UNIVERSITY OF SOUTHAMPTON

Motivation and Autonomy for Pre-Negotiation

by

Steve Munroe

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Faculty of Engineering, Science and Mathematics
School of Electronics and Computer Science

December 2005

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING, SCIENCE AND MATHEMATICS
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

by Steve Munroe

The issues and challenges that surround the concept of interaction are a key focus in agent research. Of all forms of interaction, negotiation is perhaps the most challenging and important for designers of agent-based systems. This is because agents are localised entities that try to obtain satisfaction for their own activities with only limited understanding of the activities of other agents and the goals of the system as a whole. As a result, agents often have conflicting objectives, requiring methods and techniques, such as negotiation, for the resolution of such conflicts. Bargaining, in particular, provides a way for agents to attempt to find agreements in situations of conflict where no external authority can intervene. The work in this thesis describes models and mechanisms that enable agents to use bargaining as a tool to further their aims while ensuring that any agreements reached are consistent with existing goals. We focus on pre-negotiation, that point in time before negotiation begins where decisions that affect the way negotiation proceeds are taken. In the thesis, we bring together deliberative architectures, models of motivation and negotiation to address the issues involved in pre-negotiation.

Specifically, this thesis makes three main contributions. First, it provides a model of negotiation goals that incorporates an analysis of negotiation issues, a deliberative preference determination mechanism and a novel use of motivational mechanisms within negotiation. Second, it provides an analysis and taxonomy of bilateral negotiation issues. Finally, it provides a suite of mechanisms to enable an agent to modify its approach to negotiation based upon information it obtains about the negotiation context. Combined, these contributions enable agents to be more effective negotiators in dynamic domains where user guidance is problematic.

Contents

5 Bilateral Issue Combinations 93

6 Environmental Constraints on Pre-Negotiation 115

List of Figures

List of Tables

Acknowledgements

In undertaking the work in this thesis I leant upon and required the support of many people. Chief among these was my Supervisor Professor Michael Luck who, despite my best efforts, remained sane and supportive throughout. Without his help, this thesis would not exist, and I cannot overstate my gratitude for his encouragement, advice, guidance, 'red pens' and, most of all, his unstinting belief in my ability to "get it done". I would also like to thank Mark d'Inverno for his invaluable contributions to the formal specifications in this work. Many thanks go to Nick Jennings for running and overseeing what I believe to be the best of all intellectual environments within which to come and work every day. I give thanks to all the anonymous reviewers of my work, for their insightful comments, and to the people of the IAM Group who, simply by being around, provide insight and inspiration. I also thank the EPSRC for the funding to do this work.

To all my colleagues, friends and peers, I thank you all. In particular, Ronald Ashri, Gopal Ramchurn, Raj Dash, Viet Dung Dang, Fabiola Lopez y Lopez, Cora Excelente Toledo, Terry Payne, Becky Earl, Luc Moreau, MingHua He and Xoudong Luo, Luke Teacy, Jigar Patel, and Maira Rodrigues and everyone in Bay 3. All of you helped get me through. Many friends bore the brunt of my stresses and strains through these last four years, I apologise and thank you all for remaining true. In particular, I would like to give big thanks to Serena, Kristin, Andy, Maria and Robert, Claudia and Nick, Adrian and Rodrigo, Symos, Nota, NY Emma, Chris, Danius, Dave, Naill, Katia, Martin and Laura, Mischa, Paul and Seb, Peter, Roxana, the Bencraft posse, and Sue, Martin and Tim.

Finally, and without whom I would be lost, my family, I love you all, and I couldn't have done it without you. To my sister, Deborah, and her family; Mark, Inti, Kanu, Isla and Belynne, I owe you all Big Time, and love you all to pieces. To Ray, whose been a great support and an even greater friend. To Mai and my extended family in Thailand, I send you all my love.

Finally, I would like to thank my parents. You both mean the world to me and if I've done this thesis for anyone, I have done it for the two of you.

To my Grandparents, we miss you all so much and will always love you.

Chapter 1

Introduction

1.1 Introduction

The increasing pace of modern life, for both individuals and business, engenders a need to automate many of the tasks and activities that require our attention. For many years, software engineers have responded by designing and building systems that increasingly support, manage and, in some cases, completely take charge of many activities that previously required human intervention. More recently, the concept of a computational or intelligent agent has come to the forefront of such thinking. Agents are software programs able to act and make decisions on behalf of, or in place of, humans, and today intelligent agents are active in many areas. For example, there are agents that gather information on the World Wide Web [154], guide human decision-making [6], simulate behaviour [23], manage workflow [58], and provide assistance in scheduling and planning [71]. In fact, the use of agent technology is becoming increasingly widespread and can be found in fields as disparate as business [61, 62, 73], education [139], healthcare [44], defence [123] and many more besides.

Agents do not just act in isolation; their greatest power comes from their ability to interact with each other in large distributed groups, making them extremely useful for managing open, heterogeneous and distributed systems (see [62, 63, 68, 111] for some examples). Such groups of agents are known as *multi-agent systems* (see [57] for an introduction), and the members of such systems are able to communicate information to each other regarding tasks that must be performed or problems that must be solved. Agents that are unable to solve a problem or perform a task alone can communicate with other agents in order to obtain information that will enable them to solve the problem, or to transfer the task to other agents better equipped to deal with it. Such an ability to flexibly *distribute information* and *reallocate tasks* allows these systems to exploit both individual agent characteristics, such as their action capabilities, reasoning skills, and local information, etc, and also take advantage of the possibilities inherent in the powerful social metaphors of communication, cooperation, coordination.

Systems such as these are challenging to design and construct. In particular, managing the often competing aims and desires of the members of such systems is a difficult and demanding undertaking. For this reason, much research has examined ways to manage and resolve the conflicts that can arise when individuals with conflicting aims must interact to solve problems and take action. In developing such conflict resolution techniques, the agent community has made use of many approaches and tools developed to resolve conflict within the human sphere of activity.

1.2 Conflict and its Resolution

Conflict is, unfortunately, an all too common experience, and can arise when people with differing views and desires must interact to jointly determine (or avert) a specific outcome. It can occur in the most trivial of cases, such as between a mother and her son arguing over the distribution of weekly chores, and more serious cases, such as between two sovereign states going to war over disputed territory. In all cases, however, it is generally desirable to resolve conflict in an efficient and fair manner. Since conflict is such an integral part of life, many approaches have been developed to either avoid it completely or to resolve it once it arises.

In terms of avoiding conflict, the institution of laws, the development of norms and the design of coordination mechanisms are just some of the methods that can be applied. Such approaches can only go so far, however, since it is often not possible to predict and therefore legislate against, or circumvent, every possible source of conflict. This is especially true in groups composed of individuals with a wide spread of views, aims and needs.

Though there are many methods that can be applied to avoid such conflicts, there are still many cases where conflict cannot be avoided, in which case techniques such as voting [30] or third party intervention [18] can be used to resolve the conflict. Arguably though, the most important technique used for conflict resolution is *negotiation* and, due to its importance and ubiquity in agent research, we focus our efforts in this thesis here.

1.2.1 Negotiation

There are many definitions of negotiation available in the literature, but for our purposes the general notion presented by Lomuscio *et al.* [76] is sufficient:

> *Negotiation is a process by which a group of agents communicate with one another to try to come to a mutually acceptable agreement on some matter.*

The definition is vague about the nature of the communication that occurs, since this can be structured in a number of different ways. For example, negotiators can simply state the outcomes they prefer, or they can also offer critiques of the other's statements, or even use persuasion techniques. Negotiation is, therefore, a term that covers a broad range of interactions. Common to them all, however, is the fact that they are structured interactions that have the aim of resolving some conflict. One very widely used form of negotiation is *bargaining*, in which two or more individuals put forward *offers* and *counter-offers* about the outcomes they prefer. Bargaining occurs in many situations, from the market-stall holder and a customer haggling over the price of some item on sale, to the leaders of two countries discussing the terms of a treaty. Though there can be cultural differences [52], the basic bargaining process can be seen across cultures, and most often arises in situations where individuals have a need or a desire to determine an outcome together.

In today's interdependent world, the need to work together is more apparent than ever. Countries, organisations, businesses and individuals increasingly rely and depend upon one another and, as such, effective ways of resolving conflict are essential tools for all levels of society.

1.2.2 Agent Negotiation

Agent-based negotiation research attempts to apply the lessons learned and the techniques developed from the wider field of negotiation within a computational context. It is a demanding project, since it is necessary to take what is an often difficult activity for humans and enable its use by much more cognitively limited computational entities. Despite this difficulty, research on agent negotiation techniques is maturing rapidly, and many forms of negotiation between intelligent agents have been investigated, from auctions [24], through bilateral negotiations (i.e. bargaining) [33], to argumentation-based negotiation [135].

While there have been many models and mechanisms developed that enable agents to reason and act effectively *within* negotiation (e.g. [34, 76, 86, 133, 135, 153]), there are far fewer available that focus on the problems that must be addressed in order that effective *preparation* for negotiation can be achieved. Despite the growing maturity of the research, this aspect of the negotiation process has, as yet, received little attention. A major contributing factor for this is that many existing models assume negotiation to be an isolated endeavor not affected by the broader scope of an agent's activity and concerns. However, in real-world, human negotiations, such a compartmentalisation of negotiation rarely occurs, and many external factors can influence the approach to, and the execution of, a negotiation encounter. For example, even in simple negotiations of the sort commonly seen in marketplaces where a person is haggling over the price of some item on sale, considerations over how long the haggling should continue for, whether the item should be gift-wrapped, or the desirability of a return policy should the item later be found defective, relate to other concerns of the buyer, that are external to the negotiation itself. For example, gift-wrapping may be important if the item is to be a present for another, a return policy is only useful if the buyer is likely to return to the location of the marketplace in the near future, and so on. While it is simple for a human negotiator to understand such issues, designing an autonomous agent that can reason effectively about such concerns in the context of its own activities and desires is a challenging problem.

Such issues surrounding the preparation for negotiation arise in what Saunders calls the *pre-negotiation* stage of negotiation [129]. Within this stage, a number of questions can be identified, such as: what issues should form the focus of negotiation; in what order should they be negotiated; who should be negotiated with; how to negotiate (i.e. what protocol to use); what issue settlements can be considered acceptable; and what constraints there are and what effects they have on the negotiation. Some of these areas have been examined already. For example, in [35], Fatima *et al.* explore how the negotiation agenda (or the order of issues in sequential negotiations) can be determined under time constraints, while others have examined how to identify negotiation opponents (e.g. [119, 31]). However, such aspects as identifying negotiation issues and autonomously determining settlements over those issues have had much less attention, and it is here that we focus our efforts. Figure 1.1 shows those elements that comprise the concerns of pre-negotiation; the ones dealt with in this thesis are represented as shaded boxes.

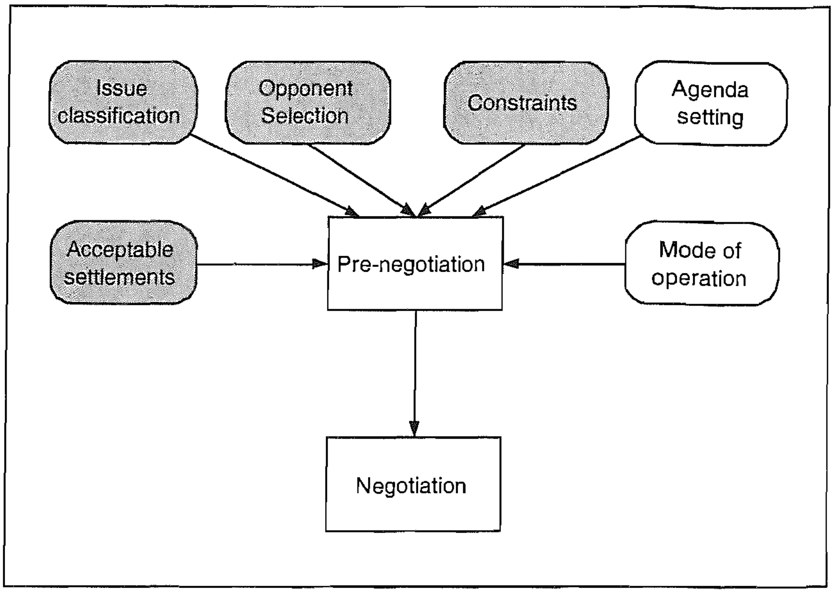


FIGURE 1.1: Pre-negotiation considerations

1.3 A Motivating Example

To motivate the discussion we provide an example that highlights the kinds of issues and concerns that might face an agent in its deliberations about a forthcoming negotiation.

Imagine two people (lets call them Alice and Bob) attempting to coordinate a joint travel itinerary. Both must reach agreement over a plan to travel together to a conference in New York. In order for their goal to attend the conference to be satisfied, both must engage in negotiation in order to explore their perhaps conflicting requirements in order to reach agreement. There are a number of potential sources of conflict. First, they must determine which airport to fly from, where the possibilities are all the London airports (i.e. Heathrow, Gatwick, Stansted and Luton). Next, the seat class (economy or business) must be agreed, since both Alice and Bob must sit together in order to discuss some conference related business. Finally, both Alice and Bob must agree on the date of travel.

Let us suppose that initially, Alice believes that she was to travel alone and makes the following deliberations. First, she decides that her preference is to fly from Gatwick, though she would also be willing to fly from Heathrow if necessary. She intends to save money by booking an economy class seat, and she would like to leave one day before the start of the conference. On realising that Bob is also going to the conference, Alice revises her preferences. Since she knows that Bob lives closer to Heathrow than Gatwick, Alice decides to drop the departure airport from the list of things she will discuss with Bob, and thus gives Bob the freedom to choose where they will fly from (Alice expects Bob will choose Heathrow). However, Alice is determined to save money on the flight, and decides to negotiate with Bob about the seat class, hoping to convince Bob to buy an economy ticket. Finally, Alice discovers that Bob is intending to travel to New York three days before the conference begins, but Alice cannot travel so early due to other work commitments, so decides to state that she can only travel one day before the conference begins and that this is non-negotiable. If Bob accepts, they will travel together; if not, Alice will have to travel alone and meet Bob at the conference.

This example presents a number of difficult problems that must be considered by anyone considering entering into a negotiation. What to negotiate about? What kinds of agreements are acceptable? How do existing commitments constrain what is considered as acceptable? Finally, how can expectations about the negotiation partner affect the ways the above questions are answered. It is the purpose of the rest of this thesis to address these questions.

1.4 Pre-Negotiation and Autonomy

The kinds of problems that must be addressed in the pre-negotiation stage have, to date, been circumvented by most agent research by assuming that either the user or designer solves these problems for the agent. Thus, the user or designer is assumed to determine the focus of negotiation and the desired outcomes that are then simply given to the agent, which then attempts to negotiate within these given constraints. However, in many situations we cannot expect these constraints to be defined in detail since they are often dependent upon the agent's situation. For example, in systems containing many hundreds of agents, each performing many different tasks, expecting a user to maintain an awareness of the agent's situation is unrealistic. In such cases, it is preferable for the agent itself to deal with the kinds of problems that currently require direct human input.

To enable such a capability, we must address the issue of agent *autonomy*. Autonomy refers to an agent being able to work away from human direction, and being the only arbiter over its own goals and actions. In this way, autonomy provides a way to enable agents to operate in domains where human oversight is difficult or undesirable to apply, and is thus seen as an approach to tackle such domains and is not an end in itself. In terms of negotiation, agent autonomy is evident in the process of making offers and counter-offers without the need for close human supervision, where the decisions regarding which offer to make next in a series of offers is left to the agent's discretion. However, in existing work, the use of autonomy has been limited to the negotiation process itself, since the negotiation objectives have already been (or are assumed to be) determined by a human. This both limits the range of contexts in which automated negotiation can be applied and constrains its value. It is far more appropriate for issues arising in pre-negotiation relating to an agent's current situation to be addressed in an autonomous fashion by the agent. In consequence, the role of autonomy and its place in negotiation is a key focus of our work.

Indeed, while autonomy is undeniably a critical issue in the field of intelligent agents [8, 17], many assume it as an emergent property rather than one that must be addressed at an architectural level. In the literature, there are two very different conceptions of autonomy. On the one hand, some researchers operationalise autonomy as the level or *degree* to which an agent can achieve its goals without assistance, and thus strongly relates the notion of autonomy to an agent's *dependence* upon others. High dependence under this definition equates to low autonomy. On the other hand, autonomy can be considered as an *absolute enabler* for generating goals in response to different situations. While the achievement of some of these goals may depend on other agents, this dependence does not affect the autonomous generation of goals.

In the context of pre-negotiation, autonomy relates to the ability to determine the focus of negotiation, or what is to be negotiated about. In this view, autonomous agents are able to generate their own negotiation preferences and to select between multiple alternative options. Thus, from a purely conceptual or theoretical point of view removed from practical considerations, autonomy can naturally be regarded as absolute, without dimension or measure of degree — one can either generate one's own goals or one cannot. For us, therefore, autonomy is not the same as independence — an agent does not simply lose its autonomy by virtue of depending on another for a particular goal. In this thesis we provide a practical implementation of this view of autonomy, in the context of negotiation, through the development of several computational mechanisms.

1.5 Aims and Principles

The agent community has been researching negotiation from many perspectives. For example, some work has investigated negotiation analytically using the techniques of game theory [68, 124], and many negotiation mechanisms have been developed that apply those techniques. In addition, multi-party negotiations, as found in auctions, have been examined in relation to multi-agent systems by many researchers (e.g. [24, 51, 87]), and there have also been efforts to examine ways to enable agents to learn within the context of negotiation (e.g. [20]).

By contrast, relatively little work has focused on the concerns that arise in the pre-negotiation stage of negotiation, though it is touched upon by some researchers. For example, Ramchurn *et al.* [120] describe agents that shape their negotiations by adding or removing issues as they learn more about their negotiation opponents, and Faratin *et al.* [34] describe how agents can manipulate the set of issues they negotiate over to respond to the dynamics of the negotiation episode. At a higher level of analysis Urbig *et al.* [146] give explicit consideration to the interactions that occur between the selection of issues, opponents and the steps made during negotiation.

None of these approaches, however, explicitly deals with the integration of the negotiation activity within the agent's broader problem-solving role. Specifically, negotiation is treated as an isolated endeavour, and the kinds of agreements that are considered acceptable are either given directly to the agent by a user, or identified by the maximisation of a utility function, which acts as a kind of implicit goal where higher utility means greater satisfaction. Neither do these approaches allow for consideration of the effect that different kinds of opponents can have on the deliberations made before negotiation. This is important, since the kind of opponent faced can lead to the selection of different issues to negotiate over (as in the flight itinerary example). Finally, none of the approaches enables an agent to make principled changes to its negotiation stance when these factors are taken into consideration.

If agents are to use negotiation effectively, several problems demand immediate attention. The first relates to the ability of agents to view negotiation in a context-sensitive manner, in order that they may prepare for negotiation appropriately given their current circumstances, needs and constraints. The second is how to make use of information about prospective opponents to inform the deliberations of an agent preparing for negotiation.

In summary, we identify a number of key areas that current research has failed to address adequately, which enables us to state a number of key goals, the satisfaction of which forms the key focus of this thesis.

- First, we aim to develop a model of what is known as the pre-negotiation stage of negotiation, so that agents can more effectively cope with dynamic domains in which it is difficult to determine negotiation issues in advance. Within this model, we have the following specific sub-goals.
- We aim to develop mechanisms to enable agents to establish their negotiation preferences without recourse to external direction or input. This is important in domains in which human control is either not possible or undesirable. To achieve

this, agents must be autonomous and be supported by appropriate negotiation goal generation machinery.

- We aim to provide mechanisms that allow an agent to establish pre-negotiation preferences that take into account, and are consistent with, its existing goals. Unlike many other models, we seek to ensure that our model of pre-negotiation is part of, and integrated with, a larger agent architecture.
- Finally, we aim to develop mechanisms to enable an agent to modify its initial pre-negotiation stance (through negotiation preferences) in the light of environmental constraints and information on potential negotiation partners. This provides a means by which agents can tailor their negotiation behaviour to the prevailing conditions in dynamic environments.

By satisfying these goals, we aim to increase the ability of agents to negotiate effectively, and to increase the efficiency of systems that require agents to negotiate with each other by minimising the number of negotiations that fail due to incompatible interests of the participants. In developing our research, we adhere to a number of principles, listed below.

- Different approaches to negotiation have been considered. The stance adopted in this work is tied intimately to the body of research that advocates the use of deliberative agent architectures rather than sub-symbolic architectures. This is because we seek to develop a model of negotiation that is, if not cognitively valid, at least cognitively plausible in the tradition of the folk psychological notions of BDI, which has had enormous success and influence.
- Where possible, it is preferable to build upon already established concepts and models rather than re-inventing new and untested approaches. By doing so, the models we develop can be integrated more easily within existing work, and the natural progression of the body of research on agents is facilitated.
- In order to avoid ambiguity and confusion, we choose to formally specify our models and theories, so that a clear understanding of our ideas can be attained. This also facilitates precise and formal reasoning about the model.
- At the same time, the formalisation of our ideas must be sensitive to the concerns of implementation, so the particular formalism adopted must be amenable to moving from the abstract formalisation to implementation of the concrete mechanisms it describes.

1.6 Contributions of the Thesis

This thesis makes three main contributions.

- First, it provides a model of negotiation goals that incorporates an analysis of negotiation issues, a deliberative preference determination mechanism and a novel use of motivational mechanisms within negotiation. This extends the state-of-the-art by making explicit the connections between deliberative agent architectures, negotiation goals and autonomous preference determination, effectively defining a new deliberative negotiation architecture.
- Second, it provides an analysis and taxonomy of bilateral negotiation issues that offers a new approach to the representation of deliberative agent negotiation interactions. In particular, it allows designers to reason explicitly about the kinds of negotiations that might arise between agents in a system.
- Finally, it provides a suite of mechanisms to enable agents to modify their approach to negotiation based upon information obtained about the negotiation context. These mechanisms are novel in that they allow explicit consideration of time and resource constraints and their expected impact on negotiation within the context of deliberative agent negotiation.

Combined, these contributions enable deliberative agents to be more effective negotiators in dynamic domains where user guidance is problematic.

1.7 Thesis Overview

We present our work on motivation and autonomy for pre-negotiation in 7 chapters. In Chapter 2 we review selected relevant work that provides the background and context for our own research efforts. Specifically, we review work on agent architectures, agent negotiation, autonomy and motivational mechanisms.

In Chapter 3 we begin the presentation of our formal ideas. As these rest on and extend an already established agent framework (SMART) we describe the important concepts defined therein and explain the formal language we use to express our ideas.

Chapter 4 details our model of negotiation goals and issues. This model forms the basis of the work that follows and provides a classification of issue types that are subsequently used in the deliberative mechanisms introduced later. We also describe in this chapter the determination of preferences over negotiation issues and how this ties into the components of the deliberative agent architecture.

In Chapter 5 we move on to an analysis of the ways in which issue classifications made by individual agents combine to produce bilateral issue types. We show the effects that such issues have on negotiation outcomes, and provide a method that enables agents to reason about the likely occurrence of such issues.

In Chapter 6 we show how the information provided by the models developed in previous chapters can be used by agents to increase their effectiveness as negotiators. Specifically we look at the constraints of time and opportunities for negotiation and how these affect the ability of agents to find agreement in negotiation. We then present mechanisms that allow agents to mitigate the effects of these constraints by modifying their negotiation stance. This chapter also provides an empirical evaluation of the mechanisms presented through the thesis. In particular, we examine how agents can increase their ability to successfully conclude negotiations under time constraints by selectively modifying their issue classifications to decrease the duration of negotiation. Finally, in Chapter 7 we summarise our work, present its main contributions and limitations before highlighting opportunities for future work.

Parts of this thesis have been presented in numerous workshops, conferences and printed publications. In particular, aspects of the motivation model presented in Chapter 3 were presented at the *Third International Central and Eastern European Conference on Multi-Agent Systems* [100]. Our ideas on motivation and autonomy that appear also in Chapter 3 were published in *Agents and Computational Autonomy : Potential Risks and Solutions* [98], *Cognitive Systems: Information Processing Meets Brain Science* [64] and *Agent Autonomy* [81]. Aspects of our work on the selection of negotiation opponents that appears in extended form in Chapter 6 have been published in the LNAI volume: *Engineering Societies in the agents World* [97] and in the *Proceedings of the First International Workshop on Rational, Robust, and Secure Negotiations* [99].

Chapter 2

Related Work

2.1 Introduction

In order to provide a background context within which the research contributions of this thesis can be understood, we provide a selective review of work by other researchers in the field of autonomous agents. Specifically, we provide some background to the field of autonomous agents by discussing current research on *agent architectures*, *negotiation*, and *autonomy*.

Reviewing agent architectures provides us with an understanding of how agents operate and how behaviour arises out of the interactions of the internal components comprising the agent. This is important for two reasons. First, as our aims include making use of, and building upon, successful agent architectures, we need to understand how such architectures operate. Second, by examining a number of influential architectures [42, 59, 38, 122], we can understand how negotiation behaviour can be generated by the internal configuration of architectural components.

