEEE Press.

J. H. Roberts and J. M. Lattin. Development and testing of a model of consideration set composition. *Journal of Marketing Research*, 28(4):429–440, 1991.

J. Rosenschein and G. Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. MIT Press, Cambridge MA, USA, 1994.

J. C. Rubiera, J. M. M. Lopez, and J. D. Muro. A fuzzy model of reputation in multi-agent systems. In *Proceedings of the 5th International Conference on Autonomous Agents*, pages 25–26. ACM Press, 2001.

A. Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica*, 50(1): 97–109, 1982.

A. Rubinstein. *Modeling Bounded Rationality*. MIT Press, Cambridge MA, USA, 1997.

J. Sabater and C. Sierra. Regret: a reputation model for gregarious societies. In C. Castelfranchi and L. Johnson, editors, *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-2002)*, pages 475–482. ACM Press, 2002.

N. M. Sadeh. *M–Commerce Technologies, Service, and Business Models*. Wiley, Hoboken NJ, USA, 2003.

F. Sadri and F. Toni. Abduction with negation as failure for active and reactive rules. In E. Lamma and P. Mello, editors, *Advances in Artificial Intelligence, 6th Congress of the Italian Association for Artificial Intelligence (AI*IA 1999)*, volume 1792 of *Lecture Notes in Computer Science*, pages 49–60. Springer Verlag, Berlin, Germany, 2000.

F. Sadri, F. Toni, and P. Torroni. Dialogues for negotiation: agent varieties and dialogue sequences. In J.-J. Meyer and M. Tambe, editors, *Intelligent Agent Series VIII: Proceedings of the 8th International Workshop on Agent Theories, Architectures, and Languages (ATAL 2001)*, volume 2333 of *Lecture Notes in Computer Science*, pages 69–84. Springer Verlag, Berlin, Germany, 2001a.

F. Sadri, F. Toni, and P. Torroni. Logic agents, dialogues and negotiation: an abductive approach. In K. Stathis and M. Schroeder, editors, *Proceedings of the AISB 2001 Symposium on Information Agents for E-Commerce*, 2001b.

F. Sadri, F. Toni, and P. Torroni. Abductive logic programming architecture for negotiating agents. In *Proceedings of the 8th European Conference on Logics in Artificial Intelligence (JELIA-2002)*, volume 2424 of *Lecture Notes in Computer Science*, pages 419–431. Springer Verlag, Berlin, Germany, 2002.

L. Samuelson. *Evolutionary Games and Equilibrium Selection.* MIT Press, Cambridge MA, USA, 1998.

T. Sandholm. eMediator: A next generation electronic commerce server. *Computational Intelligence, Special issue on Agent Technology for Electronic Commerce*, 18(4):656–676, 2002a.

T. Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135(1–2):1–54, 2002b.

T. Sandholm and V. Lesser. Leveled commitment contracting: A backtracking instrument for multiagent systems. *Artificial Intelligence Magazine*, 23(3):89–100, 2002.

M. Schut and M. Wooldridge. The control of reasoning in resource-bounded agents. *Knowledge Engineering Review*, 16(3):215–240, 2001.

J. Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, New York, USA, 1969.

C. Sierra, P. Faratin, and N. R. Jennings. A service-oriented negotiation model between autonomous agents. In *Proceedings of the 8th European Workshop on Modeling Autonomous Agents in a Multi-Agent World (MAAMAW-1997)*, pages 17–35, Ronneby, Sweden, 1997.

C. Sierra, N. R. Jennings, P. Noriega, and S. Parsons. A framework for argumentation-based negotiation. In M. Singh, A. Rao, and M. J. Wooldridge, editors, *Intelligent Agent IV: 4th International Workshop on Agent Theories, Architectures and Languages (ATAL-1997)*, volume 1365 of *Lecture Notes in Artificial Intelligence*, pages 177–192. Springer Verlag, Berlin, Germany, 1998.

C. Slade. Reasons to buy: The logic of advertisements. *Argumentation*, 16:157–178, 2002.

A. M. Spence. *Market Signaling: Informational Transfer in Hiring and Related Screening Processes*. Harvard Economic Studies. Harvard University Press, Cambridge MA, USA, 1974.

K. Sycara. Arguments of persuasion in labour mediation. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pages 294–296, 1985.