Agent negotiation has been a central strand of research in agent-based computing for many years, and can take many different forms, as shown in Figure 2.1, which illustrates the areas we review in this chapter. Also shown in the figure is the orthogonal activity of *negotiation analysis*, of which we also provide an overview.

Auctions represent the most common form of agent negotiation mechanism in use today, and are often used to distribute resources and tasks among many agents. Though we do not make a contribution to this form of negotiation, we include a review of the area in order to be inclusive.

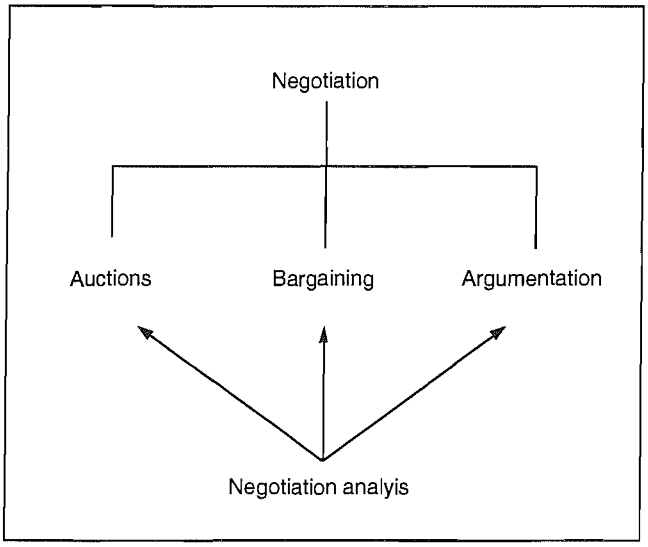


FIGURE 2.1: The different forms of negotiation

Bargaining differs from auctions in that it is usually taken to mean negotiations between two agents. In bargaining, agents exchange offers and counter-offers over various issues, such as price, time of delivery, etc, where these issues are associated with some negotiation object, such as a good, or service, for example. Our main focus is on these kinds of negotiations, and specifically how agents prepare for them. Within this, *argumentation* addresses the proposal of reasons why a particular outcome should be accepted by an agent. Such negotiation, though considered a distinct research area, can be seen as an elaboration of the bargaining process, and is a relatively new area of research that is, arguably, still in its infancy as a technique.

By contrast, *negotiation analysis* is an activity that examines different forms of negotiation scenarios with the aim of identifying optimal negotiation strategies. A major approach taken here is to apply the theories and techniques developed in *game theory*, and we provide a selective review of some of the more important and relevant results. Other forms of negotiation analysis are rare, but there have recently been some other high-level approaches, which we also outline, to highlight how they differ from game theoretic analysis.

Importantly, a defining feature of the agent-oriented paradigm is *autonomy* [152], which enables agents to operate away from human control and hence solve problems and undertake tasks where such control is limited or undesirable. This ability is important, since it is central to our concerns regarding the ability of agents to use negotiation in a flexible manner where human direction is limited or non-existent. Consequently, we discuss the *concept* of autonomy, the ways that it is understood in the literature, and methods for its implementation.

The work reviewed in this chapter is organised in the following manner. In Section 2.2 we provide some background to the field of agents. In Section 2.3 we review some important agent architectures, focussing specifically on those developed from the BDI paradigm. Next, in Section 2.4, we provide our review of the field of agent negotiation. We discuss the different forms of negotiation such as argumentation, auctions and bargaining as well as providing a discussion on negotiation analysis. Section 2.5 explains and discusses how the concept of autonomy is used within agent architectures, and how the related concept of motivation is seen as an enabler of autonomy. Section 2.6 describes several important motivational taxonomies, and Section 2.7 examines some key motivational mechanisms developed for computational agents. Finally, in Section 2.8 we present our concluding comments.

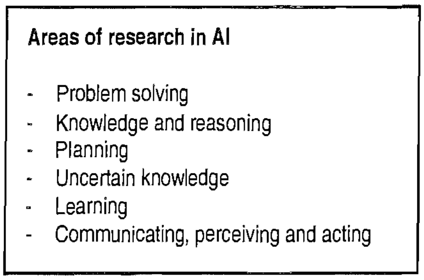
2.2 Background to Agents

The field of intelligent agents has been strongly influenced by research in artificial intelligence (AI), originally introduced at the Dartmouth Conference in 1958, where John McCarthy first coined the term [88]. Since its inception, AI has had a colourful history and has seen a number of notable achievements in application domains such as medical diagnosis [55], business [60], manufacturing [130], consumer goods [83], and entertainment [114]. However, there have also been several periods in which the image of AI has been tarnished as a consequence of its failure to meet a number of overblown predictions made of it often by the researchers themselves (e.g. [25, 92, 148]).

There are many definitions of AI offering different perspectives of the objectives and aims of the field. One such example comes from Nilsson [105]:

> *"The goal of work in artificial intelligence is to build machines that perform tasks normally requiring human intelligence"*

This definition is intentionally vague regarding the nature of intelligence, because there is little consensus on what its constituents are, and the debates around the issue are still unresolved. However, the looseness of the definitions surrounding AI has not stopped the steady increase and sophistication of the techniques and methods used, and AI has grown into a diverse and sophisticated discipline.



Areas of research in AI

- Problem solving
- Knowledge and reasoning
- Planning
- Uncertain knowledge
- Learning
- Communicating, perceiving and acting

FIGURE 2.2: Areas of research in AI

As things stand today, the field has split into a number of different but overlapping areas, each dealing with a different aspect of intelligent behaviour, and examples of the kinds of things studied in AI research can be found in any textbook. For example, Russell and Norvig [126] split AI research into the six subfields shown in Figure 2.2.

Early work in AI focussed on developing computational ways in which to represent and manipulate *symbolic descriptions* of the world and the knowledge it contains. This approach is largely based on Simon's *Physical Symbol Systems Hypothesis*, which states that a physical symbol system has the *necessary* and *sufficient* means for general intelligent action [2], although the symbolic approach has more recently been supplemented by newer approaches such as *connectionism* and the *reactive* approach. Connectionism attempts to model and explain human abilities using *artificial neural networks*, which are simplified models of the brain, composed of large numbers of units (analogous to biological neurons) together with weighted connections between the units. This approach eschews symbolic representations and instead attempts to create intelligent behaviour at the *sub-symbolic* level in which knowledge is distributed through the weights connecting the units together. The *reactive* approach also rejects symbolic representations in favour of connecting action directly to perception based on the assumption that intelligence arises through the interaction of innate behaviours with complex environments [14]. Rather than being alternative ways of achieving the same goal, these different approaches are often combined into hybrids, in which each is used to support the weaknesses of the others.

2.2.1 Computational Agents

Research on computational agents grew out of AI work as a result of a number of developments. First, with the growth in the use of the Internet, intranets and local area networks (LANs), there developed a need to cope with systems and processes that were becoming more and more *distributed*. Second, within AI itself, the focus had long been to develop *partial* models of human intelligence with the aspiration to *integrate* them sometime in the future when each of the subsystems were better understood. In an effort to meet these integrational aims, researchers began combining the different areas within AI into standalone applications. Thus, the notion of a *computational agent* became a way of focusing and integrating the disparate AI technologies in a coherent and systematic fashion. The concept of a self-contained, computational system, able to make intelligent decisions in response to localised problems and processes, also turns out to be a very powerful approach to solving the problems inherent in distributed systems. Computational agents can exist as isolated loci of control and decision making in a distributed network, responsive to the local characteristics of the system, and communicating, cooperating, and competing with each other in order to flexibly and efficiently reallocate resources and problem-solving abilities to the needs of the situation at hand. These properties of computational agents have led to a surge in the application of agent technologies to a wide range of problem domains. Since the late 1980s, computational agents have been slowly permeating into the worlds of scientific [16] and military [67] research, business applications [60], medical support systems [55] and entertainment [114].

Just like definitions for AI, definitions for computational agents abound, and no canonical definition holds sway. This is largely due to the fact that there can be many different types of computational agents. We have developed a simple taxonomy (shown in Figure 2.3) in which, from the root class of agents, two subclasses can be identified; *biological* and *computational agents*. Computational agents split into: *robotic agents*, which deal with physically instantiated robotic systems (e.g. [13, 132]); and *software agents*, which in turn splits into *intelligent agents* and *simple agents*. Simple agents comprise such entities as: computer *viruses* that occasionally gain notoriety by disrupting e-mail applications (e.g. [103, 144]); and *Artificial Life agents* which are used to study properties of living systems (e.g. [143, 96]). Intelligent agents comprise: *symbolic agents*, and consist of agents using *deliberative*, logic-based approaches and possess explicit symbolic *goals* that they attempt to satisfy (e.g. [42, 106]); and agents based around the notion of *utility maximisation*, which possess an *implicit* goal to to maximise some numeric value by pursuing various courses of action (e.g. [68]). However, these distinctions are not as clear as the taxonomy suggests and there are some hybrid agents that combine both approaches, in which symbolic goals are themselves assigned a utility (e.g. [56]).

Though agents can take many different forms, we concentrate on the sub-class of agents that are considered *intelligent* by way of their ability to *deliberate* a response to an environmental situation, in pursuit of some *explicitly defined symbolic goal*. Thus, the

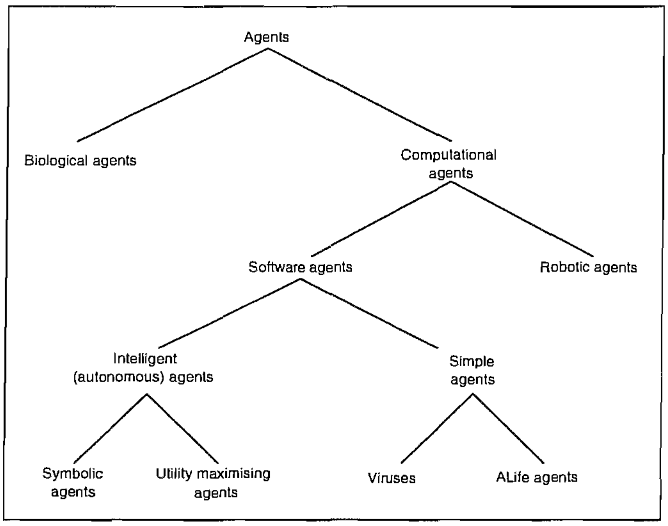


FIGURE 2.3: A taxonomy of agent types

agents that we are interested in are represented in the taxonomy as *symbolic agents*; that subclass of intelligent (sometimes called autonomous) agents.

An often used definition of intelligent agents, and one we adopt in this thesis, is presented by Wooldridge and Jennings [152], which states that an intelligent agent should have the following four characteristics.

Autonomy Agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state.

Social ability Agents interact with other agents (and possibly humans) via some kind of agent communication language.

Reactivity Agents perceive their environment (which may be the physical world, a user via a graphical user interface, a collection of other agents, the Internet, or perhaps all of these combined), and respond in a timely fashion to changes that occur in it.

Pro-activeness Agents do not simply act in response to their environment, but are able to exhibit goal-directed behaviour by taking the initiative.

One of the key defining characteristics of the agents that we are interested in is the notion that for an agent to be autonomous, it must be capable of *operating under its own direction*. This means more than simply being goal-oriented, for we contend that true autonomy derives from an ability to *generate one's own goals* in order to satisfy innate higher-order desires or pre-dispositions. These desires are the *distal* or ultimate causes of an agent's behaviour, whereas goals represent an agent's *proximate* causes of behaviour. The distal causes of behaviour are what we will call an agent's *motivations*, the set of higher-order desires that help to direct an agent's goal-directed activity; it is by virtue of their motivations that agents can be considered autonomous.

2.3 Agent Architectures

2.3.1 Introduction

There are four main classes of agent architecture that we cover in this review:

- *Reactive architectures* are characterised by mechanisms that allow for timely reactions to dynamic environments.
- *Deliberative architectures* allow *reasoning* about behaviour, and are based on folk psychology notions of human reasoning, in which beliefs, desires and intentions are seen as different kinds of mental tokens, which interact and give rise to decision-making and behaviour.
- *Social architectures* extend the deliberative model in order to address the problem of social interactions. This is achieved by including a number of components to allow the modelling of other agents, and different forms of interactions.
- *Hybrid architectures* combine reactive, deliberative and social components and commonly take a layered approach, in which each layer represents a different area of concern.

2.3.2 Reactive architectures

As the name implies, reactive architectures focus on building sets of reactive behaviours that are triggered under certain environmental conditions. An interesting aspect of reactive approaches is the ability to generate relatively sophisticated behaviours by combing simple low-level reactions. This approach is exemplified by Brook's *subsumption architecture* [14], and Agre and Chapman's *Pengi* [1] agent. Such agents, however, cannot easily interact with other agents to resolve conflicts using negotiation, and so we will say no more about them in this review (though see [15] for a discussion).

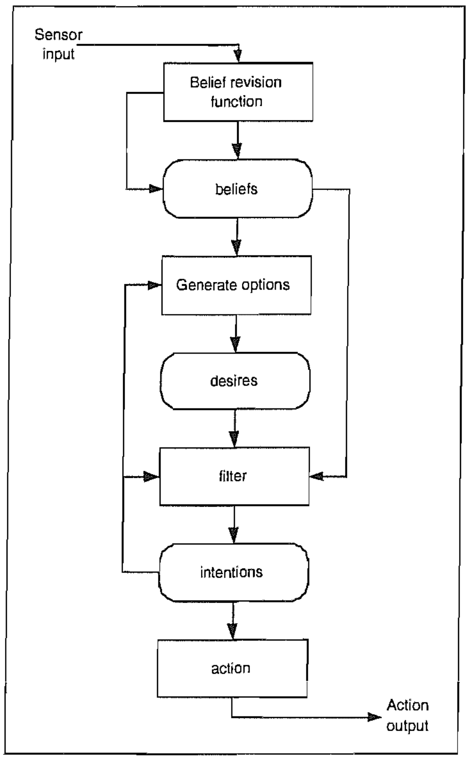


FIGURE 2.4: A generalised BDI agent architecture

2.3.3 Deliberative Architectures

Deliberative architectures are arguably the most successfully used and best known architectures in agent applications. The dominant approach for deliberative architectures is the *belief, desire, intention* (BDI) model [122] of Rao, which is built upon Bratman's theory of intentions [12]. Such agents have been used in numerous domains, including logistics [113], manufacturing [131], military operations [78] and spacecraft monitoring and control [43].

The BDI model is based on the theory of *practical reasoning* in which behaviour is driven by *goals*. Deciding which goals to pursue forms the first stage of practical reasoning (*deliberation*), and deciding how to satisfy selected goals forms the second stage (*means-end reasoning*). In essence, agents examine their beliefs about the world to determine which of their desires should be selected to form an intention, which must then be satisfied using some appropriate means. Such a generalised approach is represented by the generic BDI architecture shown in Figure 2.4, taken from [151]. In the figure, the rounded rectangles represent the three mental components of belief, desire and intention, and the rectangles are the processes that perform the deliberation and means-ends reasoning operations identified above.

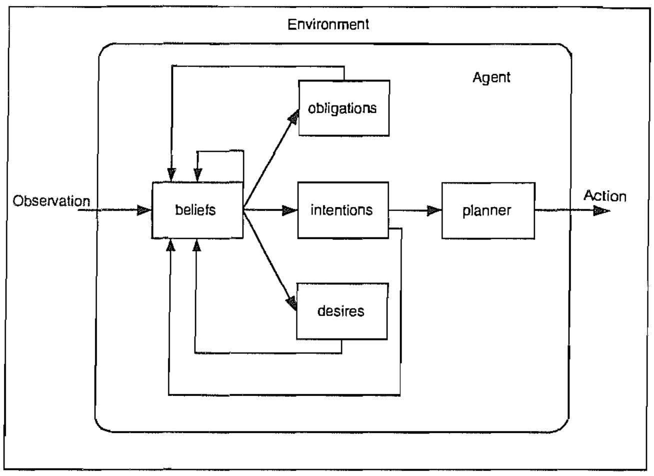


FIGURE 2.5: The BOID architecture

From this general model, a number of successful agent architectures have been developed, perhaps the two most important of which are the *Procedural Reasoning System* (PRS) developed by Georgeff [43], and its descendant, the *Distributed Multi-Agent Reasoning System* (dMARS) [26]. Several other BDI systems and models have been developed along these lines, including AgentSpeak(L) [121], an abstract programming language that can be used to implement such agents.

2.3.4 Social Extensions to BDI Model

While the first generation of BDI architectures, such as PRS and dMARS, do not explicitly deal with notions of social interaction, several models and architectures have been developed that extend the basic BDI approach to include models of other agents, communication modules and social obligations. These extensions to the BDI model can be labelled *social architectures*; we consider two such architectures below.

The first social extension to the BDI model we consider is BOID [22], which is illustrated in Figure 2.5, and from which it can be seen that, along with the traditional BDI components, a fourth component to handle social *obligations* is included. Obligations are the social commitments of one agent towards others to perform tasks or satisfy goals. They allow reasoning about tasks that an agent must fulfill for itself, as well as tasks it must fulfill for others.

With BOID it is possible to alter the kind of behaviour expressed by changing the priorities of the individual components. For example, if intentions are allowed to override

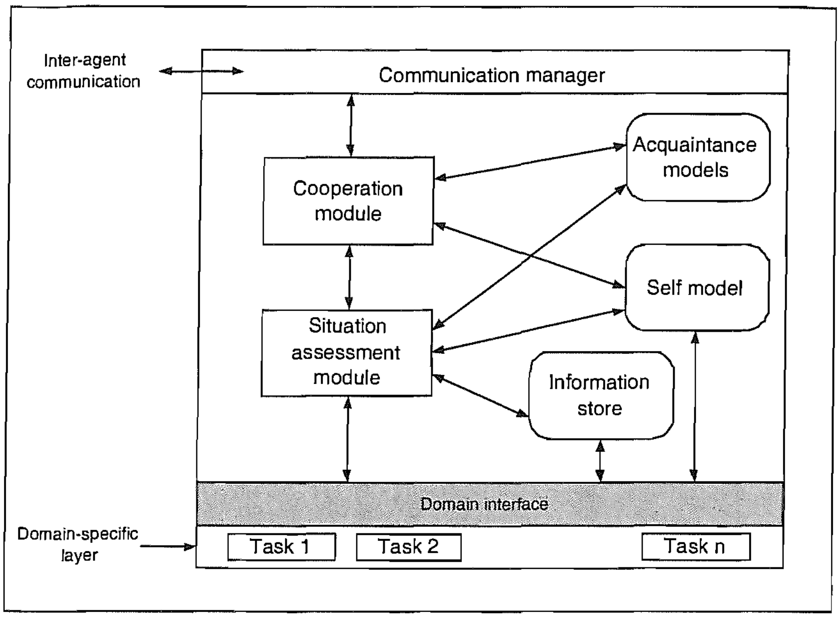


FIGURE 2.6: The GRATE* agent architecture

desires, an agent behaves in a *stable* manner, since new desires will not interrupt current attempts to satisfy existing intentions. If desires overrule intentions, then the agent displays *selfish* behaviour, and if obligations are allowed to overrule intentions then the agent acts in a *social* manner.

The GRATE* architecture, proposed by Jennings [59], differs from BOID in a number of ways. First, it is not an abstract model but an implemented architecture. Second, instead of obligations, it focuses on the formation of *joint intentions* shared by a group of agents to satisfy goals. Figure 2.6 shows the social components of the GRATE* architecture, which sit on top of a BDI-like domain layer, containing the machinery necessary for an agent to act in isolation, much like any traditional BDI agent. The social components interact with the domain layer through the domain interface.

The architecture works in the following way. Events, such as the receipt of messages, changes in the environment, etc, are analysed by the *situation assessment module*, which determines if cooperation is necessary by examining the *self model* and *information store* to see if it is possible to handle the event alone. If cooperation is needed, information about potential cooperation partners stored in the *acquaintance models* is accessed to find a suitable partner, and the agent then attempts to form a *joint intention* using the *cooperation module* and the *communications manager* (which handles inter-agent communications). If cooperation is not required, the normal BDI reasoning process is called within the *domain-specific layer*, and a *local intention* is formed.

2.3.5 Hybrid Architectures

Hybrid architectures combine reactive, deliberative and social components. One such architecture, Ferguson's Touring Machine [38], is shown graphically in Figure 2.7, which consists of *reactive*, *planning* and *modelling* layers representing, respectively, reactive, deliberative and social concerns. First, the reactive layer provides immediate responses to changes in the environment. Second, the planning layer implements a traditional means-ends reasoning system containing a library of plan schemas, which are sequences of actions and subgoals. Finally, the modelling layer contains models of the agent itself, along with models of other agents. The role of this layer is to determine when conflicts between the agent's own activities and the activities of other agents are likely to occur, and to generate new goals in an attempt to resolve them.

Each layer makes suggestions about what the agent's next action should be, while decisions about which suggestion to follow are taken by the *control subsystem*. This part of the architecture implements a number of *control rules* that either suppress information coming from a layer, or censor action outputs from a layer in order to avoid conflicts between layers.

Other such layered architectures have been developed, such as the InteRRaP architecture [95], but most take a similar approach to the one described above.

2.3.6 Discussion

Exemplified by the BDI model, deliberative architectures have been applied successfully across a wide range of domains and applications. Part of their success is due to their ability to flexibly respond to changes in the environment without the aid of human intervention. This flexibility arises from their ability to form new goals when necessary, and to satisfy these goals by engaging in context sensitive plan elaboration. In the context of this thesis, which is concerned with negotiation, the key drawback to these architectures is their lack of support for coordinating social interaction. Developments of the standard BDI model, such as the BOID model and the GRATE* architecture, overcome these difficulties by explicitly representing other agents within the reasoning process. The BOID model uses the concept of obligation to reason about tasks that must be performed for other agents, while the GRATE* architecture allows agents to reason about the capabilities of other agents, when and how to seek out assistance, and to form joint intentions with other agents to satisfy goals.

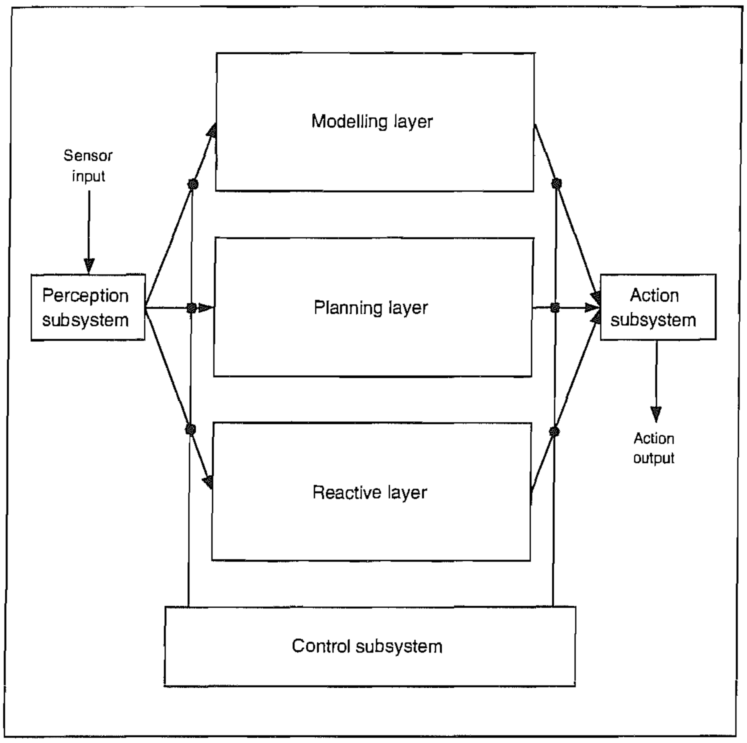


FIGURE 2.7: The Touring Machine agent architecture

Neither of these architectures, however, deals explicitly with the challenges of negotiation (though it is understood that this is not their aim). However, the ability of BDI agents to deliberate and respond to changing environmental conditions is an attractive property when considering the development of autonomous negotiators, which must also respond to the changing context within which negotiation takes place. Our aims (as stated in Chapter 1) include the development of similar kinds of deliberative mechanisms to enable agents to determine for themselves the focus of negotiation, and acceptable negotiation outcomes, by exploiting the activities and commitments the agent already possesses.

Though there are certainly other architectural approaches we could adopt (a purely utility-maximising approach, for example, that eschews explicit symbolic representation of goals in favour of an implicit utility function approach), we argue that in order for negotiation to be autonomous, the kinds of explicit representation of goals and tasks used by the deliberative paradigm are needed. The widespread use of the deliberative (typically BDI) approach means that by focusing on this paradigm we also gain maximum leverage on the applicability of our ideas.

2.4 Agent Negotiation

2.4.1 Introduction

Multi-agent systems provide a way of tackling distributed, dynamic and open problem domains. Such systems take advantage of the interdependencies between agents, exploiting both individual agent characteristics (mobility, reasoning abilities, flexible decision-making, etc), and the possibilities that arise from the ability of individual agents to interact and combine their capabilities.

As earlier stated, negotiation is a key form of interaction in such systems, and is necessary when agents have competing claims over resources or conflicting preferences over outcomes. Many different forms of negotiation are possible, and negotiation can take place between two parties or between many parties. In the former case, a bilateral bargaining negotiation mechanism can be used, while in the latter case, auction mechanisms may be more appropriate. The number of issues requiring negotiation also has an effect on the way that negotiation proceeds. Negotiating the settlement of many issues can be more difficult than if there is just one, especially if the issues interact. Participants may be more or less concerned about the equity of the settlement, some looking for purely selfish gains, some looking for a more socially equitable outcome. There are also different negotiation protocols and strategies, with those most appropriate for a given situation depending on various factors, from the dependencies between issues, to the attitudes of the participants.

In this section we outline several areas of agent negotiation, and discuss three main forms: *auctions*, *bargaining*, and *argumentation*. We also discuss two approaches to *negotiation analysis*, which is an important activity that seeks to understand negotiation in order to increase the effectiveness of its application.

2.4.2 Auctions

Auctions consists of a *good* to be bought or sold, an *auctioneer*, a set of *bidders* and an auction *protocol* [128]. The protocol defines the permissible bidding behaviour of the bidders and a payment rule that determines the price the winner of the auction pays. The auctioneer attempts to allocate the good on offer to a bidder, while maximising the price paid for the good. On the other hand, bidders try to win the good while attempting to minimise the price they pay. The auctioneer determines which protocol is to be used and the bidders use a set of bidding strategies that comply with the protocol to try to obtain the good being auctioned while maximising their utility.

There are two dimensions along which an auction protocol can vary. The first is *winner determination*, in which the final price to be paid is determined according to the payment rule used. Some rules dictate that the winner of the auction pays the price that they bid last (*first-price auctions*), while others dictate that the winner pays the second highest price bid (*second-price auction*). It turns out that second-price auctions have certain characteristics that make it desirable for all participants to bid their true value, and thus discourage deceitful bidders. The second way in which auctions can vary is in the nature of the bids: whether they are public, *open cry* bids, or private, *sealed* bids. Finally, the auction protocol itself can vary, such as whether the bidding is *ascending* or *descending*. Different combinations of these factors produce auctions of different types, the more common of which are outlined below.

English auctions are the best known auctions. They are *first-price*, *open cry*, *ascending* auctions, in which the auctioneer starts the bidding by suggesting a *reservation price* for the good (the minimum price the good can be sold for). If no bidder is willing to bid at the reservation price, the good remains unsold. If bidders are willing to bid over the reservation price, they do so by publicly announcing their bid. The last bidder to announce a bid wins the good for the price of their last bid.

Dutch auctions are *open cry*, *descending* auctions, and the auctioneer starts the bidding by announcing an artificially high reservation price. If a bidder makes a bid for the good at that price then they are allocated the good. If no one bids for the good at that price, the auctioneer successively lowers the reservation price until a bid is made, at which point the bidder is allocated the good.

First-price, sealed bid auctions are *one-shot* auctions, in which there is only one round, and each bidder makes a bid for the good. The bids are sealed, and are thus unknown to the other agents. The bidder with the highest bid is allocated the good at the price in the bid.

Vickrey auctions are *second-price*, *sealed bid* auctions. They operate in the same manner as first-price, sealed bid auctions with the exception that the winner is allocated the good at the price of the second highest bid. It turns out that this mechanism makes truth-telling the optimal strategy, since there is no incentive for the bidders to offer a bid price different from their true evaluation of the good.

With the advent of the internet, auctions have become a standard approach to automated transactions. Because they have been extensively studied in economics (e.g. [109, 70]), a great deal is known about them, and for this reason auctions are perhaps the most common form of negotiation found in multi-agent systems. Their use is widespread, ranging from the allocation of bandwidth in telecommunications networks [49], to providing recommendations to users in online settings [150]. They also have a number of desirable properties such as, for example, the possibility of identifying dominant bidding strategies (as is the case with Vickrey auctions).

2.4.3 Bargaining

In this section we focus on *bargaining*, which is a form of negotiation defined by Pruitt [115] as "*a process by which a joint decision is made by two or more parties. The parties first verbalise contradictory demands and then move towards agreements*". Bargaining can occur between more than two agents, but the common case is *bilateral bargaining*, in which there are only two.

In a series of papers ([33, 34, 134]) Faratin *et al.* develop a bargaining model for *service negotiations*, in which contracts with multiple issues are negotiated in a bilateral negotiation protocol. The work was developed in the context of the ADEPT system for business process management [62], though the model can be seen as a generalised approach to multi-issue bargaining.

Since the model is widely used and can be seen as a generalised model of bargaining, we provide a more detailed analysis than previous models. Faratin's model is composed of an *evaluation mechanism*, an *issue-set manipulation mechanism*, a *protocol* and a set of *offer-making mechanisms*. The evaluation mechanism is used to provide agents with a way to evaluate different contracts, where each contract has a number of *core* issues, and a set of *peripheral* issues that may or may not be included in a negotiation.

Figure 2.8 shows the protocol used by Faratin *et al.* as a state transition diagram. Negotiation begins when one agent approaches another and, after the initial conditions (i.e. deadlines, initial issues, etc) are set in the *pre-negotiation* stage, one agent makes an initial offer (transition from State 1 to State 2), for a contract p. The other agent can then either accept the offer, withdraw from the negotiation, or make a counter-offer. If either of the agents accepts an offer or withdraws, the protocol terminates at either of the termination points (States 5 or 6).

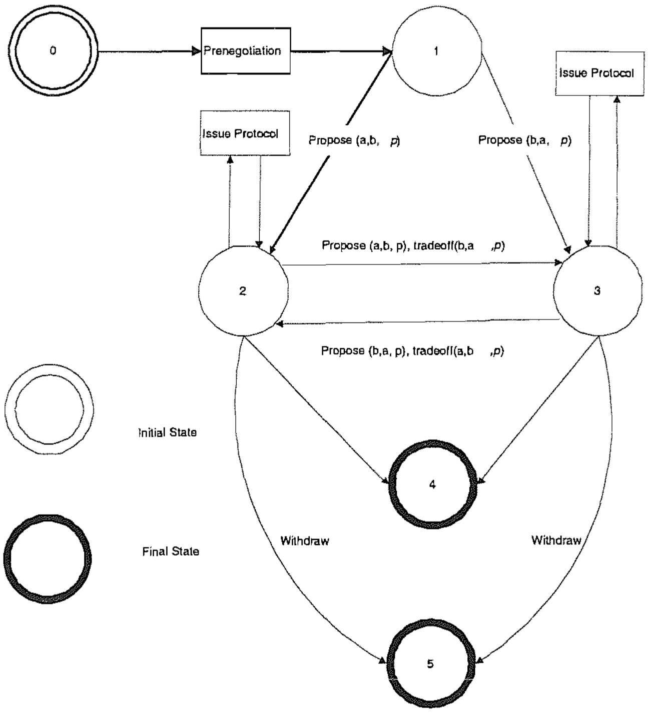


FIGURE 2.8: Faratin's bilateral bargaining negotiation protocol (after [32])

When an offer is received by an agent, it must determine whether to accept it by using the evaluation mechanism. If the agent chooses not to accept the offer, it must determine either to withdraw or to make a counter-offer that normally represents some form of *concession* over its last offer. Counter-offers are made using offer-making mechanisms, where each mechanism takes a different approach to forming the next offer. These mechanisms come in two forms, *responsive* and *deliberative*.

Responsive mechanisms react to the local information present at the time the decision is made, and generate offers by linearly combining simple decay functions, representing *tactics*:

Resource-dependent tactics model increasing levels of concession with diminishing levels of resources.

Time-dependent tactics model increasing levels of concession as the deadline for the negotiation approaches (and are a subset of resource-dependent tactics).

Behaviour-dependent tactics model concessions based on the behaviour of the opponent. For example, a large concession by the opponent may be met by another large concession (similar to tit-for-tat behaviour in the prisoner's dilemma [5]).

Deliberative mechanisms differ from responsive mechanisms in that inferences are made about the *worth* of a proposal from the opponent's point of view, rather than merely basing the next proposal on temporal or behavioural criteria. There are two kinds of deliberative mechanisms: the *trade-off* mechanism and the *issue-set manipulation* mechanism. To make a trade-off, an agent tries to find a counter-offer that has the same value as its last offer, but increases the value gained by the other agent. The use of the issue-set manipulation mechanism works in a similar way, except that here an agent adds or removes issues from the negotiation to attempt to maintain the value of the contract for itself, while increasing it for the other agent.

2.4.4 Argumentation

The kinds of negotiation presented so far consist of agents making offers and counter-offers regarding possible outcomes. In such negotiations, the participants continue with the offer-making process until one party accepts, or withdraws, or a timeout is called.

However, such methods do not use the *reasons* why an agent should prefer one outcome over another. This information is the focus of *argumentation-based* approaches to negotiation, in which agents engage in an iterative series of offers and counter-offers as before, but are also able to offer *critiques* of offers, *justifications* of positions, or *threats*, *rewards* and *punishments*. The work on argumentation-based negotiation can be split into two areas, one focusing on the use of threats, rewards and punishments (or *persuasive argumentation*), and the other dealing with the construction of logical proofs that support a position held by an agent on some matter (or *logical argumentation*).

Kraus *et al.* [69] develop a model of persuasive negotiation based on the work of Karlins and Abelson [66], in which they define six argument types: *threats*, *promise of future rewards*, *appeals to past promises*, *appeals to counterexamples*, *appeals to prevailing practice* and *appeals to self interest*. An agent *threatens* another by stating an intention to hinder some of the other agent's goals if the offer is not accepted. Similarly, a *promise of a future reward* involves an agent stating an intention to help satisfy one of the other agent's goals if the offer is accepted. An appeal to past promises tries to get acceptance of an offer by reminding the other agent that it agreed to accept such an offer in the past. An appeal to a *counter-example* can be used when an agent rejects an offer it accepted in the past without complaint. An appeal to *prevailing practices* is used when an agent refuses an offer on account of it conflicting with another of its goals. To cause the agent to agree to the goal, it is pointed out that other agents with the same existing goals have previously accepted the offer without complaint. Finally, an *appeal to self* *interest* is used when, upon receiving a rejection for an offer, the offering agent informs the rejecting agent that if it did accept the offer, another of its goals would be facilitated.

In [110], Parsons *et al.* develop an approach to logical argumentation in which agents articulate the rationale behind their particular stance on some issue, communicating it to other agents in the hope of generating consensus. A generic model of the reasoning processes that underpin argumentation is developed, based on the work of Fox *et al.* [40], in which agents argue by offering logical accounts for their positions within a negotiation. These accounts can then be attacked by an opponent using further statements to show their unsoundness. In particular, arguments can be attacked either using *rebuttals* or *undercuts*.

For example, if an argument is (φ, P) where φ represents the proposition being forwarded and P represents the set of supporting arguments for φ, a *rebuttal* occurs when an argument is found that attacks φ. An *undercut* occurs when an argument is found that attacks one of the supporting arguments in P. Rebuttals are classed as a more powerful arguments against φ than undercuts, as undercuts do not necessarily negate φ, but only weaken the support for it. Arguments are classified according to their strength as follows.

1. The class of arguments available.
2. The class of non-trivial arguments available.
3. The class of all arguments that may be made for propositions for which there are no undercutting arguments that are possible.
4. The class of all arguments that may be made for propositions for which there are no rebutting arguments that are possible.
5. The class of all tautological arguments (i.e. (φ, P), where $P = \varnothing$).

The higher the number, the stronger the argument. For example, if agent a had the intention to hang a picture with a nail that it can only get from agent b, it may make the proposition that b gives the nail to a. Agent b may either rebut this argument by making a counter-proposition that the picture does not need hanging, or it may undercut the argument by pointing out that the nail that it owns is not strong enough to hang the picture. The undercut does not directly attack the proposition to hang the picture, only the means by which it is to be hung. However, the rebuttal is a direct challenge to agent a's intention to hang the picture. Rebuttals generally take more effort to refute than undercuts and, as such, are more powerful.

2.4.5 Negotiation Analysis

Analysis of negotiation naturally begins with the work of game theory, which was developed within the field of economics and is used to examine interactions between self-interested agents. Pioneering work by von Neumann and Morgenstern [147] in 1944 has remained a powerful and influential approach to analysing interactions ever since. Here, interactions are thought of as *games* in which each *player* makes a strategic move or sequence of moves to win the game. The fundamental problem that game theory addresses is what to do given the actions available to the opposing player. Part of its power comes from the highly abstract view it takes of such interactions, in which many strong assumptions are made. For example, it assumes that players are *self interestedly rational*; that is, they will always act in a way that *maximises their own utility*. Players are also assumed to have complete knowledge of all possible outcomes in advance, and complete knowledge of an opponent's preferences over those outcomes. Given these assumptions, game theory has provided an impressive array of analytical solutions to many problems by showing how optimal outcomes can be identified.

One such solution that is particularly pertinent to our work is Nash's *axiomatic bargaining solution* [102], in which a unique solution to the bargaining problem is established in the context of a set of conditions or *axioms* that, Nash argues, any solution should satisfy. The four main axioms that Nash proposed are as follows.

Independence of utility calibrations The outcome of bargaining negotiation should not change even if the participant's evaluation of the outcome is linearly transformed. Thus, if the evaluation of an outcome is given by $eval(x) = x$, (where x is the outcome) any linear transformation of *eval* (for example, $eval(x) = 2x$), has no impact on the outcome.

Symmetry If both participants have the same evaluation function, *eval*, then the outcome must have equal value to both.

Pareto optimality Any outcome must not be inefficient, meaning that the outcome must not leave behind gains that would make one or both participants better off.

Independence of irrelevant alternatives Once a particular outcome has been rejected in favour of a better outcome, its reintroduction will never alter the final outcome. Thus, for example, if outcome C is rejected in favour of outcome B, which in turn is rejected in favour of outcome A, the reintroduction of C as an alternative can never be accepted in favour of A.

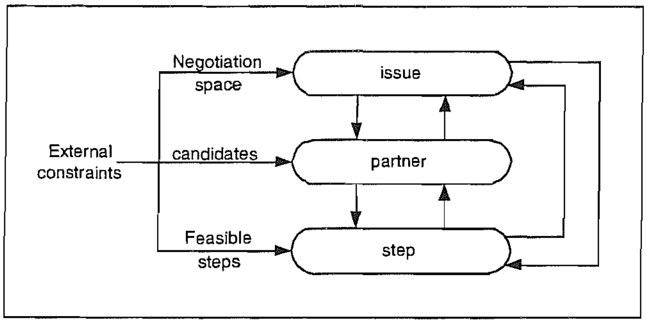


FIGURE 2.9: The C-IPS model of decision interdependencies

Nash showed that under these conditions one unique solution can be identified. However, not everyone accepts that the axioms are reasonable in all circumstances, and others such as Kalai and Smorodinsky [65] have developed different axioms that lead to different solutions. One approach that identifies the unique solution to the bargaining problem without the need for axioms is given by Rubinstein [125]. This approach shows how, by reasoning about the possible sequences of offers that can be made, one offer can be identified as the best possible, which is put forward in the very first round of bargaining and accepted immediately. The solution depends on the existence of time constraints that lessen the value of the outcome as time proceeds. The proof of this solution relies on a *hidden assumption* that there will exist a round of offers such that both participants' beliefs about the value of the outcome will be aligned; that is, both will have exactly the same beliefs about the value of the offer to both, which is common knowledge. This knowledge is needed in order to begin the backwards induction necessary to identify the unique optimal outcome.

Like many approaches that utilise game theoretic techniques, both the Nash and the Rubinstein solutions depend on a number of strong assumptions (the axioms of the Nash solution, and the hidden assumption of the Rubinstein solution). It turns out, however, that often these assumptions cannot possibly hold in the real world, where players are typically limited in the knowledge they possess about a game, their opponents and their opponents' evaluations of the outcomes. Despite this, the theory is still extremely useful, and remains an influential approach.

An alternative approach to analysing negotiation is given in [146], in which Urbig *et al.* present the C-IPS model, which provides an analysis of the interdependencies between the *constraints* on a negotiation and the selection of *issues*, *partners* and *steps* (or offers). The approach differs from game theory in that it does not try to identify optimal bargaining strategies and unique solutions, but rather considers the interdependencies between the different choices an agent makes over issues, partner selection and

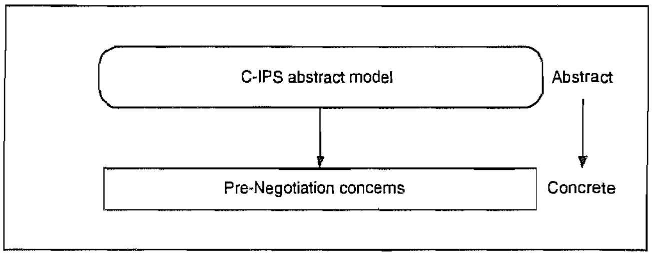


FIGURE 2.10: Relationship between C-IPS and pre-negotiation

negotiation offers. It is distinctive as it explicitly recognises the importance of such interdependencies on the process of negotiation, which has not previously been examined.

The C-IPS model is shown graphically in Figure 2.9, where external constraints affect the space of decisions of an agent for selecting issues, partners and negotiation steps. For example, the negotiation space determines the issues that can be chosen; the set of available candidates limits the choices of negotiation partners; and the set of feasible steps that can be taken limits the choice of the offers that can be made. The interdependencies between the issue, partner and step choices also affect the decisions that can be made for each. Thus, the selection of a set of issues determines the set of available partners (in that not all potential partners will be able to negotiate over the issues selected). The choice of partner then affects the kinds of steps or offers that can be made. By developing a framework for representing such dependencies, a level of analysis can be performed that is higher than that provided by game theoretic approaches. Urbig *et al.* also argue that by separating out the three decision processes of issue, partner and step, the approach facilitates the design of agent negotiators.

The work in the C-IPS model can be seen as a high level view of negotiation, exploring the relationships that can arise between constraints, issues, partners and negotiation steps. Although it provides a good abstract representation of these factors, it becomes possible to identify more issues that can be explored by drilling down into the details. We argue that these issues are best explored from the perspective of pre-negotiation, and the kinds of problems we investigate in this thesis can be seen as addressing the issues and problems identified in C-IPS at a higher granularity. The relationship between C-IPS and pre-negotiation can be seen in Figure 2.10, where C-IPS takes a higher level view of pre-negotiation.

2.4.6 Discussion

Auctions offer a very simple, but powerful, approach to negotiation. In effect, they provide a mechanism whereby multiple one-to-one negotiations can take place between the auctioneer and each bidder. However, the nature of auctions means that they are most readily applied to negotiations over one criterion, most commonly price, although increasingly work is extending their use to multiple criteria (e.g. [24]). The large amount of analytical work that focuses on auctions has now made them relatively simple to implement since their mechanics are well understood, resulting in them being the dominant negotiation mechanism in use today in agent systems. Although popular, we do not consider auctions in our work, since we are not concerned with the development of novel auction protocols that are then imposed on the participants, but rather on the reasoning of individual agents in one-to-one interactions, and how they come to identify their concerns in a forthcoming negotiation. For this reason, we focus solely on bilateral interactions in which considerations of the opponent can influence and alter the way an agent prepares for a bargaining encounter.

Faratin's bargaining model (examined in Section 2.4.3) is arguably the best known bilateral bargaining model. This is partly due to its comprehensive approach, dealing not only with the actual process of negotiation, but also touching upon some of the challenges of pre-negotiation, i.e. when issues are determined and preferences over outcomes are established. The model is presented as a *wrapper* component (so called since it wraps around an agent) that becomes active when there is a need for negotiation. In this way, Faratin argues that it can be readily incorporated into many agent architectures, since it can operate independently from them. However, this independence means that it is not clear how the activities of the agent, taken generally, interact and influence the wrapper's operation. Since our aim is to show how such factors can be exploited so that negotiation becomes integrated within the agent's broader concerns, the approach taken by the model for the pre-negotiation stage is insufficient for our needs.

A recent negotiation framework developed by Bartolini *et al.* [10] provides a modular approach to modelling negotiations of many different types. The framework incorporates a generalised interaction protocol that can be specialised to fit the application domain using sets of declarative rules. The work provides a flexible approach to modelling negotiation without the need to adopt a fully fledged coordination mechanism. The aim differs from ours in that the focus is on providing a general negotiation framework, but it does not consider the reasoning mechanisms required by agents in pre-negotiation to ensure effective preparation for negotiation.

Argumentation is a more recent approach towards negotiation. As yet, however, the techniques developed for agent argumentation are relatively new, and it remains to be seen how effective and widespread such an approach will be. Specifically, there is as yet no clear implementation strategy or application context, and most models remain at the abstract level. For these reasons, we do not consider argumentation further.

The game theoretic analyses of bargaining provide analytic solutions to the bargaining problem and remain of great influence and popularity. However their applicability is often limited due to the strong assumptions they impose. Despite this difficulty they offer *yardstick* solutions against which more practical approaches can be measured. By contrast, the C-IPS approach to negotiation analysis aims not to identify optimal outcomes, but to explicate the interdependencies between issues, negotiation partners and the offers made during negotiation [146]. As such, it is a conceptual tool that can be used by designers of agents to facilitate the development of negotiation mechanisms.

Missing from both of these approaches are tools to enable agents to identify for themselves the kinds of negotiation they are facing. This requires classification schemes that identify negotiations with different characteristics based upon the nature of the negotiation and the kinds of negotiation opponent the agent is facing. Such a means of analysing and categorising negotiations would allow agents to modify their approach so that they can avoid the kinds of negotiations that are unfavourable to them and, more importantly, increase their ability to engage in successful negotiations.

2.5 Autonomy and Motivation

2.5.1 Autonomy and Goals

In Section 2.3 we described several autonomous agent architectures. Yet, as outlined in Chapter 1, there are two distinct views of autonomy, as *independence* and as the *absolute enabler* for the generation of goals. Our view is aligned with this latter approach and, from this perspective, even if the achievement of a goal depends upon the participation of others, it does not affect an agent's ability to generate it. This does not negate the independence view of autonomy, but rather separates autonomy from dependence and treats them both as distinct concepts without overlap.

Since our focus in this thesis is on agents that can autonomously generate goals for negotiation, and determine for themselves the constraints and preferences over those goals, we focus here on the generative approach. In particular, we focus on *motivation* as a central concept and review several motivational taxonomies and mechanisms.

2.5.2 Motivation

A concept often associated with the goal generation view of autonomy is *motivation*, which is taken to represent the higher-level desires of an agent, and performs the task of influencing which of an agent's goals are generated. Given our earlier discussion, we will not labour an analysis, but more details and discussion of the contrast between the autonomy as independence view and this generative-based perspective can be found in [81].

From the generative view of autonomy, agents are autonomous if they have the ability to generate their own goals. Different goals may be generated by different motivations, naturally leading to the construction of *motivational taxonomies*. Within the field of psychology, several researchers have developed such taxonomies (for example, [74, 85]), with perhaps the best known being Maslow's *hierarchy of needs* [85].

According to Halliday [48], motivation does not refer to a specific set of readily identified processes, though for practical purposes it can be discussed in terms of *drives* and *incentives* which, respectively, are the *pushes* and *pulls* of behaviour. Drives are internally generated signals that tell an organism when it has violated a *homeostatic balance* such as hunger, thirst, etc. By contrast, incentives originate outside of an organism, can be embodied by any object, entity, or situation, and can vary in their attractiveness, arousing more or less motivation. Furthermore, incentives can be both positive and negative. For example, a positive incentive usually causes *approach* behaviours, such as a person deciding to buy a car due to the attractiveness of its specifications, whereas a negative incentive causes *avoidance* behaviours, such as a shy person avoiding social interaction.

Motivation has long been seen as a key concept in the organisation of behaviour within the *psychological* and *ethological* sciences. In *cognitive psychology* researchers come close to the meaning of motivation that is required for computational systems. For example, Kunda [72] informally defines motivation to be "any wish, desire, or preference that concerns the outcome of a given reasoning task", suggesting that motivation affects reasoning in a variety of ways, including the *accessing*, *constructing* and *evaluating* of beliefs, evidence, and information.

2.6 Motivational Taxonomies

2.6.1 Maslow's Hierarchy of Needs

Maslow's motivational taxonomy consists of an ordered list of five motivational sources of influence, called the *hierarchy of needs* which, Maslow argued, act as the forces underlying all human behaviour. The needs are: *to satisfy physiological requirements*; *to ensure physical safety*; *to form affiliative bonds*; *for achievement*; and *for self actualisation.* In Figure 2.11, we can see these represented as a five-tiered pyramid, with each level representing a different form of need.