K. Sycara. Persuasive argumentation in negotiation. *Theory and Decision*, 18(3): 203–242, 1990.

K. Sycara. The PERSUADER. In D. Shapiro, editor, *The Encyclopedia of Artificial Intelligence*. John Wiley & Sons, January 1992.

TAC. *The Trading Agent Competition*. World Wide Web, http://www.sics.se/tac/, 2003.

R. H. Thomason. Desires and defaults: A framework for planning with inferred goals. In A. G. Cohn, F. Giunchiglia, and B. Selman, editors, *KR 2000: Principles of Knowledge Representation and Reasoning, Proceedings of the Seventh International Conference, Breckenridge, Colorado, USA*, pages 702–713. Morgan Kaufmann Publishers, 2000.

P. Torroni. A study on the termination of negotiation dialogues. In C. Castelfranchi and L. Johnson, editors, *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2002)*, pages 1223–1230, New York, USA, 2002. ACM Press.

P. Torroni and F. Toni. Extending a logic based one-to-one negotiation framework to one-to-many negotiation. In P. P. A. Omicini and R. Tolksdorf, editors, *2nd International Workshop on Engineering Societies in the Agents World*

II (ESAW 2001), volume 2203 of *Lecture Notes in Computer Science*, pages 105–118. Springer Verlag, Berlin, Germany, 2001.

D. Traum. Speech acts for dialogue agents. In M. J. Wooldridge and A. Rao, editors, *Foundations And Theories Of Rational Agents*, pages 169–201. Kluwer Academic Publishers, Dordrecht, Netherlands, 1999.

H. R. Varian. Mechanism design for computerized agents. In *Proceedings of the USENIX Workshop on Electronic Commerce*, New York, July 1995.

J. von Neuman and O. Morgenstern. *The Theory of Games and Economic Behaviour*. Princeton University Press, Princeton NJ, USA, 1944.

B. von Stengel. Computing equilibria for two-person games. In R. Aumann and S. Hart, editors, *Handbook of game theory*, volume 3, chapter 45, pages 1723–1759. Elsevier Science Publishers (North-Holland), Amsterdam, Netherlands, 2002.

D. N. Walton and E. C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. SUNY Press, Albany NY, USA, 1995.

S. Weiss. *Handheld Usability*. Wiley, Hoboken NJ, USA, 2002.

M. Wooldridge, N. R. Jennings, and D. Kinny. The gaia methodology for agent-oriented analysis and design. *Autonomous Agents and Multi-Agent Systems*, 3(3): 285–312, 2000.

M. J. Wooldridge. *Reasoning about Rational Agents*. MIT Press, 2000.

M. J. Wooldridge. *An Introduction to MultiAgent Systems*. John Wiley & Sons, Chichester, England, 2002.

M. J. Wooldridge and S. Parsons. Languages for negotiation. In W. Horn, editor, *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI-2000)*, pages 393–397. John Wiley & Sons, 2000.

P. R. Wurman. *Market Structure and Multidimensional Auction Design for Computational Economies*. PhD thesis, University of Michigan, August 1999.

M. Yokoo. The distributed constraint satisfaction problem: Formalization and algorithms. *IEEE Transaction On Knowledge and Data engineering*, 10(5):673–685, September 1998.

M. A. Young. *Rational Games: A Philosophy of Business Negotiation from Practical Reason*. Quorum Books, Westport, CT, USA, 2001.

B. Yu and M. P. Singh. Distributed reputation management for electronic commerce. *Computational Intelligence*, 18(4):535–549, 2002.

L. A. Zadeh. Similarity relations and fuzzy orderings. *Information Sciences*, 3: 177–200, 1971.

F. Zambonelli and H. V. D. Parunak. Towards a paradigm change in computer science and software engineering: a synthesis. *The Knowledge Engineering Review*, 2004. (to appear).

F. Zambonelli, N. R. Jennings, and M. Wooldridge. Developing multiagent systems: the gaia methodology. *ACM Trans on Software Engineering and Methodology*, 12(3):317–0370, 2003.

F. L. B. Zeuthen. *Problems of Monopoly and Economic Warfare*. Routledge and Sons, London, UK, 1930.

I. Zukerman, R. McConachy, and K. B. Korb. Bayesian reasoning in an abductive mechanism for argument generation and analysis. In *Proceedings of the 15th National Conference on Artificial Intelligence*, pages 833–838. AAAI Press, 1998.