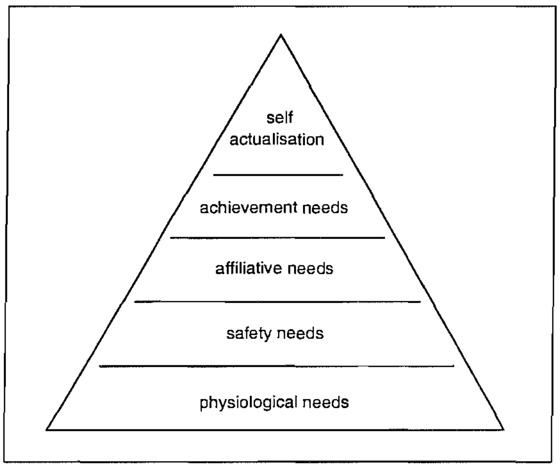


FIGURE 2.11: Maslow's hierarchy of needs

According to Maslow, needs are ordered from the lowest level of the pyramid to the top, with each only becoming active when the need immediately below it is satisfied. Thus, for example, it is only when a person has satisfied his physiological requirements that he can begin to worry about safety, and only when safety needs are met can he start to think about satisfying affiliative needs, and so on.

2.6.2 Morignot and Hayes-Roth's Motivation Taxonomy

In an attempt to map Maslow's hierarchy onto the agent domain, Morignot and Hayes-Roth [93, 94] developed a taxonomy of motivations for agents, where each motivation corresponds to a layer in the hierarchy, as shown in Table 2.1. Starting from the bottom, physiological needs map onto the concerns an agent may have about maintaining its energy levels. This relates to physically embodied agents, but could also map to

TABLE 2.1: Maslow's motivational need model mapped onto agent concerns

Human	**Agent**
Self-actualisation	Exploring
Achievement	Achieving own goals
Affiliation	Achieving other agents' goals
Safety	Avoiding harmful states
Physiological	Maintaining energy levels

concerns of software agents about maintaining any resource (for example bandwidth, money, and so on). Safety maps to concerns regarding the avoidance of harmful states. Again, this refers to physical robots that might be concerned about staying away from treacherous environmental features such as holes, or excessively hot areas. Affiliation maps to concerns for the achievement of other agents' goals, and thus relates to the notions of cooperation and benevolence. Achievement refers to the desire of an agent to satisfy its own goals, and can be linked to the notion of selfishness. Finally, self-actualisation is mapped to exploration, again in terms of physically embodied agents. For software agents, this could map onto any actions that result in extra information about the environment or other agents being gained.

2.6.3 Ferber's Agent-Based Taxonomy

Ferber [37] proposes an alternative motivational taxonomy that describes a number of different motivational sources of concern for agents. The taxonomy is a part of Ferber's agent model in which motivations are seen as part of an agent's *conative system* — that part of an agent that determines what actions it should take.

The taxonomy describes five sources of motivation: *personal*, *environmental*, *social*, *functional* and *relational*.

- Personal motivations include those things that give pleasure to an agent, as well as any task the agent has formed *commitments* to achieving.
- Environmental motivations refer to the desirability of objects or situations, and thus constitute an example of *incentives* since they *pull* the agent towards action.
- Social motivations provide the ways in which an agent's society or organisation can exert influence upon it, and are often instantiated through the use of *deontic* injunctions, such as *obligations* [9] and *societal norms* [77].

- Functional motivations refer to the tasks an agent may be charged with undertaking, and represent its core capabilities. They also belong to the set of activities for which the agent derives pleasure from satisfying and, thus, coincide with personal motivations.
- Relational motivations refer to the influence that other (single, as opposed to group) agents can impose. Here, goals are adopted in response to requests and demands from peer agents.

While the taxonomy provides an interesting set of motivational sources of influence, the categories described by Ferber are somewhat vague and overlapping, and it is not clear, for example, how functional motivations differ from personal ones.

2.7 Motivational Mechanisms

2.7.1 Early Notions of Motivation for Autonomous Agents

While much work has been done within psychology to categorise and explicate motivational processes, only now are they beginning to be put into a *computational context.* One early discussion of motivation in the context of autonomous agents is given by Simon [136], who takes motivation to be "*that which controls attention at any given time*," and explores the relation of motivation to *information-processing behaviour.* In a development of Simon's ideas, Sloman argues explicitly for the need for *motives* in computational systems [138]. For Sloman, motives represent the forces acting on an agent's decision-making, and include *desires*, *wishes*, *tastes*, *preferences* and *ideals.* Importantly, Sloman distinguishes between two types of motives, *first-order motives*, which directly specify goals, and *second-order motives*, which generate new motives, or resolve conflicts between competing motives. This relatively early work presents a picture of a *two-tiered* control of behaviour: *motives* occupying the higher level, providing the drive or urge to produce the lower level *goals* that specify the behaviour itself. In subsequent work, the terminology changes to distinguish between non-derivative motivators or *motivations*, and derivative motivators or *goals*, rather than between motivators and goals themselves [11]. Nevertheless, the notion of *derivative* and *non-derivative* mental attitudes makes one point clear: that there are *two* levels of attitude, one which is in some sense innate (i.e. motivations), and which gives rise to the other (i.e. goals).

2.7.2 Motivated Autonomy

From a more computational perspective, d'Inverno and Luck describe the SMART framework for autonomous agent systems, in which *autonomy* is associated with *motivation* [28]. For an agent to be autonomous, they argue, it must have the ability to *generate* goals, with motivation providing the mechanism to do so. For example, the motivation of safety can be associated with the goal of avoiding obstacles which, in turn, can be associated with the actions required to achieve such results. d'Inverno and Luck also describe how motivations may vary over time according to the internal state of an agent. For example, if an agent spends a long time without food, then the hunger motivation will increase, but when the agent feeds, the hunger motivation will decrease.

In [80] Luck *et al.* offer an example model of motivation, which includes the notion of motivational strength or *intensity*. The intensity of a motivation can either be variable, depending on external and internal factors, or fixed at some constant value. Motivations are represented as a triple, $\langle m, v, b \rangle$, known as an *m-triple*, where m is the type of motivation (for example, greed or curiosity) and is drawn from the set of available motivations, $m \in M$ for that agent, v is a real number representing the intensity of the motivation, and b is a boolean value, which is either *True* when the strength value, v, is fixed, or *False* when it is variable. Autonomous agents are then described as embodying a set of n motivations, each of which is represented as an *m-triple*. Thus, the types of motivations in M define the agent being considered, while each motivation in M has an intensity that depends on both the state of the agent itself and its environment. Lastly, the model also discusses the possibility of using *intensity thresholds*, which must be exceeded by the intensity of an associated motivation before that motivation is allowed to influence the goal generation process.

The work of d'Inverno and Luck represents a *framework* for agent design, and is therefore necessarily general and broad in its scope. The discussion of motivation is, thus, relatively abstract, allowing many possible implementations beyond the examples provided and, as such, offers a general model from which to proceed to more detailed, implementable designs.

2.7.3 Motivation and Pro-Active Behaviour

In [107], Norman and Long describe a motivational mechanism for the generation of both *reactive* and *proactive* goals within a BDI framework. They argue that BDI architectures are only able to generate reactive goals, by which they mean goals created in response to changes in beliefs which, in turn, are dependent upon changes in the environment. For example, an agent may generate a goal to restock a warehouse because of an environmental change that leads it to form the belief that the warehouse is empty. In this view, obtaining proactive goals from standard BDI architectures is difficult and problematic. This is because proactive goals require that an agent can predict *what* environmental states will occur, and also *when* they will be manifest. Assuming that this is possible, there is still the problem of how an agent should represent its expectations about which future environmental states will evolve. To do this in a standard BDI architecture, an agent would have to hold beliefs about what it will expect to believe in the future. Thus, to generate a proactive goal to restock the warehouse, the agent must form the belief at time t_1, say, that at some later time, say t_2, it will form the belief to restock. However, this is not sufficient to cause the generation at time t_1 of the goal to restock, as the belief at time t_1 may prove incorrect. By contrast, if the agent waits until the belief that the warehouse needs restocking is formed, then the agent is again generating goals reactively.

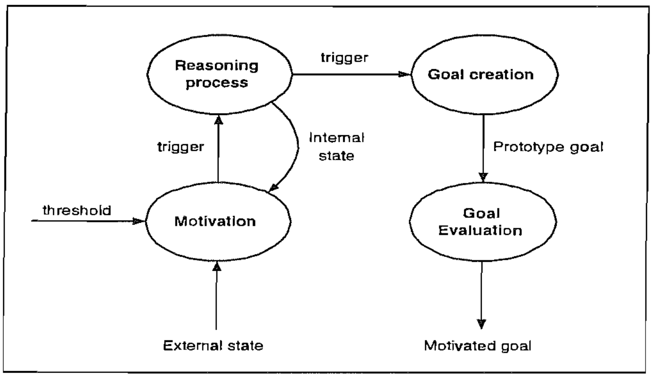


FIGURE 2.12: Norman and Long's motivated agent mechanism

To get around this dilemma, Norman and Long develop a motivational mechanism that includes *alarms*, which trigger a *reasoning process* in which an agent can *consider the necessity* of scheduling the creation of a future goal. The goal creation mechanism developed by Norman and Long is shown in Figure 2.12, in which it can be seen that information from both the environment and the internal state of an agent is passed to the motivations, which allows the agent to respond to *unpredictable* events, and leads to the creation of reactive goals. In contrast, information from inside the agent comes in the form of *predictions* about the future evolution of the environment. Furthermore, an *alarm* (which increases in intensity as time progresses) is attached to each prediction, and is used to alert the agent to the times at which it needs to consider creating a goal for an expected future event. If an alarm is intense enough, or the environmental information is considered motivationally relevant, a *deliberation goal* is triggered, which tells the agent to give further consideration to the information and to determine whether or not the generation of a goal is warranted. If a goal is required, the goal creation process is triggered. A goal is then passed to the *goal evaluation module*, in which the positive and negative effects of the goal are considered and, if the goal is evaluated favourably, it is adopted.

2.7.4 Motivating Negotiation

Recently, work has focussed on using motivational concepts within negotiation settings. In [156], Zhang *et al.* explore the use of motivation to examine a range of negotiation attitudes from self-interested attitudes to cooperative ones. Agents are given two types of motivation, one related to goals, and one related to relationships. The agents can choose to negotiate over the payment for a task by transferring *motivational quantities*, which are related either to goals or relationships. The quantities are used as tokens and, in order for an interaction to be successful, both agents must value the particular type of motivational token on offer. The strength of the relationship-based motivation represents the amount by which one agent is likely to cooperate with another. As relationship motivations increase in strength through the transference of tokens, the agents become more likely to cooperate in future encounters. This approach highlights the manner in which motivation can be used to represent an agent's concerns in a flexible and dynamic manner. As the relationships between agents change through repeated interactions and the transference of motivational tokens, the nature of the negotiations also changes (from competitive to more cooperative). The notable feature of this research is in the way it uses motivation to adapt the approach taken to negotiation in response to the context within which it takes place.

2.7.5 Discussion

In this section we have introduced some of the basic notions of motivation, described some influential motivational taxonomies and have presented a number of mechanisms that allow motivation to be used in agent architectures.

The inclusion of the concept of motivation in agent architectures allows for a higher-level of control than is possible with traditional goal centered approaches. Motivated agents are able to generate goals that are appropriate to the current context, and can distinguish between goals in terms of the value they offer. Traditional symbolic architecture do not make this distinction between goals, since all goals are of equal worth (but see [56] for an example of a utility-based approach to address this). This makes it difficult to address domains such as negotiation, which often deal with the relative worth of different outcomes or offers. Motivation thus offers a way to increase the flexibility of deliberative architectures by providing the means whereby agents can express preferences between different goals, where these preferences reflect the current needs of the agent.

However, one area that has not received much attention is the use of motivation in *mediating* agent interactions. The scope here is great, with motivations providing a means by which the reasons underlying such interactions can contribute to, and constrain, the processes of interaction. In particular, there seems to be much to be gained from adopting a motivational approach to the structuring of *negotiation*. Motivation can provide the necessary means by which an agent can make distinctions between different negotiated outcomes in terms of its motivational needs. However, despite the desirability of taking a motivational approach to negotiation, little work has been done to explicitly address this issue.

2.8 Conclusion

A number of points can be made in summary of the reviews presented in this chapter. First, we have seen how deliberative agent architectures have been successfully used in many domains, and how extensions to enable social interactions have been developed by several researchers. We argue that such architectures provide the basis for the development of deliberative mechanisms that allow agents to approach negotiations in a manner consistent with their other, existing activities.

While auctions are an important existing method of agent negotiation, and argumentation is a promising new area of research, it is bargaining that fits best with our aims of increasing the level of autonomy for agent negotiators. Bargaining offers us the right kind of context in which to explore the relationships between an agent's current activities and its aims and desires regarding the negotiation outcome. Moreover, the nature of such interactions, being generally one-to-one, allows us to explicitly consider the opponent in deliberations about how an agent should best prepare for negotiation.

However, a difficulty in applying deliberative approaches to negotiation is the lack of methods to allow agents to express preferences between different outcomes. Motivational mechanisms seem to offer an effective approach to take here, and we have reviewed work that shows how such motivational mechanisms can be integrated within deliberative architectures to allow the expression of such preferences. Though motivation has been used in such architectures in several contexts, it has rarely been used to explore the challenges associated with negotiation, and has not at all been used to develop techniques to facilitate the kinds of decisions that must be made in pre-negotiation.

In summary, therefore, we build on existing state-of-the-art work on agent architectures, negotiation, and motivation in order to develop a new formal model of negotiation goals and to provide mechanisms that allow agents to tailor their approach to negotiation.

In order, therefore, to satisfy these aims to improve the effectiveness of agent negotiators, the following key points must be addressed.

- A flexible and coherent method of expressing preferences within deliberative agent architectures must be developed if they are to be able to reason effectively about negotiation and their outcomes. Motivational mechanisms seem best suited to this, and a method for applying them to the negotiation activity must be developed.
- Negotiations must be integrated within the current activities and context of the negotiator. The determination of what to negotiate about and what outcomes are acceptable must be made in relation to existing preferences and activities. This ensures the autonomy of the agent, and ensures the approach adopted for negotiation is sensitive to the agent's current situation.
- Considerations of the negotiation opponent and other environmental factors must be integrated within the reasoning conducted about the best approach to take to negotiation. This can increase the chance that negotiations will end successfully.

Chapter 3

Motivation and Pre-Negotiation

3.1 Introduction

Our aims in this thesis lead us to consider frameworks that provide us with a complete set of expressive foundational concepts that ideally are clearly defined and suffer no ambiguities. There are several frameworks that define many of the necessary concepts of negotiation (e.g. [33, 68, 10]), but these typically do not attempt to try to develop a complete agent model, and instead focus only on those concepts that are central to negotiation activity. Moreover, they are often constructed with a view to a particular type of negotiation rather than presenting a general framework and, furthermore, they typically do not focus on deliberative agents but on utility maximising agents. Our approach, therefore, is to examine how negotiation can be integrated into a generalised agent model and, in doing so, we employ the SMART agent framework, since it offers several distinct benefits. First, SMART provides a high level approach to the modelling of agents and agent systems, in which commitments to any particular instantiation of such concepts are minimal, which gives us the freedom to extend the framework to incorporate our ideas without unnecessarily restricting our approach. Second, SMART includes the important concept of *motivation*, which is used as an enabler of autonomy through the process of *goal generation.* Since we are concerned with agents that use negotiation in an autonomous manner, we require that *all* aspects of negotiation be related to the broader context of an agent's existing activities, so that it can be used effectively and in a manner sensitive to ongoing concerns.

The work in this chapter forms the foundations that the rest of the work builds upon, and has the following structure. Section 3.2 introduces the Z specification language, Section 3.3 discusses issues surrounding agent frameworks as well as introducing the

SMART agent framework. Section 3.5 introduces our motivational extension to the SMART framework and Section 3.6 concludes.

3.2 Formal Methods and Z

Many of the concepts and definitions within agent research have long been contested [41], with much that has yet to be agreed upon. When developing new models and theories, therefore, it is important to be clear and unambiguous in the presentation and use of terms so that no misunderstandings arise. Using natural language to describe theories and models cannot offer this level of preciseness, since many terms are unclear and open to different interpretations, and so it is often necessary to adopt more rigourous approaches.

To aid the presentation and understanding of computational models, *formal methods* play an important role. Formal methods provide ways to describe computational systems in clear and unambiguous terms and thus helps the designer avoid inconsistencies, ambiguities and incompleteness in the development and presentation of concepts and ideas that might otherwise be introduced by using natural language. By adopting the clear and unambiguous vocabularies that formal methods provide, both the development and the dissemination of models is improved, since inconsistencies can be checked (mechanically in some cases [155, 142]), properties of the system can be verified, and communication of the ideas developed is facilitated.

A number of different formalisms are available, for example temporal logics, deontic logics and modal logics. All build upon and extend first order predicate logic and are being used to develop agent theories across the range of agent activity. A difficulty that many of these approaches have, however, is in moving from the formal specification of models to their implementation in computational systems. Fortunately, formalisms have been developed that attempt to bridge the gap from formal abstract specifications to real-world implementation. One such formalism that does this particularly well is the Z specification language, which we adopt in this thesis to describe the models we develop.

3.2.1 The Z Specification Language

In order to meet our requirements for clarity and precision in the development and description of our architecture, we present our ideas formally. In choosing a formal language, there are several options. Formal languages cover a wide spectrum, from the relatively abstract yet expressive logics, to languages that lie closer to implementation details but suffer in their level of expressibility. For our aims, we need a language that provides enough expressibility to allow us to model the kinds of concepts relating to agents and negotiation, but that is also at a level that allows considerations of implementation issues. One language that meets these concerns is the Z specification language [141], which combines expressivity with features that facilitate implementation. Z also has a number of other benefits, as follows.

- By making use of such long established and well understood formalisms as set theory and first order predicate logic, Z is more accessible than many other formalisms.
- It is an extremely expressive language allowing a consistent, unified and structured account of computer systems.
- It is gaining increasing acceptance in the AI and agent community (e.g. [21, 45, 79, 53]).
- Z is well supported both in written publications such as books (e.g. [75, 141]), and software support such as animation tools (e.g. [142]), type checkers [155], and case studies [54].

Below, we provide a brief overview of the Z specification language. We present only the foundational concepts in order to facilitate understanding of the subsequent formal presentation. There are many publications that outline Z in exhaustive detail (e.g. [75, 141]), and we direct the reader to these for a full account of the language. Our presentation borrows heavily on the work of d'Inverno and Luck in [28].

3.2.2 Schemas

Z distinguishes itself with the introduction of the concept of a *schema* that allows for a modular approach to the construction of a specification. Schemas are data structures that comprise two parts: an upper, declarative part that defines a set of state variables and their types, and a lower, predicate part that defines the restrictions and relationships between the variables and types introduced in the upper part. For example, the schema, *Sum*, defines three state variables as natural numbers and describes the relationship between them in the predicate part of the schema. Thus, we have the three state variables, *x*, *y*, and *z*, which represent three natural numbers. In the predicate part we define the relationship between them by stating that *z* is equal to the sum of *x* and *y*.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\mathit{Sum}} \\ \hline
x, y, z : \mathbb{N} \\ \hline
z = x + y \\ \hline
\end{array}
$$

Any state variable represented in a schema can be identified by using a dot notation to join the name of the schema in which the state variable was defined with the variable. Thus, *Sum.x* indicates the variable *x* defined in the *Sum* schema. In Z, it is possible to include one schema within the definition of another, a technique called *schema inclusion.* This allows for a compositional approach that enables the development of more complex data types.

3.2.3 Operations

Operations on state-based variables are defined in terms of *changes of state.* More specifically, an operation defines the relationship between state before the operation takes place and state after the operation has taken place. State variables after an operation are decorated (so x becomes x'), inputs are denoted by the ? symbol, and outputs by the ! symbol. For example, in the schema *InputSum*, two state variables, $a?$ and $b?$, are defined as natural numbers. Each variable is shown with the ? symbol to show that their values are inputs to the schema. The variable, $sum!$, also defined as a natural number, is indicated to be an output. The value of $sum!$ is shown in the predicate part of the schema to be the value obtained by adding the value of $a?$ to $b?$.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\mathit{InputSum}} \\ \hline
a?, b? : \mathbb{N} \\
sum! : \mathbb{N} \\ \hline
sum! = a? + b? \\ \hline
\end{array}
$$

When performing operations on state it may be desirable to indicate when the state variables in a schema are allowed to change and when they must not. To do this, Z uses Δ to indicate that the variables defined within the schema change, and Ξ to indicate that the variables do not change. For example, below we give the schema for *SmallBox*, which contains state variables for height and width, both represented as natural numbers.

SmallBox

$height, width : \mathbb{N}$

$height = 5 \land width = 10$

We can define another schema, *ScaleBox*, and use *SmallBox* through schema inclusion. In the predicate part of *ScaleBox*, we state that the width of *SmallBox* is doubled. We use Δ to show that the variables of *SmallBox* change as a result.

ScaleBox

$\Delta SmallBox$

$SmallBox.width' = SmallBox.width \times 2$

3.2.4 Given Sets

In some cases, it may be necessary to introduce a new type without giving any details. For example, we may wish to introduce the notion of an *action* without saying anything about how an action may be represented. To do this we can introduce a *given set*, which allows the introduction of a type without the need for further definition, and is essential in 'bootstrapping' a specification so that the model being described has a foundation on which to rest. Thus, to represent the notion of an action without further detail we can use the following form.

$[Action]$

3.2.5 Relations, Functions and Sequences

Relations are ordered sets of pairs $\mathbb{P}(X \times Y)$ between a source type and a target type (here, X and Y, where X is the *source* type and Y is the *target* type). If each element in the source type is related to only one element in the target type then the relation is a *function*. A function is *partial* if *not all* elements in the source type are related to elements in the target type, and *total* when *each* element in the source type is related to an element in the target type. The *domain* of a relation or function is that subset of the source that is related to the target. Conversely, the *range* is that subset of the target that is related to the source. A *sequence* is a function whose domain is the set of natural numbers and whose range is the set of elements in the sequence. For example, the set *Towns* contains four towns: manchester, brighton, london and cardiff.

$$Towns = \{manchester, brighton, london, cardiff\}$$

Now suppose a truck driver must make deliveries in each town but must determine in which order, or sequence, to do this. The sequence is given the name *route* as follows.

$$route : \mathrm{seq}\, Towns$$

The sequence, *route*, thus provides a *mapping* ($\mapsto$) from the set of natural numbers to the elements of the set *Towns*. An example mapping appears below.

$$route = \{1 \mapsto brighton, 2 \mapsto london, 3 \mapsto manchester, 4 \mapsto cardiff\}$$

There is also a shorthand version of the above form.

$$route = \langle brighton, london, manchester, cardiff\rangle$$

3.2.6 Set Comprehension

In Z it is possible to construct a set using a shorthand technique called *set comprehension*. Instead of enumerating all the elements of a set, it is possible, for example, to denote the set of all items x of type T such that P is true, using the following notation.

$$\{x : T \mid P\}$$

If we wish to place a restriction on the form that x takes, we can write the following.

$$\{x : T \mid P \bullet k\}$$

This denotes the set of all items x of type T in the form k such that P is true. For example, below we construct the set of the squares of the natural numbers between 5 and 10.

$$\{\forall n : \mathbb{N} \mid (n \geq 5) \wedge (n \leq 10) \bullet (n \, * \, n)\}$$

We provide a summary of the Z notation in Figure 3.1. At this point we will say no more about the Z language, but we will explain our usage of the language as we proceed through the specification of our model.

3.3 Agent Frameworks

In the development of any model, identifying a set of concepts that maintain the level of generality required while providing enough expressivity is a difficult undertaking. Where possible, it is desirable to build upon already well established models and theories in order to exploit the gains they have already made, both in their level of understanding and acceptance in the wider research community and also the concepts and theories they have already introduced. In looking for such a basis, however, we must keep in mind our aims and avoid those models that may constrain us or lead us into representations unsuitable for our domain. Two main concerns are:

- conceptual clarity and expressiveness; and
- explicit representation of the notion of autonomy.

Agent negotiators have most commonly been used to satisfy the goals of human users in various forms of purchase negotiations (for example, [50]). In such situations, it is important for the agent to be able to respond quickly and flexibly to the offer-making behaviour of the opponent within the boundaries set out by its user. This requires that agents are capable of making autonomous decisions during the course of the negotiation about which strategies to adopt and what offers to make. In general up to now, the user of the agent determines exactly what needs to be negotiated, what reservation limits exist, and so on, and only the decisions made during negotiation are the responsibility of the agent. However, in systems in which negotiation is used by agents as just one of many possible means of satisfying their goals, the identification of what to negotiate about, what reservation limits to set and, hence, the boundaries within which the negotiation will take place, cannot come directly from a user but must, instead, be determined by the agent itself.

While agent negotiation frameworks are becoming increasingly sophisticated in allowing agents to adapt and respond to the interactions within a negotiation episode, they have not, in general, addressed those problems faced by agents using negotiation as a tool for the satisfaction of their own goals. In the same manner that agents need to

Definitions and declarations

a, b	Identifiers
p, q	Predicates
s, t	Sequences
x, y	Expressions
A, B	Sets
R, S	Relations
$d;\ e$	Declarations
$a == x$	Abbreviated definition
$[a]$	Given set
$A ::= b\langle\langle B\rangle\rangle$ $\mid c\langle\langle C\rangle\rangle$	Free type declaration
$\mu\, d \mid P$	Definite description
let $a == x$	Local variable definition

Logic

$\neg p$	Logical negation
$p \land q$	Logical conjunction
$p \lor q$	Logical disjunction
$p \Rightarrow q$	Logical implication
$p \Leftrightarrow q$	Logical equivalence
$\forall X \bullet q$	Universal quantification
$\exists X \bullet q$	Existential quantification

Sets

$x \in y$	Set membership
$\{\}$	Empty set
$A \subseteq B$	Set inclusion
$\{x, y, \ldots\}$	Set of elements
$(x, y, \ldots)$	Ordered tuple
$A \times B \times \ldots$	Cartesian product
$\mathbb{P} A$	Power set
$\mathbb{P}_1 A$	Non-empty power set
$A \cap B$	Set intersection
$A \cup B$	Set union
$A \setminus B$	Set difference
$\bigcup A$	Generalized union
$\# A$	Size of a finite set
$\{d;\ e \ldots \mid p \bullet x\}$	Set Comprehension

Relations

$A \leftrightarrow B$	Relation
$\operatorname{dom} R$	Relation Domain
$\operatorname{ran} R$	Relation Range
R^{-1}	Relational Inverse
$A \lhd R$	Domain restriction
$A \rhd R$	Anti-range restriction
$R \oplus S$	Relational overriding

Functions

$A \rightarrow\!\!\!\!\!+ B$	Partial function
$A \rightarrow B$	Total function

Sequences

$\operatorname{seq} A$	Sequence
$\operatorname{seq}_1 A$	Non-empty
$\langle\rangle$	Empty
$\langle x, y, \ldots\rangle$	Sequence
$s \frown t$	Concatenation
head s	First element
tail s	All but first

Schema notation

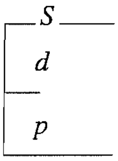
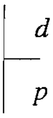
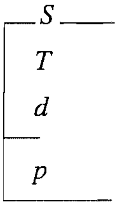
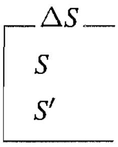
	Schema
	Axiomatic def
	Inclusion
	Operation
$z.a$	Component

FIGURE 3.1: Summary of Z notation (taken from [26])

be able to display autonomous decision-making during the course of a negotiation to respond to opponent behaviour, so too must agents using negotiation as a tool in this way be able to respond flexibly to the changing contexts in which negotiation might be needed. This requirement for autonomy means that any framework we adopt to model the kinds of agent negotiators we are interested in must allow us to reason effectively about autonomy, what it is, how it can be constrained and what effect it can have on the operation of an agent.

3.3.1 The SMART Agent Framework

To address these concerns, we adopt the SMART (Structured and Modular Agent Relationships and Types) agent framework developed by d'Inverno and Luck [28]. SMART provides a set of concepts that allow designers of agent systems to represent, model and analyse different agent systems. As such, SMART is extremely expressive, allowing many different aspects of agents and agent systems to be represented and reasoned about in clear and unambiguous terms. Moreover, the concepts defined in SMART are expressed using the Z notation, thus allowing properties of any system modelled by SMART to be formally established. To date, SMART has been used to explore such problems as *cooperation* [46], *normative behaviour* [77], *relationship analysis* [3] and *agent architectures* [4].

The SMART framework is based on a simple set of concepts that are combined and developed to represent progressively more complex and sophisticated ideas relating to agents and agent systems. In this section, we describe in more detail the foundational concepts of SMART that we will later build upon to define our own model of autonomous negotiators.

3.3.2 Attributes

SMART uses the abstract concept of an *attribute* to represent any *potentially perceivable property* of the world. Thus, by taking a portion of the world and listing its attributes, we provide a *description* of that part of the world. For example, by listing attributes of a table such as its colour and number of legs, we get a (partial) description that could be represented as follows.

```
colour (table, orange), numberOfLegs (table, 4)
```

At any point in time, a particular attribute may not actually be perceived by any one agent, but nevertheless, the attribute must be *potentially perceivable* at some time. If a feature of the world cannot possibly be perceived then it is not, under SMART's definition, an attribute.

Formally, an attribute is defined as a given set, so that nothing is said about how it is constructed.

$$[\text{SMART}\mathit{Attribute}]$$

If *all* the potentially perceivable properties of the world are aggregated together, a *complete* representation or description is produced, which contains all those things that an agent may perceive, and thus defines an agent's *environment.* Formally, SMART simply states that environments are composed of non-empty sets of attributes.

$$\mathit{Environment} == \mathbb{P}_1\ \text{SMART}\mathit{Attribute}$$

This definition says nothing about how attributes may be related. For example, although SMART can describe the fact that a table is green and that it has four legs, it cannot say that both properties (being green and having four legs) belong to the same table. However, this allows SMART to assume as little as possible about how agents' perceptual machinery may be implemented, thus guaranteeing the generality of the model.

3.3.3 Percepts

Though SMART says nothing about the *detail* of an agent's perceptual systems, it does discuss how an agent's perception can be *limited.* Any agent implementation must deal with computational limitations, and an agent's perceptual machinery is often limited in the amount of information it can process. Thus, as some parts of an agent's environment may not always be accessible through perception, SMART defines an agent's current percepts as its *view* of its environment, which may be limited.

A *view* is simply another description of the environment that is formed by the interaction between the environment and the agent's perceptual machinery. This conceptualisation thus allows SMART to give *View* the same definition as *Environment*, i.e. a non-empty set of attributes.

$$\mathit{View} == \mathbb{P}_1\ \text{SMART}\mathit{Attribute}$$

It is important to note that a view is an *internal* representation of the environment, formed by the filtering of the external environment through an agent's perceptual machinery. This distinction allows SMART to represent cases where an agent's view of its environment contains errors or is incomplete, caused by limitations or faults in the perceptual process.

3.3.4 Actions

Now, in order for an agent to be useful, it must be able to affect its environment in some manner. SMART thus defines the concept of an *action* as any discrete procedure or activity that results in a transformation of the environment by the addition or deletion of attributes. This is similar to *add* and *delete* lists introduced in Fikes and Nilsson's STRIPS planning system [39], which indicates which predicates must be added to or deleted from a logical representation of the world after a plan has been executed.

Formally, actions are defined simply as a given set.

$$[Action]$$

3.3.5 Goals

So far, we have described the way SMART defines the concepts representing an agent's environment, its view of that environment and how actions affect environments. However, as yet, we have not described SMART 's way of representing how agents determine which actions to perform. That is, agents thus far defined have no *direction*, and consequently no way to guide their behaviour towards desired ends. To give agents such direction and purpose, SMART defines *goals*, which represent the ends to which agent action is directed, and take the form of descriptions of *desired* environmental states.

Since goals represent descriptions of desired environmental states, they are formally defined in the same manner as environments, i.e. as non-empty sets of attributes.

$$\text{SMART}Goal == \mathbb{P}_1\ \text{SMART}Attribute$$

3.4 Motivation

3.4.1 Introduction

Much of computing, especially artificial intelligence (AI), is conceptualised as taking place at the *knowledge level*, with computational activity being defined in terms of *what* to do, or *goals*. Computation can then be undertaken to achieve those goals, as is typical in planning, for example. However, the reasons why the goals arise in the first place are typically not considered, yet they may have important and substantial influence over their manner of achievement. If goals determine *what* to do, these reasons, or *motivations*, determine *why* and consequently how. The best illustration of the role of motivation in computing is perhaps in relation to autonomous agents which, in essence, possess goals that are *generated* within, rather than *adopted* from, other agents [28]. These goals are generated from motivations, higher-level non-derivative components that characterise the nature of the agent. They can be considered to be the desires or preferences that affect the outcome of a given reasoning or behavioural task.

3.4.2 Domain Motivations

Since agents are reactive and pro-active, they have the ability to *respond* to the current environment, as well as the ability to display *goal-directedness*. Just what an agent is expected to respond to, and be pro-active towards, is strongly linked to the agent's *role* within the system it inhabits. Such roles will, of course, vary from domain to domain, and there is thus little to be said about their general characteristics. However, if we accept that an agent must have some *domain specific* roles that demand the satisfaction of a set of goals, then it is possible to place these tasks under the control of some set of *domain motivations*, which will facilitate the *generation* of these goals at *relevant* times and determine their relative *value* to the agent.

For example, an agent may have a domain role of maintaining the cleanliness of a warehouse. Part of this role may be to satisfy a number of goals related to ensuring that the various boxes in a collection of such boxes are stored at specific predefined locations. To control the generation of such goals, we can instantiate a domain motivation for tidiness, which is sensitive to situations in which boxes are not in their right location. At some point in time an agent may become motivated to generate a set of goals, the satisfaction of which would result in a tidy warehouse.

3.4.3 Constraint Motivations

Motivation can be used as a natural mechanism to represent the *constraints* over the ways in which a goal may be satisfied. In this way, motivation provides a natural way to conceptualise *meta-level control* processes. Thus, motivation can be used to monitor aspects of an agent's situation such as risk, cost, efficiency etc. Along with domain motivations, therefore, *constraint motivations* perform the task of imposing restrictions on the use of resources and the importance to be placed on their use.

3.4.4 Social Motivations

Social ability allows agents to combine their individual capabilities together to deal with problems that are not possible, or at least much more difficult, to solve by individual agents working in isolation. The two characteristics of action and sociability are what marks out agent-based computing from other forms of computational approaches, and are what gives the technology leverage over the complex, open and dynamic domains that are the mainstay of applications for agent technology.

Whereas most existing work on motivated agents has focused on the goal generation aspects of motivation, motivation can also have a direct effect on an agent's *interactions*. Motivation offers a natural mechanism for evaluating requests for assistance. For example, an agent in a dynamic, open environment may have many forms of relationship with other agents ranging from completely cooperative to purely selfish relationships. Moreover, an orthogonal concern is how *important* a relationship is. Importance may derive from numerous sources, such as authority, rank, or from the level of dependence of one agent to another and, furthermore, numerous gradations may exist between the extremes; an agent must have some way to quantify these measures of selfishness and importance.

The notion of the importance of a relationship between two agents is central to how they will interact. Generally, the more important one agent is to another the more it will gain from helping the agent satisfy its goals. Agents will vary from having relationships in which the utility gained by one agent in the relationship has no importance to the other agent to situations in which the utility gained by one agent is worth many times that utility to the other agent. With the above considerations in mind, *social motivations* can play a vital role in managing the social interactions of agents.

3.4.5 Motivation and Negotiation

Motivation has been discussed by various researchers in terms of goal generation [28], cooperation [47], proactive behaviour [107], norms [77], planning [19] and information processing [91]. Our main concern in this thesis, however is in the use of motivation within negotiation, and specifically how it affects the reasoning of an agent in the pre-negotiation stage. The autonomous use of negotiation is desirable in systems where it is not possible or desirable for a human controller to direct the agent's activities, such as in highly dynamic domains. This entails, however, that agents must be able to make effective decisions about how to negotiate given that they may already be engaged in other activities. In particular, the outcomes of negotiation must be examined to ensure that already adopted goals are not compromised unless there is some gain in doing so. To achieve this, it is necessary that we have some way of comparing the gains from satisfying a goal via negotiation with the gains from already adopted, or *existing* goals. It is in this sense that motivation becomes a useful construct to adopt, since we can determine the *importance* of goals in relation to their respective motivations to determine if it is worth compromising on one in order to satisfy another.

All of the above types of motivation are relevant to different aspects of behaviour. Rather than arbitrarily introducing these distinctions as part of our model for negotiation, however, we are able to proceed to consider negotiation with a single form of motivation that covers them all. The discussion merely serves to highlight how motivations may influence behaviour in different circumstances. In the next section, we build up a model of motivation based on the basic concepts introduced by SMART.

3.5 The Motivation Model

In most agent architectures the highest level of analysis relating to agent activity is defined by an agent's goals. In contrast, SMART includes a further level of analysis by incorporating the construct of *motivation*, which represents higher-level desires and is the source of goals. Actions are taken to satisfy goals which, in turn, satisfy or mitigate motivations. This extra component determines *why* an agent may be working to satisfy some goal, and provides the conceptual framework for the development of mechanisms to enable and control *autonomy*. By introducing motivation, it becomes possible to provide a rationale for why an agent might attempt to satisfy a given goal. For example, an agent with a goal to replenish its energy supply may adopt the goal only if it is motivated to do so, i.e. when its hunger motivation is active. At other times it may not consider this goal important and may reject it. Motivations thus provide reasons *why* an agent *generates* a goal, or why it *adopts* a goal that has been suggested by another agent.

Within SMART, motivation is represented simply as a given set.

$$[\text{SMART}\mathit{Motivation}]$$

3.5.1 Motivation for Pre-Negotiation

SMART provides us with a notional conception of motivation and its role in goal generation. Our need for motivation lies in its potential for providing us with a way to assign value or *worth* to an agent's goals. This enables negotiation to be considered in the context of already generated goals so that we can approach negotiation in a manner sensitive to ongoing activities. However, the abstract representation of motivation presented in SMART is limited (by design), so we must elaborate the motivation model substantially.

In particular, in this thesis, we develop and extend the existing abstract model of motivation given in SMART to enable us to reason about motivation and, more specifically, goals with motivational worth, in the context of pre-negotiation. We start by highlighting SMART'S notion that motivations can have more or less influence over goal generation depending on circumstances. For example, imagine a robot that normally explores its environment in an effort to construct a map, but must sometimes recharge its batteries. Here, motivations of *curiosity* and *hunger* might lead to the generation of specific goals at different times, with a *changing balance of importance* as time passes. Similarly, when undertaking a reasoning task, the nature and degree of reasoning that is possible must be determined by the need: in the face of a critical medical emergency, a coarse but rapid response may be best; in experimental trials, repeatability and accuracy are needed, often regardless of the time taken. In all of these cases, motivation is seen to dynamically respond to the needs of the situation, with motivations becoming more important at some points in time and less important at others.

3.5.2 Motivational Intensity

The importance of a motivation to an agent is determined by its strength or *intensity*. Motivations with high intensity have more influence over the generation of goals than those with lower intensities. Furthermore, a motivation can be considered *active* or *inactive* depending on whether the intensity of the motivation exceeds a given *threshold*. The combination of both intensity and threshold provides a way to control the amount of influence a motivation has on the agent's goal generation process. Both are defined as numbers in the positive naturals.

$$\mathit{Intensity} == \mathbb{N}_1$$

$$\mathit{Threshold} == \mathbb{N}_1$$

3.5.3 Motivational Cues

Levels of intensity change over time in response to the occurrence of events that impact upon an agent's ability to carry out its role. For example, a hunger motivation increases in intensity when the energy of an agent falls to a low level, or when the opportunity to obtain cheap energy presents itself. In general, certain situations arise when it becomes necessary or prudent to *generate* goals. In order to determine when these events occur, an agent must be able to represent knowledge about its environment. SMART begins this process by providing agents with a *view* containing an agent's current percepts of its environment. However, the view can only represent things the agent is currently perceiving, but many things that are important to an agent may not be currently perceivable. For this reason, we must also define the way an agent can store information across time. This is traditionally done using the concept of *belief*.

Beliefs are the internal representation of the environment, and can be positive or negative. For example, it is possible to hold the belief that it is raining and, similarly, it is possible to hold the belief that it is *not* raining. To represent the truth value of such statements we give the type, *Literal*, which is defined to be either an attribute or the negation of an attribute.

$$\mathit{Literal} ::= \mathit{pos}\langle\langle \mathrm{SMART}\mathit{Attribute}\rangle\rangle \mid \mathit{neg}\langle\langle \mathrm{SMART}\mathit{Attribute}\rangle\rangle$$

Beliefs are then defined formally as literals.

$$\mathit{Belief} == \mathit{Literal}$$

Once an agent holds a belief that an event with relevance to its motivations has occurred, the intensity of the affected motivations change accordingly. In our model, we adopt the notion of *motivational cues*, which are simply beliefs which, when true, cause a change to motivational intensity.

Formally, a *motivational cue* is described in the schema, *MotivationalCue*, in which a cue contains a *belief* that either increases or decreases intensity by the amount given in *effect*.

MotivationalCue
$belief : Belief$
$effect : Intensity$

Now, in order to manage the generation of different types of goals, agents need more than one motivation, and consequently a way of distinguishing between different motivations is needed. This is done by providing each motivation with a unique identifier drawn from a set of unique motivation identifiers represented as the given set, *MotiveID*.

$[MotiveID]$

3.5.4 Motivation Definition

Having described all the basic components of motivation, we can now proceed to give a formal definition. Motivations are distinguished by unique *id*s taken from the set of all motivation identifiers. The influence of a motivation over the goal generation process is represented by its *intensity*, which only has an effect if it is above an intensity *threshold*. Finally, the intensity of a motivation is determined by a set of motivational *cues*.

Motivation
$id : MotiveID$
$intensity : Intensity$
$threshold : Threshold$
$cues : \mathbb{P}\, MotivationalCue$

Motivations generate goals when their intensity exceeds their thresholds. To determine which goals to generate, we must associate those motivations that are relevant to each goal. The relationship between motivations and goals is a many-to-one relationship, in that more than one motivation might cause the generation of any one goal. For example, I might travel to another country to satisfy a motivation for enjoyment, but I might instead do so to satisfy a motivation to flee from the police. In general, many motivations can be responsible for generating any particular goal. The link between motivations and goals is defined formally in the relationship, SMART*MotivationGoal*, which associates a set of motivations with a goal.

$$\text{SMART}\mathit{MotivationGoal} == \mathbb{P}\,\mathit{Motivation} \times \text{SMART}\mathit{Goal}$$

3.5.5 Generating Goals

Autonomy is a property that arises from the ability to determine one's own agenda through the generation of goals. SMART describes how this ability can find representation in an agent model through the inclusion of a *motivational* component that provides the reasons why any goal is adopted, as it is through satisfying goals that motivational needs can be addressed. For example, an agent capable of satisfying a goal to obtain energy will only attempt to achieve the goal if it has a motivational need for the energy. More concretely, the agent will only attempt to satisfy a goal to *eat* if it has the motivational need, i.e. if it is *hungry*. This is contrary to other views, in which *goals* are seen as the primary force behind action ([138, 137]). As described in [80], however, motivations represent the *drive* behind activity that pushes the agent into action.

A goal, therefore, is generated only if there is a motivational need for it. In order for a goal to be generated, there must be at least one motivation associated with it through the SMART*MotivationGoal* relationship that is *active*, where active motivations are defined as those that have intensity levels exceeding their intensity threshold. Thus, only those goals that are associated with active motivations are generated.

The set of generated goals is identified by the operation schema *GenerateGoals*. (For simplicity, and to avoid introducing unnecessary machinery, we have not provided an agent's state schema; we concentrate therefore only on the salient aspects, giving a partial specification.) The predicate part of the schema describes the conditions that must hold for a goal to be in the set of *generated goals*. It states that for this to occur, the goal must be associated with a motivation that has an intensity greater than its threshold.

GenerateGoals

$$
\begin{array}{|l}
\hline
\text{SMART}goals : \mathbb{P}\,\text{SMART}goal \\
\hline
\text{SMART}goals' = \text{SMART}goals \cup \{g : \text{SMART}Goal \mid g \in \text{SMART}goals \bullet \\
\qquad (\exists m : Motivation;\ mg : MotivationGoal \mid \land \\
\qquad m \in \operatorname{dom}(mg \rhd \{g\}) \land m.intensity \geq m.threshold) \bullet g\} \\
\hline
\end{array}
$$

3.5.6 Goal Worth

As described above, any goal that has been generated is associated with at least one motivation that has an intensity greater than its threshold. The higher the intensity of a motivation, the more important is that motivation, and the more influence it has on activity. Now, in situations where we must make choices about which goal to adopt, an obvious way to proceed is to select those goals for which there is a greater motivational need. Thus, goals associated to motivations with higher intensities should be selected as intentions over those that have been generated by less intense motivations.

In order to do this, we associate the intensity of motivations to the goals they generate, where the intensity determines the goal's *worth*. Then, generated goals with higher worth can preferentially be selected as intentions over those with lower worth. Assigning worth to goals can be achieved in a number of different ways. We will not elaborate a discussion of this, since it is not the aim of our work to consider such alternatives. The important point is that distinctions can be made between goals, and a simple approach of using the motivation with the highest intensity to assign worth to a subsequently generated goal is all that is required for our purposes in this thesis. However, this carries the possible consequence that a goal generated by multiple motivations, each with individually low intensities, might be considered to have lower worth than a goal generated by one motivation with high intensity, even if the sum of the intensities of the motivations generating the former goal is larger than the latter. Since this has no impact for the development of our model, we can ignore such eventualities.

Worth is simply a number in the positive naturals:

$$Worth == \mathbb{N}_1$$

For a goal, worth is defined by the *goalworth* function, which takes a goal and a motivation-goal association as arguments, and returns the goal's worth.

$$goalworth : \mathrm{SMART}Goal \rightarrow MotivationGoal \rightarrow Worth$$

$$\begin{array}{l}
\forall g : \mathrm{SMART}Goal;\ mg : MotivationGoal \mid g = second\ mg \land \\
\quad (\exists m_1 : Motivation \mid m_1 \in first\ mg \land \\
\quad (\forall m : Motivation \mid m \in first\ mg \land \\
\quad m \neq m_1 \bullet goalworth\ g\ mg = m_1.intensity \Leftrightarrow \\
\quad m_1.intensity > m.intensity))
\end{array}$$

3.6 Conclusion

This chapter provides the foundational concepts upon which the rest of our model is built, which are largely borrowed from the SMART agent framework. For the work in this thesis, SMART provides the particularly important concept of motivation, which allows the worth of goals to be considered and, thus, enables comparisons between goals to be made on the basis of their respective worths and deliberations to be made regarding which goals to pursue and which to drop.

Although we lean heavily on SMART in this chapter, we make important extensions to its model of motivation. SMART provides a purposely general and abstract view of motivation that requires elaboration and grounding before it can be used in implemented agent systems and, in this chapter, we provide a more detailed model that describes more specifically how worth can be assigned to goals. This elaboration allows us to show later how the effects of different negotiation settlements on existing goals can be determined.

Other aspects relating to motivation are not covered, since they are not relevant to our concerns, such as the updating of motivational intensities, and the selection of goals as intentions. While these are important features of a motivational model for computational agents, we do not consider them here, but other work has addressed these issues and can be found elsewhere [46, 108].

Chapter 4

A Model of Negotiation Goals

4.1 Introduction

At the heart of any negotiation is the *negotiation objective*, which is some object, service or task for which an agreement is sought. In agent negotiation frameworks, the object of the negotiation is commonly represented as a list of items describing its main features. For example, a negotiation over the purchase of an airplane ticket would include a list of defining characteristics such as the destination and the date of travel. Taken together, these features describe the overall goal or *objective* of the negotiation, and are not normally themselves negotiable.

In general, different kinds of negotiation objective can be taken to be *goals* of some kind that require satisfaction. Negotiation over the satisfaction of such goals presumes that there are other aspects of these goals that *are* amenable to negotiation (normally called the *issues* of negotiation), in that they may be the subject of *compromise* between the parties involved. However, traditional representations of goals (in artificial intelligence and computer science) as fixed descriptions of desired states of affairs do not lend themselves readily to the process of negotiation. In many existing negotiation frameworks (e.g. [32, 68]), therefore, such goals are *made* negotiable by *appending* a set of negotiation issues to them. These issues then form the focus of negotiation, usually relating to properties of the goal itself, such as price, quality, and so on.

Yet, just as the majority of work on goal satisfaction and planning, for example, has omitted a consideration of the generation or origination of goals, so too has research on negotiation largely omitted consideration of the determination of negotiation issues. In addition to this omission, the construction of *preferences* over these issues is also not often considered in this context. Indeed, issues are generally presented as *given* components of the negotiation goal, and preferences over how these issues are to be settled are assumed to come either from a human user ([82]) or from predetermined measures of utility placed on the different ways in which the issues can be settled (e.g. [68]). In contrast, our aim is to enable the internal, dynamic generation of such preferences, thus allowing autonomous negotiation by agents in domains where user guidance is limited or undesirable. This provides a more flexible way to deal with dynamically changing environments so that negotiations can be conducted in a way that reflects the current situation of agents, improving the chances of obtaining settlements that meet these current needs.

Preferences are important, since they guide the compromises that are made on negotiation issues, moving away from the most strongly preferred settlement, or the *aspiration settlement*, towards increasingly less preferred settlements, where the limit on what a negotiator finds minimally satisfactory represents his *reservation settlement*, and is the point at which no further compromises can be made. Thus, a buyer hoping to purchase some product may have an aspiration price of $10 but, if necessary, may compromise and increase the amount he is willing to pay until he has reached his reservation price, when he will make no further compromises. The key idea here is that, as negotiation proceeds, negotiators traverse a space of preferences, either until an agreement is found, or until the reservation settlement is reached.

Taken together, the questions facing an agent when making preparations for a negotiation include understanding how the issues of a negotiation are to be determined, which parts of a negotiation goal should be negotiable, and how preferences over alternative settlements are to be identified. These questions form the problems that are addressed in this chapter, which has the following structure. In the next section we describe the goal model, and in Section 4.3 we present our model of issues. In Section 4.4 we describe how preferences over issues can be determined, and in Section 4.5 we offer some concluding remarks.

4.2 The Goal Model

In Chapter 3, we described how SMART defines goals to be sets of *attributes* describing desired states of affairs. However, for situations in which an agent must negotiate over the satisfaction of a goal, we need a more flexible representation, since such attributes do not provide enough information to allow reasoning over alternative instantiations of a goal. For example, a SMART attribute can describe the fact that a table is red, but not

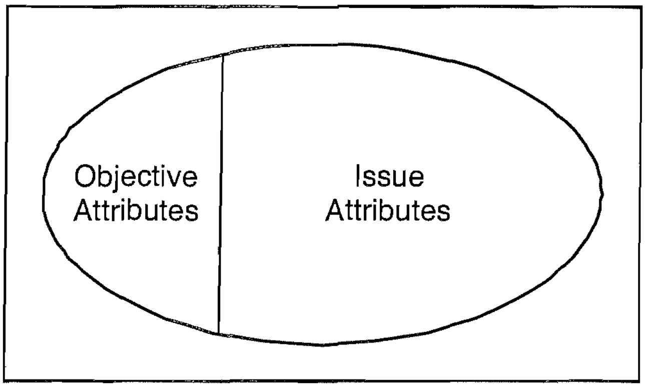


FIGURE 4.1: The goal model

that a table may be one of several different colours, or that the colour of the table is to be determined through agreement. Thus, for negotiation, we require that attributes are able to represent a space of *alternatives* over which an agent's preferences can range. Such attributes can then naturally represent negotiation issues.

As briefly discussed above, the negotiation objective is often represented by a list of non-negotiable characteristics, or attributes, much like any traditional symbolic goal representation. However, for goals that are to be satisfied through negotiation, a further set of attributes must be included. It is these attributes that describe the issues around which negotiation will proceed, with common examples being *price*, *time of delivery* and *quality*. Thus, goals for negotiation must contain two qualitatively different types of attributes: those that represent the objective of the negotiation, and those that are candidates for negotiation as issues. Such goals are in essence partially defined, or *partial goals*, since some of these attributes are as yet *uninstantiated* and may only become instantiated through negotiation. The inclusion of such *potential* issues allows an agent to tailor negotiation to its current circumstances by determining what should be negotiated, rather than relying on external input or a predetermined approach to identifying issues. This kind of representation enables a greater degree of flexibility to cope with dynamic domains in which negotiations may not be predetermined, and allows for the satisfaction of goals through dynamic determination of issues.

In order to make this clear, consider Figure 4.1, which shows a Venn diagram representing a goal as a set of attributes. A part of this set is composed of *objective attributes*, which define the main objective and are non-negotiable. The remaining attributes are *issue attributes*, which may be selected as negotiation issues and are, therefore, in contrast to objective attributes, potentially negotiable.

TABLE 4.1: An example goal

Attribute Name	**Objective**	**Issue**
`destination`	✓	
`date`	✓	
`seat_class`	✓	
`price`		✓
`departure_airport`		✓
`direct_flight`		✓

TABLE 4.2: A partially instantiated goal

Objective	
`destination`	`new_york`
`date`	`jan_12_2005`
`seat_class`	`business`
Issue	
`price`	`?`
`departure_airport`	`?`
`direct_flight`	`?`

Our approach thus involves redefining goals to include two sets of attributes, the former of which comprise the fixed objective of the goal, and the latter comprising a potential set of negotiation issues. Initially, therefore, a negotiation goal comprises a set of *objective attributes* for describing what must be achieved, and a set of *issue attributes* that may or may not be negotiated over. Table 4.1 shows the initial structure of an example of such a goal in which the *objective* is to travel to a destination on a specific date with a requirement on the class of travel (for example, business class). The price that is to be paid, the departure airport, and whether or not the flight is direct, are all *issue* attributes. Initially, the goal consists of just these types of attributes (i.e. objective and issue).

A partial instantiation of such a goal is shown in Table 4.2, where the objective is to fly to *New York* on the 12^{th} *of January 2005* in a *business class* seat. The *issue attributes* of the goal are as yet uninstantiated, so the values are represented by `?`. These issues gain their meaning in the context of the objective of the goal (i.e. the set of objective attributes) so, for example, the issue of *price* can be understood in the context of buying a business class flight ticket to New York on the date specified.

Based on these intuitions, we can build a formal model of the goals we require for our pre-negotiation model.

4.2.1 Symbolic Goals

As our aim is to build upon and develop deliberative architectures, we begin by describing some primitive concepts used by them in order to ground our model. Deliberative architectures take a *symbolic* perspective to representing information, which is described *declaratively* in *predicate* form using *constants* to represent known objects of the world and *variables* to represent unknown objects. For example, the price to pay for some service can be represented by a predicate symbol *price* and a constant representing some monetary resource, say 40, which means that the service has a price of $40. In first order predicate logic this would take the following form: `price(40)`. If the price is *unknown*, then we simply replace the constant with a variable, i.e. `price(X)`.

The constants and variables of a predicate are *terms*, represented as two given sets, *Const* and *Var* which, respectively, represent all *constants* and *variables*. A term is then simply defined to be either a *constant* or a *variable*.

$$[Const, Var]$$
$$Term ::= constant\langle\langle Const\rangle\rangle \mid variable\langle\langle Var\rangle\rangle$$

In general, predicates can contain any number of terms. Thus, for example, the symbolic attribute describing the state in which a box is on a table contains two terms, one representing the table and one representing the box, such as `on(table, box)`. However, since issue attributes are always associated with a set of objective attributes, they normally only contain one term (for example, `departure_airport(heathrow)`) so, when we come to define attributes for negotiation, we limit the number of terms of issue attributes to just one.

In order to describe issues we must be able to label them, for which we use *predicate symbols* represented by the given set, *PredSym*. Combining a predicate symbol with terms forms an *atom*; the general case allows for any number of ordered terms, so we must define an atom to contain a possibly empty *sequence* of terms.

$$[PredSym]$$

Atom

predicate : *PredSym*
terms : seq *Term*

TABLE 4.3: Symbolic attributes

Predicate symbol	**Terms**
`destination`	`new_york`
`date`	`jan_12_05`
`seat_class`	`business`
`departure_airport`	`X`
`price`	`Y`
`direct_flight`	`Z`

Thus, atoms can be used to describe features of the environment, and perform the same role as SMART attributes. To this elaboration of a SMART attribute we give the name *SymAttribute*, to capture the fact that we are modelling *symbolic descriptions*.

$$SymAttribute == Atom$$

Using the example goal from Table 4.2, we can reformulate the intuitive description of the example goal's attributes using our formal language. Both *objective* attributes and *issue* attributes can now be represented as predicate symbols and associated terms, as shown in Table 4.3, where the upper case letter, X, Y and Z represent variable terms.

Recall that objective attributes are those that have been given a unique value, so they must contain no variable terms. Formally, this is described below in the *ObjectiveAttribute* schema. The predicate part of the schema states that all of the terms in the attribute are constants.

$$\begin{array}{|l}
\multicolumn{1}{l}{\textit{ObjectiveAttribute}} \\
\hline
\textit{SymAttribute} \\
\hline
\forall t : Term \mid t \in (\operatorname{ran} sym.terms) \bullet t \in (\operatorname{ran} constant) \\
\hline
\end{array}$$

In our example of buying a flight ticket, all those attributes describing the objective of the goal are ground and represented by *objective attributes*. Table 4.4 extracts these attributes from the complete set defined in Table 4.3, and thus describes a traditional symbolic goal formed of ground attributes. Such goals are formally defined below as a set of *ObjectiveAttributes*.

$$SymGoal == \mathbb{P}_1\ ObjectiveAttribute$$

TABLE 4.4: Objective attributes

Predicate symbol	**Terms**
`destination`	`new_york`
`date`	`jan_12_05`
`seat_class`	`business`

TABLE 4.5: Three issue attributes and their values

Issue Attribute	**Values**				
`price`	90	100	110	120	130
`departure_airport`	`luton`	`heathrow`	`gatwick`	`stansted`	
`direct_flight`	`yes`	`no`			

4.2.2 Goals and Issues

As discussed earlier, the above representation of attributes is insufficiently expressive to allow us to reason about negotiation issues, in which *alternative* settlements must be considered. Thus, we develop our model by addressing such limitations, and build a representation of both negotiation goals and issues.

In existing negotiation frameworks, issues are typically represented as issue-value pairs, where an issue is represented by a label, such as *price*, and the value is some *constant* taken from an appropriate set of constants. For example, *price* = *40* represents the issue of price, currently with the value of 40, which is drawn from the set of non-negative integers representing monetary units such as dollars. Table 4.5 shows three issue attributes, `price`, `departure_airport` and `direct_flight`, along with a set of values for each, from which issue-values can be drawn to settle the issue. At the beginning of a negotiation, however, such issues will not yet have been settled so, in addition to representing issues, we must also represent the fact that they are not yet settled (and thus contain variables).

Issue attributes must contain *one* term which, since it has not yet been settled, is a variable term. The set of possible settlements of this variable term are drawn from an associated set of issue-values, represented by constants; in order to remain neutral over the various different types of constants that are possible, we define them to be a synonym, *IssValue*, of type *Const*.

IssValue == *Const*

We define issue attributes formally in the *IssueAttribute* schema, which includes a *SymAttribute* variable, a *range* of issue-value constants that can be used to replace the variable term, and an *ordering* over the issue-value range representing the *preferences* expressed over the range of issue-values. The predicate states that *range* must have the same elements as *order*, and that the number of terms is limited to one.

IssueAttribute

$sym : SymAttribute$

$range : \mathbb{P}\, IssValue$

$order : \operatorname{seq} IssValue$

$range = \operatorname{ran}(order)$

$\# sym.terms = 1$

Note that, in the *ordering* imposed on issue-values, two contiguous elements might be equally preferred (that is, we cannot make a judgement between them). We will consider this more fully when we come to *quantify* preferences later in this chapter.

Now, when an issue attribute has been instantiated with an issue-value from its range, we say that the attribute is ground. This is similar to the notion of an objective attribute in that the variable term in the issue attribute is replaced by a constant term. The definition of such an attribute is simply an *IssueAttribute* with an extra predicate, as in the *ObjectiveAttribute* schema.

SimpleGroundAttribute

$IssueAttribute$

$\forall t : Term \mid t \in (\operatorname{ran} sym.terms) \bullet t \in (\operatorname{ran} constant)$

In fact, such an attribute represents a *settlement* of an issue attribute, which allows us to provide an alias for it as a *Settlement*.

$$Settlement == SimpleGroundAttribute$$

We are now able to define the kinds of goals that satisfy our requirement for including issues. Recall that a goal includes a set of objective attributes, and a set of issue attributes from which negotiation issues can be drawn. Formally, this is described in the *Goal* schema, which states that a goal must contain a non-empty set of objective attributes (which can be simple or otherwise) and may also contain some issue attributes.

Goal

$objectiveattributes : \mathbb{P}_1 \; ObjectiveAttribute$

$issueattributes : \mathbb{P} \; IssueAttribute$

We can use this model to represent both traditional symbolic goals (in which case *issueattributes* is an empty set), and goals that may be negotiated (in which case *issueattributes* is a non-empty set from which negotiation issues can be determined).

4.3 The Issue Model

On generation of a goal, the task is to determine the need for negotiation over the different issue attributes of the goal, with two possible outcomes: an issue attribute can either be selected for, or excluded from, negotiation. Issues for negotiation are classified as *negotiable issues* and can only arise if there are alternative issue-values that are preferred as settlements. The exclusion of an issue from negotiation can arise for two reasons. First, there may be *only one* issue-value that is acceptable, in which case they are identical to those that form the fixed, objective part of the goal, and we therefore classify them as *fixed issues*. Second, issues may be excluded from negotiation if we are indifferent to *all* possible issue-values, in which case they are classified as *slack issues*. In this sense, the issues have no importance and can be dropped. We assume here that issues that are not rejected for negotiation, due to either of these two reasons, must be negotiated over. However, in Chapter 6 we show how this assumption can be relaxed so that an agent can choose not to negotiate over an issue in response to information gained about the negotiation context.

In this respect, the three classes of issues (i.e. fixed, slack and negotiable) are solely determined in relation to preferences over issue-value settlements, and are thus generated using endogenous criteria (i.e. the existing goals of the agent, as we will see in Section 4.4.1), thus ensuring their compatibility with the agent's current activities and concerns.

We can illustrate this if we consider again the example goal from Table 4.2, but now with the issues classified into one of the three issue types, as in Table 4.6. The *departure airport* issue now has a uniquely preferred settlement, *heathrow*, and is thus classified as a fixed part of the objective (separated from the existing objective attributes by a horizontal line to represent its different origin), so will not be negotiated over. *Price* has been classified as a negotiable issue indicating that the agent will consider different prices (i.e. it is willing to negotiate over price), shown by the sequence of prices, where

TABLE 4.6: An example negotiation goal

Objective	**Preferred Issue-value(s)**
destination	new_york
date	jan_12_05
seat_class	business
departure_airport	heathrow
Negotiable	
price	⟨90,95,100⟩
Slack	
direct_flight	{yes, no}

the order represents varying degrees of preference from the least preferred on the left to the most preferred on the right. This is the only negotiable issue, since the *direct flight* issue has been classified as slack, indicating that there is no preference over the issue-values, shown by the set {yes, no}, which includes all possible values.

To summarise, we have described three different classes of issue attributes of a goal to reflect the different status of issues in a negotiation. These three classes of issue are informally defined below.

- Issues that are non-negotiable, or *fixed issues*, are those for which there is only one acceptable issue-value.
- Issues towards which the agent is indifferent, or *slack issues*, are those for which no distinction can be made between different issue-values in terms of preference.
- Issues that are selected for negotiation, or *negotiable issues*, are those for which there are several acceptable issue-values.

In terms of our goal model, the classification achieves a partitioning of the issue attributes into one of the three types of issue defined above. This is shown graphically in Figure 4.2, in which the issue attributes defined in the goal are now classified into either fixed, slack or negotiable issues (while objective attributes remain unchanged but now include the set of fixed issues, separated by a dashed line for clarity).

Based on these principles, we can construct a model of negotiation goals that captures the relationships between goals, issues and preferences, to facilitate decision-making in pre-negotiation.

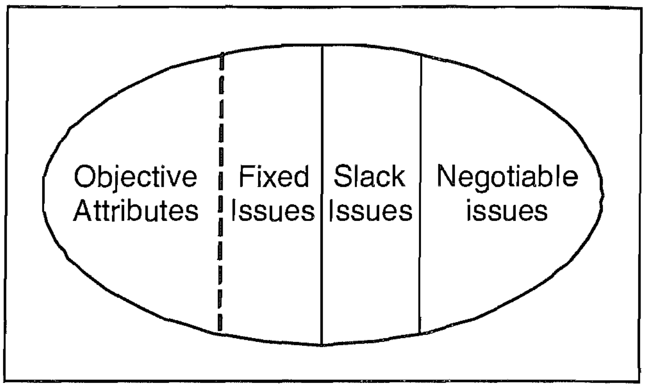


FIGURE 4.2: A schematic of a negotiation goal

4.3.1 Preference Structure and the Classification of Issues

The important point to note about the issue classification above is that it relies fundamentally on *preferences*. However, we have not yet made clear what we mean by preference. A preference is a relation between two issue-values, indicating the relative desirability of each. So, for example, if $\{a, b\}$ represents a set of two issue-values, and if a is preferred to b, this is represented as $\langle b, a \rangle$ with the angled brackets representing an order. Here, issue-values to the right are preferred more than issue-values to the left, and we have a left to right ascending order. The rightmost issue-value is called the *aspiration* settlement, representing the most desirable settlement the agent can achieve. Assuming there is more than one possible value in the set, then at some point to the left of the aspiration settlement lies the *reservation* settlement, or the least acceptable settlement. Any issue-value further to the left of the reservation settlement in this sequence is considered *unacceptable* in the sense that the outcome is worse than the least acceptable value.

If only one issue-value in a set of values is acceptable, then it is both the aspiration *and* the reservation settlement, and all other issue-values are unacceptable. Thus, if $\{a, b, c\}$ is a set of three issue-values, and only a is acceptable, then b and c are unacceptable and a is both the reservation and aspiration settlement. In line with existing models of negotiation, agents try to secure their aspiration settlement, but are willing to concede to other issue-values up to and including their reservation settlement if necessary. In certain situations, however, all possible issue-values may have negative consequences and, in this case, the aspiration (and reservation) settlement is the issue-value that minimises the losses incurred.

Since preferences over issue-values are indicated by their *order*, from the the aspiration settlement to the reservation settlement, we need some means of establishing that order. We can do this by giving each issue-value a numeric *score*, determined in some appropriate fashion, to indicate the desirability of an outcome. Now, since we also want to represent acceptable and unacceptable settlements in relation to the reservation value, we require a scoring mechanism that can distinguish between these different classes. This is easily achieved in the general case in an intuitive fashion by adopting a scoring function that gives negative values for all those less desirable than the reservation settlement, and non-negative values for the reservation settlement and those more desirable. This is sensible, since the reservation settlement will be that which offers the minimal benefit. However, in the special case where there are no positively scored settlements (since none give benefit in themselves) but the issue is still identified as relevant to the agent, the reservation settlement is taken to be that with the highest negative score (giving the least loss). In summary, the reservation settlement is defined as:

- the settlement with the lowest positive score; or
- the settlement with the highest negative score if all other settlements are negatively scored (in which case it is also the aspiration settlement).

The scoring function outlined above also allows us to represent the case where issue-values are equally preferred and receive the same score.

The *range* of an issue attribute covers an unordered set of issue-values, from which individual issue-values can be drawn to transform the issue into a settlement. For example, the set of London airports contains individual airports, any of which can be used to settle the *departure airport* issue:

$$London\_airports = \{\texttt{luton, heathrow, gatwick, stansted}\}$$

An ordering on this set, represented as a sequence, indicates the preference expressed towards each airport in ascending order so that, for example, *gatwick* is the least preferred (the reservation settlement) and *heathrow* the most preferred (the aspiration settlement):

$$Airport\_order = \langle\texttt{gatwick, luton, stansted, heathrow}\rangle$$

However, to represent the case where two or more issue-values are equally preferred we require a *sequence of sets*, where each set in the sequence contains equally scored issue-values. The sequence below partitions the original order into sets of equally scored

TABLE 4.7: Slack issue

Issue-value sets	**Polarity**
{gatwick, luton, stansted, heathrow}	+
∅	-

Issue-value sets	**Polarity**
∅	+
{gatwick, luton, stansted, heathrow}	-

TABLE 4.8: Fixed issue

Issue-value sets	**Polarity**
{heathrow}	+
{gatwick, luton, stansted}	-

Issue-value sets	**Polarity**
{heathrow}	$-_{highest}$
{gatwick, luton, stansted}	-

issue-values, and indicates that *heathrow* attracts a higher score than *stansted* which, in turn, attracts a higher score than *gatwick* and *luton*, both of which are equally scored.

$$Airport\_order_{partitioned} = \langle\{\texttt{gatwick, luton}\}, \{\texttt{stansted}\}, \{\texttt{heathrow}\}\rangle$$

We can use this representation to help classify issues as described above. First, slack issues are those for which all possible issue-values are equally scored. This is shown in Table 4.7, in which all airports are contained in the same set and are either positively scored or negatively scored.

On the other hand, fixed issues are those for which only one issue-value is acceptable and all others are unacceptable. This leads to the case in which the partitioned order contains two sets. The first contains all those issue-values that are unacceptable, and the second contains the single issue-value that is acceptable. Thus, in our example, *heathrow* is the only issue-value that is acceptable, and thus represents both the aspiration and reservation settlement for this issue, and the remaining airports are all unacceptable. This is shown in Table 4.8, in which two cases are shown: the aspiration settlement is either the only airport that is positively scored (in the top part of the table) or it is the airport that has the highest negative score out of the complete set of negatively scored airports (in the lower part of the table).

There is, additionally, a special case of a fixed issue when there are two sets: one containing positively scored issue-values and one containing negatively scored issue-values. In this case, the preference is for any issue-value in the positively scored set.

TABLE 4.9: Negotiable issue

Issue-value sets	Polarity
`{luton},{stansted},{heathrow}`	+
`{gatwick}`	-

This is a form of fixed issue, but one in which we have a *set* of aspiration and reservation settlements, rather than just one settlement. However, for reasons of clarity of exposition, we do not consider this form of issue any further, since it can be reduced to the existing categories.

Finally, negotiable issues are those for which there are several issue-values attracting different positive scores. In such cases, the sequence contains sets of issue-values whose scores increase as we move from left to right. This captures the property of *concession*, since, as we move from the aspiration to the reservation settlement, each set represents increasingly less preferred issue-values. Table 4.9 shows such an issue, for which `luton`, `stansted`, `heathrow` are the acceptable settlements and `gatwick` is unacceptable.

4.3.2 Preliminary Notions

Having introduced our ideas, we are now able to present the formal model of issue types and the characteristics that identify them. To proceed, we present several auxiliary functions that are necessary for what follows.

In order to classify the different types of issue in negotiation (i.e. fixed, slack and negotiable), we define six auxiliary functions in order to examine the structural properties of the preference ordering placed on an issue's possible settlements. The first four functions are defined in the following generic schema definition. The function, *head*, returns the first element of a sequence, while *tail* returns a sequence of elements minus the head of the sequence. The function, *last*, returns the last element in a sequence, and the function, *front*, returns all but the last element of a sequence.

$$\begin{array}{|l}
[X] \\
\hline
head : \mathrm{seq}\,X \rightarrow X \\
tail : \mathrm{seq}\,X \rightarrow \mathrm{seq}\,X \\
last : \mathrm{seq}\,X \rightarrow X \\
front : \mathrm{seq}\,X \rightarrow \mathrm{seq}\,X \\
\hline
\forall x : X;\ s : \mathrm{seq}\,X \bullet \\
\qquad head \langle x \rangle ^\frown s = x \land \\
\qquad tail \langle x \rangle ^\frown s = s \land \\
\qquad last\ s ^\frown \langle x \rangle = x \land \\
\qquad front\ s ^\frown \langle x \rangle = s \\
\hline
\end{array}$$

Lastly, we define the functions, *elementscore* and *partition*. The *elementscore* function provides a numerical score associated with an element (representing the degree of preference for an issue-value), and *partition* converts a sequence of elements into a sequence of sets of equally scored issue-values, as described above.

$$\begin{array}{|l}
[X] \\
\hline
elementscore : X \rightarrow \mathbb{N} \\
partition : \mathrm{seq}\,X \rightarrow \mathrm{seq}(\mathbb{P}\,X) \\
\hline
\end{array}$$

These functions are required so that we can examine the structure of the *order* imposed on a set of issue-values determined by an agent's preferences that will then allow us to make issue classifications.

4.3.3 Issue Classification

We now proceed to examine formally the different classes of issues that we informally introduced above. As discussed, each class of issue is characterised by the structure of the preference ordering imposed on the different ways in which the issue can be settled.

First, slack issues are defined in the *SlackIssue* schema, which extends the definition of *IssueAttribute* by stating that the tail of the *partitioned order* is empty, and that the quantity of issue-values in the head of the partitioned order is equal to the quantity in the range.

SlackIssue

IssueAttribute

$tail(partition(order)) = \varnothing$

$\#head(partition(order)) = \#range$

Fixed issues are those for which there is *only one* acceptable issue-value, as formally defined in the *FixedIssue* schema, which again extends *IssueAttribute*. In this case, the predicate states the two criteria that define fixed issues. Either there is one issue-value in the *last* set of the sequence returned by *partition* that must have an *elementscore* greater-than or equal to 0, and the issue-values in the *last* set of the remainder of the sequence score less than 0 or, all issue-values score negatively, in which case the one with the highest negative score is contained in the last set, and is therefore the only acceptable issue-value. The predicate ensures that fixed issues have only one acceptable issue-value, which is the one in the last set of the partitioned order.

FixedIssue

IssueAttribute

$\#last(partition(order)) = 1$

$elementscore(last\ order) \geq 0$

$elementscore(last(front\ order)) < 0 \vee$

$\#last(partition(order)) = 1$

$elementscore(last\ order) < 0$

$elementscore(last(front\ order)) < elementscore(last\ order)$

Finally, *negotiable* issues are defined by the following condition: there must be at least two issue-values with *different* positive scores, so that at least one of the other issue-values in the *front* of the order must also be positively preferred. (In fact we only need to ensure that the *last* of these is positively preferred.)

Formally, this is shown in the *NegotiableIssue* schema. The predicate states that the *last* issue-value is positively valued (i.e. *elementscore* returns a value greater than 0) and that, for the *last* issue-value in the *front* of *order*, its *elementscore* must similarly be greater than 0. We also define the *activerange*, which is that part of the *range* containing *acceptable* issue-values (i.e. excluding unacceptable ones) and, similarly, the *activeorder*, which is simply the preference ordering placed on the *activerange*, in which the last element represents the aspiration settlement and the first element represents the reservation settlement.

$NegotiableIssue$

$IssueAttribute$

$activeorder : \mathrm{seq}\, IssValue$

$activerange : \mathbb{P}\, IssValue$

$$\#tail(partition(order)) > 1 \land elementscore(last\ order) > 0$$
$$\land\ elementscore(last(front\ order)) > 0$$

4.3.4 Negotiation Goals

A negotiation goal forms the initial guiding structure around which a negotiation proceeds. It should define which attributes are not to be negotiated, which are open for negotiation, and which have no consequence, and can therefore be discarded. These three constituents represent the *fixed*, *negotiable* and *slack* issues defined above.

We can now build a formal definition of negotiation goals by extending the definition provided in Section 4.2.2 for goals. Recall that a goal contains a set of objective attributes that describe the central aim of the goal, and a set of issue attributes from which the sets of fixed, negotiable and slack issues are determined. Negotiation goals also include a set of *negotiable issues*, a set of *slack issues* and a set of *fixed issues*. Finally, we define three axiomatic functions, *fixedtoissue*, *negtoissue* and *slacktoissue*, which strip each of the issue types of the extra information they contain and return issue attributes to show that they are all subsets of the issue attributes of the negotiation goal, and are mutually exclusive (in that their intersection is the empty set).

$$fixedtoissue : \mathbb{P}\, FixedIssue \rightarrow \mathbb{P}\, IssueAttribute$$
$$negtoissue : \mathbb{P}\, NegotiableIssue \rightarrow \mathbb{P}\, IssueAttribute$$
$$slacktoissue : \mathbb{P}\, SlackIssue \rightarrow \mathbb{P}\, IssueAttribute$$

TABLE 4.10: The final structure of a negotiation goal

Attribute Name	Objective	Issue	Fixed	Negotiable	Slack
`destination`	✓				
`date`	✓				
`seat_class`	✓				
`price`				✓	
`departure_airport`			✓		
`direct_flight`					✓

NegotiationGoal

$$\begin{array}{l}
g : Goal \\
negotiableissues : \mathbb{P}\, NegotiableIssue \\
slackissues : \mathbb{P}\, SlackIssue \\
fixedissues : \mathbb{P}\, FixedIssue
\end{array}$$

$$\begin{array}{l}
negtoissue(negotiableissues) \subseteq g.issueattributes \\
slacktoissue(slackissues) \subseteq g.issueattributes \\
fixedtoissue(fixedissues) \subseteq g.issueattributes \\
negtoissue(negotiableissues) \cap slacktoissue(slackissues) = \varnothing \wedge \\
\quad slacktoissue(slackissues) \cap fixedtoissue(fixedissues) = \varnothing \wedge \\
\quad negtoissue(negotiableissues) \cap fixedtoissue(fixedissues) = \varnothing
\end{array}$$

The example goal shown in Table 4.10 shows the various attributes of the goal when the different status of each issue in the set of issue attributes has been determined. In the table, *price* has been classified as a negotiable issue, *direct flight* has been classified as a slack issue, and *departure airport* has been classified as fixed issue and therefore becomes part of the objective, along with the other three previously identified objective attributes of *destination*, *date* and *seat class*. Note also that the goal no longer contains any unclassified issue attributes, since these have now all been transformed into either objective, negotiation or slack issues.

4.4 Preference Determination

Given the above model of goals, issues and preferences, and based on the understanding that preferences determine the nature and classification of issues into fixed, slack and negotiable, we must still consider how preferences are established. Traditionally, preferences are determined externally by the designer or user, but this violates our aim of the

TABLE 4.11: Three goals

Predicate	**Terms**
`attend`	`london_meeting, jan_19_05`
`meet`	`bob, conference_dinner, jan_20_05`
`attend`	`conference, jan_21_05`

internal generation of such preferences. However, since we are concerned with negotiation in deliberative agents that possess explicit goals, we can exploit this representation of goals to determine preferences over different negotiation settlements. Specifically, we can examine the effects of potential settlements on existing goals (or intentions in the BDI view) and, by determining the nature of these effects, we can establish how preferable a given settlement is in comparison to others.

There are three forms of effect that settlements can have on existing goals. They can either:

- conflict with goals, making them impossible to satisfy;
- hinder the satisfaction of goals by conflicting with one or more of their subgoals; or
- facilitate the satisfaction of goals by satisfying one or more of their subgoals.

Settlements that *conflict* with or hinder the achievement of existing goals ought (from a rational perspective) to be less preferable than those that *facilitate* the achievement of such goals. This premise provides the basis on which we can develop preferences over different issue-value settlements. For example, by examining the effects of alternative settlements, we can derive the relative preference of each, and thus construct a preference-based ordering over all possible settlements. In this section, we explore and define these effects and show how they can be used to determine an agent's preferences.

To make the above intuitions more concrete, imagine that Alice has the following three goals, which are shown with their predicates and terms in Table 4.11:

1. attend a business meeting in London on the 19^{th} of January;
2. attend a conference in New York on 21^{st} of January; and
3. meet with a colleague at the pre-conference dinner on the 20^{th} of January.

Alice's goal to attend the conference involves satisfying a subgoal to buy an airplane ticket, which has an issue attribute that refers to the *date* of the flight, with a *range* consisting of the following possible settlement dates: `jan_19_05`, `jan_20_05` and `jan_21_05`. Since `jan_19_05` is the date of the meeting, the use of this as a settlement causes a conflict to arise between Alice's goal to fly on that date and her goal to attend the meeting. Flying on `jan_20_05` hinders Alice's goal of meeting her colleague at the pre-conference dinner, but still gets her to the conference in time. However, if Alice can arrange to meet her colleague at another event on another date, then flying on `jan_20_05` merely hinders the goal to meet her colleague, but no longer directly conflicts with it since she can make other arrangements to satisfy that goal. Finally, if Alice chooses to fly on `jan_21_05`, she will miss both the dinner and the first day of the conference. In this way, by examining the different possible settlements available we can determine the effects (both positive and negative) of each on existing goals.

4.4.1 The Effects of Settlements on Goals

To continue the presentation of our model, we now provide the formal definitions of the different effects that settlements can have on goals. Detecting the effect of a settlement on a goal is, in the general case, intractable, since it involves considering all possible *logical consequences* of the settlement. Although there are heuristic solutions in various branches of computer science, some of which might be applied here, the problem is more general than our work in this thesis, so we do not give a detailed analysis. The problem is ubiquitous in many areas of AI but can be dealt with in many practical instances by simple pattern matching approaches or resolution [126], and the best approach to take is often domain specific. For our purposes we adopt the common solution of abstracting out the problem (while recognising its importance) by defining two different kinds of effects. Specifically, we provide axiomatic definitions of situations in which a settlement leads, by logical consequence, to a state that has a *positive effect* on a goal, or to a state that has a *negative effect* on a goal. We then use these abstract concepts to construct the relationships by which a settlement *conflicts*, *hinders* and *facilitates* a goal. Note that the use of axiomatic definitions here and in other parts of the thesis represent simplifications, the instantiation of which are in many instances domain specific. In keeping with our general approach therefore, we omit a detailed analysis.

We define two relationships: the first, *positiveEffect*, represents those cases where a settlement positively affects the satisfaction of a goal; the second, *negativeEffect*, represents those cases where a settlement negatively affects the satisfaction of a goal.

$positiveEffect : Settlement \leftrightarrow NegotiationGoal$

$negativeEffect : Settlement \leftrightarrow NegotiationGoal$

Using these relationships we can now define the cases in which settlements conflict, hinder and facilitate goals. We say that a settlement facilitates a goal when the settlement leads, by logical consequence, to a state that is a subset of all those states defined in the *positiveEffect* relation.

$facilitates : Settlement \leftrightarrow NegotiationGoal$

$facilitates \subseteq positiveEffect$

A settlement *conflicts* with a goal if the settlement leads, by logical consequence, to a state that is a subset of those states defined in the *negativeEffect* relation.

$conflicts : Settlement \leftrightarrow NegotiationGoal$

$conflicts \subseteq negativeEffect$

To describe the *hinder* effect, we must first define the *subgoal* relation. A goal is a subgoal of another when the first contributes to the satisfaction of the second. Several properties of the subgoal relation are defined in [29], and we adopt them here. First, we simply define the relation that arises from one *goal* conflicting with another.

$goalconflict : NegotiationGoal \leftrightarrow NegotiationGoal$

The subgoal relation is *consistent*, so that for any pair of goals, g_1 and g_2, where g_2 is a subgoal of g_1, $(g_1, g_2) \notin goalconflict$. A goal is *reflexive*, so that it is its own subgoal, and the relation is *transitive*, so that if g_2 is a subgoal of g_1, and g_3 is a subgoal of g_2, then g_3 is also a subgoal of g_1. Finally, the relation is *well-founded* meaning that no goal has an infinite chain of subgoals.

$subgoal : \mathbb{P}(NegotiationGoal \times NegotiationGoal)$

$\forall g_1, g_2 : NegotiationGoal \bullet (g_1, g_2) \in subgoal \Rightarrow (g_1, g_2) \notin goalconflict$

$\forall g_1 : NegotiationGoal \bullet (g_1, g_1) \in subgoal$

$\forall g_1, g_2, g_3 : NegotiationGoal \bullet ((g_1, g_2) \in subgoal \wedge (g_2, g_3) \in subgoal) \Rightarrow (g_1, g_3) \in subgoal)$

$\forall g : NegotiationGoal \bullet (\#\{g_1 : NegotiationGoal \mid (g_1, g) \in subgoal\} \in \mathbb{F})$

We are now able to say that a settlement *hinders* a goal if it *conflicts* with any of the goal's subgoals. Note that the relation, *hinder*, does not entail that the hindered goal cannot be satisfied, since the agent may possess other ways to bring about its satisfaction and thereby overcome a subgoal conflict.

$$\begin{array}{|l}
\mathit{hinders} : \mathit{Settlement} \leftrightarrow \mathit{NegotiationGoal} \\
\hline
\forall s : \mathit{Settlement};\ g : \mathit{NegotiationGoal} \bullet \\
\quad \mathit{hinders}\ s\ g \Leftrightarrow \\
\quad (\exists g_1 : \mathit{NegotiationGoal} \bullet ((g_1, g) \in \mathit{subgoal} \land (s, g_1) \in \mathit{conflicts}))
\end{array}$$

Given the above relations, we can now classify goals that are either hindered, in conflict with, or facilitated by, a settlement. We do this by defining three functions, each of which takes a settlement and a set of goals as arguments, and return a set of goals. The first function, *hinderedgoals*, states in its predicate that the set of goals returned are those in the range of the *hinders* relation when it is domain restricted to the *Settlement* argument. The second function, *conflictedgoals*, states that the goals returned are those in the range of the *conflicts* relation, domain restricted to the *Settlement* argument. Finally, the function, *facilitatedgoals*, states that the goals returned are those in the range of the *facilitates* relation, again domain restricted to the *Settlement* argument.

$$\begin{array}{|l}
\mathit{hinderedgoals} : \mathit{Settlement} \to (\mathbb{P}\,\mathit{NegotiationGoal}) \to (\mathbb{P}\,\mathit{NegotiationGoal}) \\
\hline
\forall s : \mathit{Settlement};\ gs : \mathbb{P}\,\mathit{NegotiationGoal} \bullet \\
\quad \mathit{hinderedgoals}\ s\ gs = \\
\qquad \{g : gs \mid g \in \operatorname{ran}(s \lhd \mathit{hinders}) \bullet g\}
\end{array}$$

$$\begin{array}{|l}
\mathit{conflictedgoals} : \mathit{Settlement} \to (\mathbb{P}\,\mathit{NegotiationGoal}) \to (\mathbb{P}\,\mathit{NegotiationGoal}) \\
\hline
\forall s : \mathit{Settlement};\ gs : \mathbb{P}\,\mathit{NegotiationGoal} \bullet \\
\quad \mathit{conflictedgoals}\ s\ gs = \\
\qquad \{g : gs \mid g \in \operatorname{ran}(s \lhd \mathit{conflicts}) \bullet g\}
\end{array}$$

$$\begin{array}{|l}
\mathit{facilitatedgoals} : \mathit{Settlement} \to (\mathbb{P}\,\mathit{NegotiationGoal}) \to (\mathbb{P}\,\mathit{NegotiationGoal}) \\
\hline
\forall s : \mathit{Settlement};\ gs : \mathbb{P}\,\mathit{NegotiationGoal} \bullet \\
\quad \mathit{facilitatedgoals}\ s\ gs = \\
\qquad \{g : gs \mid g \in \operatorname{ran}(s \lhd \mathit{facilitates}) \bullet g\}
\end{array}$$

4.4.2 Goal Worth

With the new definition of goals given above, we must now revisit how goals are assigned worth. In Chapter 3, we described how the worth of a SMART goal is determined by the strength of the motivations responsible for its generation. Specifically, the worth of a goal is given by the motivation with the highest intensity. We maintain this approach but, to keep our model consistent, relate this to our *new* goal definition, as shown below where *MotivationGoal* defines the relationship between a set of motivations and the goal they can generate.

$$\mathit{MotivationGoal} == \mathbb{P}\,\mathit{Motivation} \times \mathit{Goal}$$

The worth of a goal is thus derived from the motivation associated to it with the highest intensity.

$$\mathit{goalworth} : \mathit{Goal} \rightarrow \mathit{MotivationGoal} \rightarrow \mathit{Worth}$$

$$\begin{array}{l}
\forall\, ng : \mathit{Goal};\ mg : \mathit{MotivationGoal} \mid ng = \mathit{second}\ mg\ \wedge \\
\quad (\exists\, m_1 : \mathit{Motivation} \mid m_1 \in \mathit{first}\ mg\ \wedge \\
\quad (\forall\, m : \mathit{Motivation} \mid m \in \mathit{first}\ mg\ \wedge \\
\quad m \neq m_1 \bullet \mathit{goalworth}\ ng\ mg = m_1.\mathit{intensity} \Leftrightarrow \\
\quad m_1.\mathit{intensity} > m.\mathit{intensity}))
\end{array}$$

Given this mechanism for establishing the worth of goals, we can do the same for negotiation goals. However, our model of negotiation goals contains *issues* that can change the worth of a goal (increasing or decreasing), depending on how they are settled. In order to determine the worth of such goals, therefore, we must include the potential worth of such issues.

Upon generation, the worth associated to a goal is assigned to the objective set of attributes as for regular goals above, but the different status of issues provides different amounts of additional worth. Issue attributes that are classified as *fixed* have a known worth that can be directly added to the objective worth of the goal, since their instantiation is known and, thus, their effects on the existing goals are also known. Similarly, slack issues are defined as issues for which any instantiation has the same worth, so the worth of a slack issue is also known. The same cannot be said, however, for negotiation issues whose instantiation is as yet unknown. However, in Chapter 6, Section 6.4.2, we describe how to determine the expected worth of such issues and so we can calculate (in an approximate manner) the worth they add to the goal. Thus, to determine the total worth of a negotiation goal, we take the different issues and add their associated worths to the overall worth of the goal. This is shown formally in the *NegGoalWorth* schema.

$$
\begin{array}{|l}
\textit{NegGoalWorth} \\
\hline
\textit{mots?} : \mathbb{P}\,\textit{Motivation} \\
\textit{ng} : \textit{NegotiationGoal} \\
\textit{objectiveworth} : \textit{Worth} \\
\textit{fixedworth} : \mathbb{P}\,\textit{FixedIssue} \rightarrow \textit{Worth} \\
\textit{slackworth} : \mathbb{P}\,\textit{SlackIssue} \rightarrow \textit{Worth} \\
\textit{expectedissueworth} : \mathbb{P}\,\textit{NegotiationIssue} \rightarrow \textit{Worth} \\
\textit{totalgoalworth} : \textit{Worth} \\
\hline
\textit{objectiveworth} = \textit{goalworth}\ \textit{ng.g}(\textit{mots?} \times \textit{ng.g}) \\
\operatorname{dom} \textit{fixedworth} = \textit{ng.fixedissues} \\
\operatorname{dom} \textit{slackworth} = \textit{ng.slackissues} \\
\operatorname{dom} \textit{expectedissueworth} = \textit{ng.negotiationissues} \\
\textit{totalgoalworth} = \textit{objectiveworth} + \\
\qquad \textit{fixedworth}(\textit{ng.fixedissues}) + \textit{slackworth}(\textit{ng.slackissues}) + \\
\qquad \textit{negworth}(\textit{ng.negotiationissues}) \\
\hline
\end{array}
$$

4.4.3 Evaluation of Issue Settlement Impact

Having determined the different effects of issue settlements on goals, we need now to provide a way of evaluating these effects so that we can build up the preference ordering over different possible settlements. To do this, we must examine the sets of hindered, conflicting and facilitated goals, and calculate a numerical value to use as the basis of a *score* for a given settlement. This allows us to compare the scores of different settlements and provide an ordering.

Recall that goals are generated and given worth by motivations, so we can use the worth associated to existing goals to obtain the scores given to each potential settlement depending on their effects on those goals. Using our example of a trip to New York, we can imagine that Alice's goal to attend the meeting in London is generated by her motivation to advance her career, and has a high level of associated worth. Selecting a date that conflicts with this goal should, therefore, receive a high *negative* score that reflects the worth of the conflicting goal. If a settlement of the *date* issue merely hinders the goal, the score it receives should be proportionally less than if it conflicts with it, to reflect the fact that the consequences, though negative, are less than if there is an outright

TABLE 4.12: Example settlement effects on three of Alice's goals

date settlements	**meeting**(10)			**colleague**(4)			**conference**(6)			Σ
	c	h	f	c	h	f	c	h	f	
`jan_19_05`	-1.0					0.4			0.6	0.0
`jan_20_05`			1.0		-0.2				0.6	1.4
`jan_21_05`			1.0		-0.2		-0.6			0.2

conflict. Finally, any date that facilitates the goal should receive a positive score based on its worth.

This example is illustrated in Table 4.12, in which the three possible settlements of the *date* issue are listed in the first column. The effects of each settlement on Alice's three goals of attending the London *meeting*, meeting a *colleague* at the pre-conference dinner and attending the *conference* in New York (each shown with an example worth in brackets) are indicated by the example scores in the table body. These scores are related to the worth of the goals affected and, thus, conflicting settlements are taken as the worth of the goal they effect multiplied by -0.1, hinder effects are given by the worth of the goal they effect multiplied by -0.05 (to reflect the proportionally smaller effects they have on goals) and, finally facilitate effect scores are given as the the worth of the goal multiplied by 0.1 (we have used arbitrary scaling factors for the purpose of exposition, but they are in any case domain specific). Thus, if we take the third settlement (`jan_21_05`) as an example, we can see that flying on this date facilitates (f) the *meeting* goal but hinders (h) the *colleague* goal and conflicts with the *conference* goal. The sum of the scores for each settlement is given in the last column, where we can see that the `jan_21_05` has a total score of 0.2, since the date facilitating the *meeting* goal is given a score of 1.0 (the worth of the facilitated goal multiplied by 0.1), the hinder effect on the *colleague* goal has a score of -0.2, and the conflict effect on the *conference* goal gives a score of -0.6. Thus, we subtract the negative scores from the positive scores to give the total score for the settlement, where this is derived entirely from the effects on existing goals.

4.4.4 Scoring Settlements Against Existing Goals

In traditional deliberative agent architectures, goals are not assigned a value but, by incorporating motivational mechanisms into such architectures, goals can be generated and given value or *worth* by the motivations responsible for their generation. This provides us with the means to evaluate the impact that different issue settlements have on goals.

Formally, therefore, we start by defining a function, *sum*, which takes a set of goals and their associated worths and returns the sum. The function is defined recursively with the base case applied to an empty set returning 0.

$$
\begin{array}{|l}
sum : \mathbb{P}(NegotiationGoal \times Worth) \\
\hline
sum\ \varnothing = 0 \\
\forall w : Worth, g : NegotiationGoal, gws : \mathbb{P}(NegotiationGoal \times Worth) \bullet \\
\qquad sum(\{(g, w)\} \cup gws) = w + sum(gws)
\end{array}
$$

The score given to a particular issue settlement is then calculated using the function, *worthscore*, defined below. As a settlement is to be scored against a set of existing goals, we give as arguments a settlement and a set of goals (along with their worths) representing the existing goals. The score given to a settlement is then simply the sum of the worths of the facilitated goals minus the sum of the worths of the conflicting and hindered goals.

$$
\begin{array}{|l}
worthscore : Settlement \longrightarrow \mathbb{P}(NegotiationGoal \times Worth) \longrightarrow Worth \\
\hline
\forall s : Settlement;\ gws : \mathbb{P}(NegotiationGoal \times Worth) \bullet \\
worthscore\ s\ gws = sum(facillitatedgoals\ s\ (\mathrm{dom}\, gws)) - \\
(sum(conflictedgoals\ s\ (\mathrm{dom}\, gws)) + \\
(sum(hinderedgoals\ s\ (\mathrm{dom}\, gws)))
\end{array}
$$

Note that the score given to a settlement can be both positive or negative, since the negative effects on hindered and conflicting goals may be greater than the positive effects on the set of facilitated goals.

4.4.5 Preference Construction

Once each potential settlement has received a score based on its effect on existing goals, we can represent the preference ordering over the issue-values used to make settlements. As an example, consider again Alice's goal to travel to New York for a conference. The departure date can be settled using the set of dates below.

$$Dates = \{\texttt{jan\_19\_05},\ \texttt{jan\_20\_05},\ \texttt{jan\_21\_05}\}$$

By comparing each possible date against Alice's existing goals, we can discover if they hinder, conflict or facilitate any of them, and thereby give each a score using the functions defined in Section 4.4.4. Then, we can examine the scores of any two dates and determine when one is preferred to another. So, for example, if the score given to `jan_19_05` is greater than the score given to `jan_21_05`, then `jan_19_05` is preferred over `jan_21_05`.

Thus, for our example, the set of dates will be ordered in the following manner. First in the order is `jan_19_05`, since this has been given the lowest score (0.0), next is the `jan_21_05`, since it has the next lowest score (0.2), and finally the aspiration settlement of `jan_20_05` has a score of 1.4, which is placed in the last position of the order.

$$Dates = \langle \texttt{jan\_19\_05},\ \texttt{jan\_21\_05},\ \texttt{jan\_20\_05} \rangle$$

4.5 Conclusion

Autonomous agents generate their own goals and so, when these goals are to be satisfied through negotiation, we must be able to identify which parts of the goals are to form the negotiation issues. This involves reasoning about the different ways in which the issues can be settled and deriving preferences over these alternatives. In order to do this in an autonomous manner, it is important that we provide the necessary mechanisms to guide the derivation of these preferences and, in this chapter, we have shown how this can be achieved.

In order to be able to classify issues, it is necessary that the representation of goals is sufficiently expressive to allow reasoning about which parts should become issues. Our model of goals provides this representation, by defining goals to be composed of two qualitatively different kinds of attributes. *Objective attributes* describe the main part of the goal that is to be achieved, and *issue attributes* form the set of attributes that can potentially be made into negotiation issues. These latter attributes are examined in reference to existing goals in order to determine preferences over how they might be settled in a negotiation.

In this way, the identification of appropriate settlements in terms of existing goals is similar to the formation of intentions in BDI architectures in general, which are tested for their compatibility with existing intentions [27]. Thus, our approach links in with the traditional operation of BDI agents, and exploits similar notions of consistent action as described by Bratman [12]. However, we provide a more fine-grained view in which the different ways a goal can be instantiated, specifically in relation to the issues of negotiation goals, are examined against already instantiated, existing goals.

By understanding the effects of different potential settlements of issues on existing goals, we can determine the strength of preferences towards them, in order to generate an ordering over the different settlements. By making use of the existing goals of agents in the derivation of preferences towards issues, we not only ensure that such preferences are determined autonomously, but we also ensure that the preferences so determined are sensitive to an agent's ongoing activity-related context, i.e. its goal-related activity. This means that the preferences expressed over the issues of a goal are guaranteed to be consistent with ongoing concerns, and thus we can ensure that any negotiated settlement subsequently reached through negotiation is similarly consistent with current goals.

We abstract out the problem of identifying when settlements conflict or facilitate goals, though there are methods that could be applied. In particular, it may be possible to constrain the relationships between settlements and goals, though this would require domain knowledge and is not assumed in our general treatment.

Chapter 5

Bilateral Issue Combinations

5.1 Introduction

In this chapter we turn to the combinations of issue classifications by different agents that arise for a particular negotiation goal to produce negotiations with different characteristics. More specifically, different combinations of issue types (i.e. fixed, slack and negotiable) determine the nature of the negotiation. For example, if I want to buy a car with power steering, I classify the issue of power steering as fixed. However, all the cars offered by my supplier can be ordered with or without power steering at no extra cost, so power steering is a slack issue since, for him, it is of no consequence which I choose. In this case, the issue becomes irrelevant, since negotiation over it is redundant given the (lack of) preferences of the salesman. Alternatively, if none of the supplier's cars come with power steering then the issue for the salesman will be fixed but to a different settlement to mine, i.e. no power steering. In this case the issue may cause the negotiation to fail since, with our current preferences, no agreement can be found.

Such combinations of issues are a natural consequence of the *bilateral* decision process of negotiation, in which the outcome is determined jointly through the preferences of both participants. Even though each participant starts by unilaterally determining what it wants to negotiate over, the final outcome can only be established once the unilateral classifications of each participant are determined, and the bilateral combinations that arise identified.

The combination of unilateral issue types produces new, *bilateral issues*, and by examining these we can understand the properties of the resulting negotiation which, in turn, enables the determination of the difficulty of finding an agreement.

This chapter has the following organisation. In Section 5.2 we introduce the taxonomy of bilateral issue types. In Section 5.3 we describe how the bilateral issues can be determined by a buyer agent using information about sellers, and in Section 5.4 we offer some concluding remarks.

5.2 A Taxonomy of Bilateral Issue Types

As we have seen in the previous chapter, different unilateral issue classifications reflect preferences over the different ways in which issues can be settled. A preference for only one settlement leads to the issue being classified as fixed; a preference for a range of different settlements leads to the issue being classified as negotiable; finally, when no preference between different settlements can be identified, the issue is slack. When considering the unilateral issue classifications of *both participants*, however, the resulting combinations produce different outcomes.

5.2.1 Zones of Agreement

In order to establish the outcomes that result, it is necessary first to determine the set of *preferences* for each participant from the unilateral perspective and, second, to establish the points of overlap or intersection. Negotiation can be seen as a search to find a settlement within the preferences of both participants, where this space is known as the *zone of agreement* (ZoA). This is simply the set of settlements that *both* participants consider acceptable. For example, suppose Alice and Bob hope to travel together, so must settle the issue of which airport to depart from, where the possible settlements are the London airports given below:

$$London\_airports = \{\texttt{luton, heathrow, gatwick, stansted}\}$$

Alice's preferences are for $\{$`luton, heathrow, gatwick`$\}$, while Bob's are for $\{$`gatwick, stansted`$\}$. In this case, the ZoA contains that settlement option common to both, i.e. `gatwick`.

Recall that our model defines for each issue-attribute a *range* that includes all the issue-values that can be used to settle the issue. On this range an ordering is defined representing preferences over the issue-values, and it is the properties of the preference ordering which are then used to classify the issue as either fixed, slack or negotiable. To facilitate the presentation of our model we provide the axiomatic function, *preferences*, which takes an issue attribute and returns the preference ordering placed on its issue-values, where the first issue-value is the reservation settlement and the last is the aspiration settlement. We also define a function, *preferenceset* which, given an ordered preference sequence returns a set containing all those issue-values in the sequence.

$$preferences : IssueAttribute \rightarrow \operatorname{seq} IssValue$$

$$preferenceset : \operatorname{seq} IssValue \rightarrow \mathbb{P}\, IssValue$$

The definition of the ZoA is then, simply, the intersection of the sets of preferences identified by each participant. This is shown in the *ZoneOfAgreement* schema, which includes two *IssueAttribute* variables (one for each agent in the negotiation) and defines three other variables, two of which represent the sequences of issue-values of each participant and the third is the ZoA. The predicate part of the schema obtains both agents' preferences using the *preferences* and *preferenceset* functions defined above, and the union of the two sets is defined as a subset of the union of both issue attributes' *ranges*. Finally, the schema defines the ZoA as the intersection of these two sets.

ZoneOfAgreement

$$\begin{array}{l}
i1, i2 : IssueAttribute \\
s1, s2 : \operatorname{seq} IssValue \\
zoa : \mathbb{P}\, IssValue \\
\hline
s1 = preferences\ i1.range \\
s2 = preferences\ i2.range \\
(preferenceset\ s1 \cup preferenceset\ s2) \subseteq (i1.range \cup i2.range) \\
zoa = (preferenceset\ s1 \cap preferenceset\ s2)
\end{array}$$

5.2.2 Bilateral Issue Types

We can now use this notion of the ZoA to identify different classes of bilateral issues resulting from the different possible combinations of unilateral issues.

- *Congruent* issues are those for which the unilateral classifications made by the participants combine to produce issues for which the aspiration settlements of both are the same.

- Conversely, *deadlocked* issues arise when there is no overlap of the preference sets of the participants, i.e. when the ZoA is empty.
- Finally, *competitive* issues arise when the preferences of the participants are *partially compatible*, which occurs when they share preferred settlements but do not share the same aspiration settlement. Furthermore, within the competitive issue class, two sub-classifications can be made.
  - When one participant's preference set contains only one issue-value (i.e. its unilateral issue classification is for a fixed issue), and this is also in the negotiable preference set of the other, we say that the issue is *one-way negotiable.*
  - Alternatively, when both participants' preference sets contain more than one settlement and these sets overlap, then the issue is *two-way negotiable.*

We consider each of these categories in more detail below and provide formal definitions for each.

5.2.3 Deadlocked Issues

As described above, a deadlocked issue is one for which there is no overlap of the preference sets of each participant, i.e. the ZoA is empty. To take the `airport_departure` issue as an example, if an agent, say the buyer, prefers the settlements {`luton, heathrow`}, and the seller prefers {`gatwick, stansted`}, then it is clear that there is no overlap in their preferences, so this issue is in deadlock.

Formally, we define an issue as deadlocked in the following schema. The schema includes the *ZoneOfAgreement*, and the predicate part simply states that the ZoA is an empty set.

Deadlocked

ZoneOfAgreement

$zoa = \varnothing$

5.2.4 Congruent Issues

Congruent issues are those for which the participants share the same aspirational settlement. This can occur in two cases. First, if either or both of the agents classify an issue as slack, then the issue is congruent since, if a participant is indifferent to the settlement of the issue (as defined by a slack issue), then the preferences for this agent must include *all* the alternative settlements defined in the issue's range, and therefore must include the aspirational settlement of the other participant. Second, when neither classifies the issue as slack but both select the same aspirational settlement the issue is congruent.

The *Congruent* schema gives the formal definition of the above description. Its predicate states that either agents' *preferenceset* must contain all the settlements defined in the associated issue attribute's *range* (thus defining a slack issue), or both participants aspirational settlement (i.e. the last in their respective *preferences*) is the same.

Congruent

ZoneOfAgreement

$$\begin{array}{l} \#preferenceset(s1) = \#range(i1) \lor \\ \#preferenceset(s2) = \#range(i2) \lor \\ last\, s1 = last\, s2 \end{array}$$

5.2.5 Competitive Issues

Issue classifications that produce *competitive issues* are those for which the participants preferences are partially compatible. This means that although the participants' aspirational settlements are not the same, there are some less preferred settlements that are common to both, and it is the competition between the participants to obtain the best of these that defines the competitive bilateral issue type.

Recall there are two sub-classifications of competitive issues: one-way and two-way negotiable. In order to define issues that are two-way negotiable, we must ensure that both participants have classified the issue as negotiable which, by definition, entails that both sets of preferences contain more than one settlement. We must ensure also, however, that both sets contain fewer than the complete range of settlements, since this would entail a slack classification. Finally, the ZoA must contain at least one settlement since, otherwise a deadlocked issue is defined.

Formally, this is defined in the *TwoWayNeg* schema, in which both sets of preferences are restricted to contain more than one settlement but less than the total number defined in the *range*.

$$\begin{array}{|l}
\hline
\textit{TwoWayNeg} \\
\textit{ZoneOfAgreement} \\
\hline
\#\textit{preferenceset}\ s1 > 1 \wedge \#\textit{preferenceset}\ s2 > 1 \\
\#\textit{preferenceset}\ s1 < \#\ i1.\textit{range} \wedge \#\textit{preferenceset}\ s2 < \#\ i2.\textit{range} \\
\#\textit{zoa} \geq 1 \\
\hline
\end{array}$$

One-way negotiation issues involve only one of the participants having preferences for more than one settlement, while the other has a preference for only one, which must be included within the former set. To define this, we state that one of the participant's preference sets contains only one settlement, which must also be included in the preferences of the other participant, and that this set must contain more than one settlement. The ZoA must also contain just one settlement, i.e. the one common to both preference sets. The *OneWayNeg* schema describes this formally.

$$\begin{array}{|l}
\hline
\textit{OneWayNeg} \\
\textit{ZoneOfAgreement} \\
\hline
\exists_1\ s : \mathbb{P}\,\textit{IssValue} \mid s \in \{\textit{preferenceset}\ s1, \textit{preferenceset}\ s2\} \bullet \#s = 1 \\
\#\textit{zoa} = 1 \\
\hline
\end{array}$$

We also provide a data type definition for one and two way negotiable issues that makes the development of the model in later chapters simpler to follow.

$$\textit{Competitive} ::= \textit{oneway}\langle\langle \textit{OneWayNeg} \rangle\rangle \mid \textit{twoway}\langle\langle \textit{TwoWayNeg} \rangle\rangle$$

5.2.6 An Example: Combining Unilateral Issue Classifications

Bilateral issues arise from the unilateral issue classifications of each participant. Assume, for example, that a buyer must classify three issue-attributes (classifying one as slack, one as fixed and the other as negotiable), while a seller must also make its own classifications. For the issues that the buyer classifies as slack, the seller can make one of three classifications; it can classify the issue as similarly slack, as fixed, or as negotiable. In the same way, out of those issues the buyer has classified as fixed, the seller may classify them again as either slack, fixed or negotiable, and so on.

TABLE 5.1: An example set of bilateral issues that arise from agents' unilateral issue classifications for three issues

Buyer/Seller	**Slack**	**Fixed**	**Negotiation**
Slack	con	-	-
Fixed	-	con / dead	-
Negotiation	-	-	comp/ dead / con

In order to clarify the above description, consider Figure 5.1, in which the unilateral classifications of three issues for a buyer and seller are represented, as well as the resulting bilateral issue types. On the left of the figure is a negotiation goal, beside which the potential classifications of the buyer are shown. To the right of this, the potential classifications of the seller in relation to the buyer's classifications are also shown. In the figure, example classifications made by both participants are circled to show the resulting bilateral issue. We can see that the combinations result in a negotiation in which there is one issue that is either competitive or deadlocked (negotiable-negotiable), one that is either congruent or deadlocked (fixed-fixed) and one that is congruent (slack-slack). Whether or not the first two issues are deadlocked depends upon the existence of a ZoA. Table 5.1 summarises the bilateral issues that result from the unilateral issue classifications of the agents in the figure (where *comp* is competitive, *con* is congruent and *dead* is deadlocked).

5.2.7 Bilateral Negotiation

Now that we have defined the possible *bilateral issues* that can arise from the unilateral issue classifications of the negotiation participants, we can define negotiation itself. We are concerned only with negotiations between two participants, so negotiation is defined as an interaction between two agents in which one wants to have a goal satisfied, the *buyer*, and the other can satisfy the goal, the *seller*.

Formally, negotiation is described in the *Negotiation* schema, which includes a *buyer* and a *seller*, both of which are defined as *autonomous agents* (introduced as a given set). Also included is the definition of a negotiation goal and, lastly, the schema includes the bilateral issue types, *deadlocked*, *competitive* and *congruent*, resulting from the combination of unilateral classifications by the participants.

$[AutonomousAgent]$

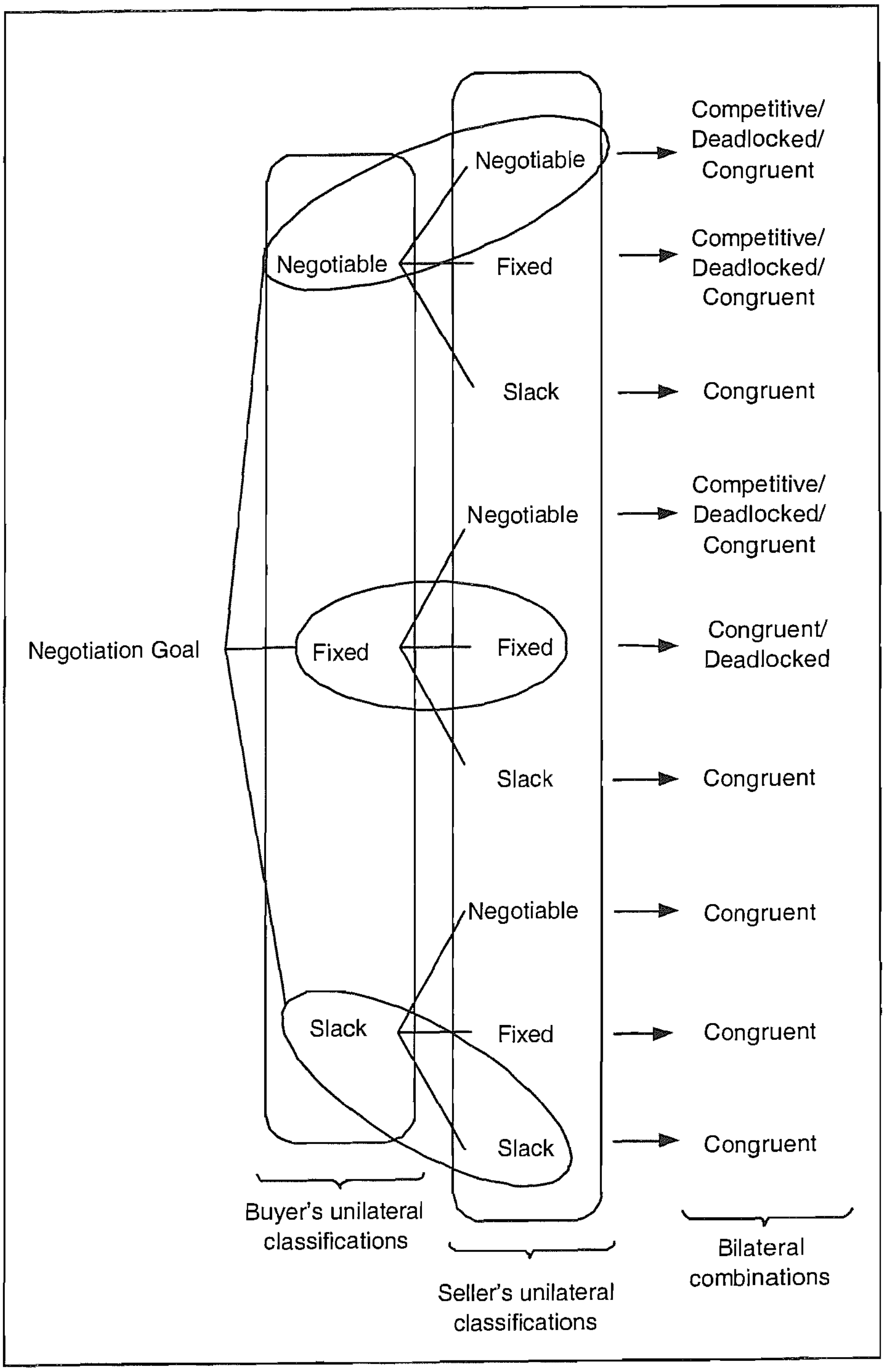


FIGURE 5.1: Issue classification combinations

Negotiation

$\mathit{buyer}, \mathit{seller} : \mathit{AutonomousAgent}$
$\mathit{NegotiationGoal}$
$\mathit{deadlocked} : \mathbb{P}\,\mathit{Deadlocked}$
$\mathit{competitive} : \mathbb{P}\,\mathit{Competitive}$
$\mathit{congruent} : \mathbb{P}\,\mathit{Congruent}$

5.2.8 Negotiations With Different Issue Profiles

An analysis of the different bilateral issue types that are possible for an issue, given the unilateral classifications of the participants, provides us with a way to identify the degree of difficulty in reaching a settlement over that issue. For example, it is clear that finding a settlement for a deadlocked issue is harder than finding one for a congruent issue. However, negotiations often involve more than one issue and, by analysing the different *patterns* of bilateral issue types, we can build up a picture of how difficult a negotiation (considered as a collection of such bilateral issues) may be.

Given the bilateral issues discussed above, we can describe negotiations comprising different bilateral issue profiles, shown in Figure 5.2. Unary negotiations are those containing either all congruent, all competitive or all deadlocked issues (1-3 in the figure). Negotiations containing two forms of bilateral issue types are binary negotiations (1-4 in the figure) and, finally, ternary negotiations, in which all three types of issues occur (7 in the figure). The *relative proportion* of bilateral issue types has consequences for the negotiation; for example negotiations in which the majority of issues are in deadlock are more difficult to find agreements for than those that contain just one. In the next section we show how buyers can use information about sellers' past issue classifications and settlement choices to estimate the relative proportion of such bilateral issues in forthcoming negotiations. Although the analysis is presented from the perspective of the buyer, the process can be applied to the seller in an identical manner.

5.3 Bilateral Issue Analysis

Now that we have a model of bilateral negotiation issues, we need to examine how it might be used by a buyer to better understand the kind of negotiation in which it is participating and the chances of success for the different issues. Since any individual negotiation is in itself unpredictable, we can only establish the likelihood of success if

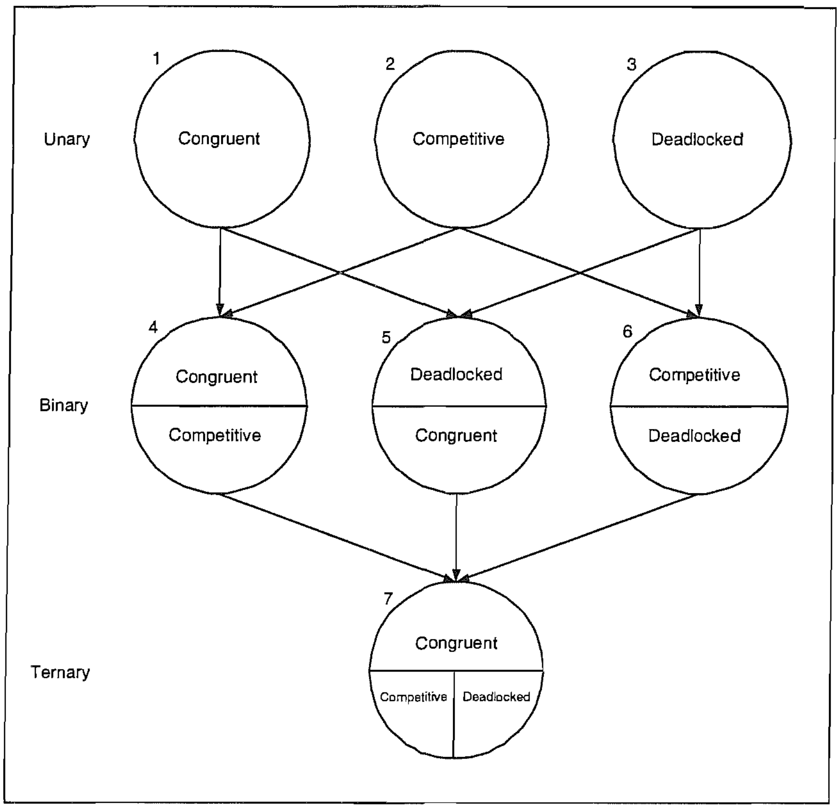


FIGURE 5.2: Negotiations comprised of different bilateral issue types

we have information on prior negotiations with the same seller over the same issues. More specifically, a buyer requires historical information about sellers to enable it to form expectations about the seller's likely future issue classifications. The buyer can thus build up a record of past negotiations with sellers, recording classifications of issues and their preferred settlements of those issues. In turn, this allows the buyer to determine likely bilateral issue types. Such historical information allows a buyer to evaluate the risk involved in negotiating with a given seller, such as the risk that deadlock may occur. This information can then be exploited in order to provide expectations about sellers' likely issue classifications and settlement preferences in forthcoming negotiations.

The information we require regarding sellers must provide some record of their classifications of issues in previous negotiations over the same goal, which can be obtained either from direct interactions, or from third-party interactions. (Using third-parties brings its own problems, however, first since they must be trusted, and second since the history of interactions with one party may not reflect the likely interactions with another. Addressing problems of trust, however, is not within the scope of this thesis, and instead we simply assume we have access to accurate information, and that there

TABLE 5.2: The negotiation issues and their settlements

Issue	Settlements
`departure_date`	`{19_jan_05, 20_jan_05, 21_jan_05}`
`seat_class`	`{economy, business, first}`
`departure_airport`	`{gatwick, heathrow, luton, stansted}`

is some regularity in interactions that enables inferences of this kind to be made. Solutions to the issues arising in relation to trust and reputation can be found in a range of sources, including [47, 84, 112, 117, 118]).

As just indicated, we assume that agents perform *repeated negotiations* with each other over the same goal, and that sellers display some regularity in their issue classifications; that is, we do not rely on third party assessments.

5.3.1 Example Scenario

Consider the scenario in which Bob wants to buy an airplane ticket to travel to a conference. Bob can obtain a ticket from several suppliers, but each has constraints on the tickets offered, such as the *date of departure*, the *seat class* and the *departure airport*, which represent the negotiation issues under consideration. Each issue has several potential settlements, shown in Table 5.2.

5.3.2 Issue Classification Profiles

We begin by considering the *relative frequency* of classification of issues by sellers into the unilateral issue types of fixed, slack and negotiable. Relative frequency is simply the proportion of classifications of a particular issue as fixed, slack or negotiable in relation to the total of number of times it has been negotiated. Thus, over ten different negotiations for the same goal, if an issue is classified by a seller as fixed on six occasions, then it has a relative classification frequency of 0.6. This can be done for all issues, and all unilateral issue types. For example, in Table 5.3, a set of classification frequencies is given for the different issues of the goal, with the `departure_date` issue having a relative frequency as *fixed* of 0.5.

This kind of information is important, since it allows the buyer to estimate the likelihood of an issue being classified as fixed, negotiable or slack, which then helps to determine the likely bilateral issue types.

TABLE 5.3: Seller Classification frequencies for three issues

Issue attributes	Issue types		
	Fixed	**Slack**	**Negotiable**
`departure_date`	0.1	0.7	0.2
`seat_class`	0.4	0.4	0.2
`departure_airport`	0.5	0.2	0.3

TABLE 5.4: Different bilateral issue outcomes based on unilateral classifications of two negotiators

Buyer/Seller	Slack	Fixed	Negotiable
Slack	con	con	con
Fixed	con	con / dead	comp / dead / con
Negotiable	con	comp / dead / con	comp / dead / con

5.3.3 Settlement Selection Profile

In order to determine the kind of negotiation that will result with a given seller, we must understand how the different settlements preferred by the buyer and the seller for each issue forms the issue's ZoA. However, since this is based on an examination of the frequencies associated with a seller's issue classifications and expected settlement preferences, we refer now to the *expected zone of agreement* (EZoA).

Like issues, the *offers* previously made by sellers can be used to determine likely settlements, being derived from a combination of the frequency of prior settlement offers, and their *position* in an offer sequence. If we assume that the first offer is the most preferred settlement, with further offers representing increasingly less preferred settlements, then recording the relative frequency and position of settlements in offer sequences allows us to build up expectations about the likely preference for settlements in future negotiations.

For each combination of unilateral issue types, a number of different outcomes are possible depending on the nature of the EZoA. In Table 5.4 we show the full range of outcomes based on an analysis of the EZoA, which we examine in detail below.

5.3.4 Expected Fixed Seller Issues

5.3.4.1 Negotiable Buyers

When a seller classifies an issue as fixed, it proposes *only one* settlement. If we assume the buyer classifies the issue as negotiable, three possible outcomes can result: a

TABLE 5.5: Selection frequencies of different settlements for the departure airport issue with a fixed seller

Settlement	**Selection frequency**
`luton`	0.5
`gatwick`	0.33
`stansted`	0.16

deadlocked, a one-way negotiable or a congruent bilateral issue.

Returning to the example scenario, suppose we focus on the `departure_airport` issue, for which the buyer's preference set is as follows.

$$\{\texttt{gatwick, heathrow, luton}\}$$

In previous negotiations over the same goal in which the seller classified this issue as fixed, its preference was for different settlements at different times. If, over six previous negotiations, it preferred `stansted` once, `gatwick` twice, and `luton` three times, the relative frequency of `luton` is therefore 0.5. Similarly, `gatwick` has a frequency of 0.33, and `stansted` of 0.16, as shown in Table 5.5. Now, by considering the buyer's current preferences, we can determine the likelihood of arriving at a deadlocked, congruent or one-way negotiable bilateral issue type.

First, we examine the likelihood of a deadlocked issue arising. This is simply the sum of the frequencies of the seller's previously preferred settlements that are *not* in the buyer's current preference set, as shown in Figure 5.3, multiplied by the frequency that the issue has been classified as fixed by the seller, which (as Table 5.5 shows) is 0.5. In the figure, each settlement is shown with frequencies (according to prior seller negotiations) given in brackets. Summing the relative frequencies of the elements that are not in the preference set of the buyer, and multiplying by the classification frequency, gives an expectation of deadlock for the issue. It is clear from the figure that this is simply the product of the value for `stansted`, which is 0.16, and 0.5 = 0.08.

In the same manner, we can determine the expectation of a one-way negotiable bilateral issue as the sum of the relative frequencies of those settlements that are common to both participants multiplied again by the seller's fixed classification frequency for the issue. This is shown in the figure as the intersection of the preference sets (i.e. `gatwick` and `luton`), giving an expectation of $(0.33 + 0.5) \times 0.5 = 0.42$.

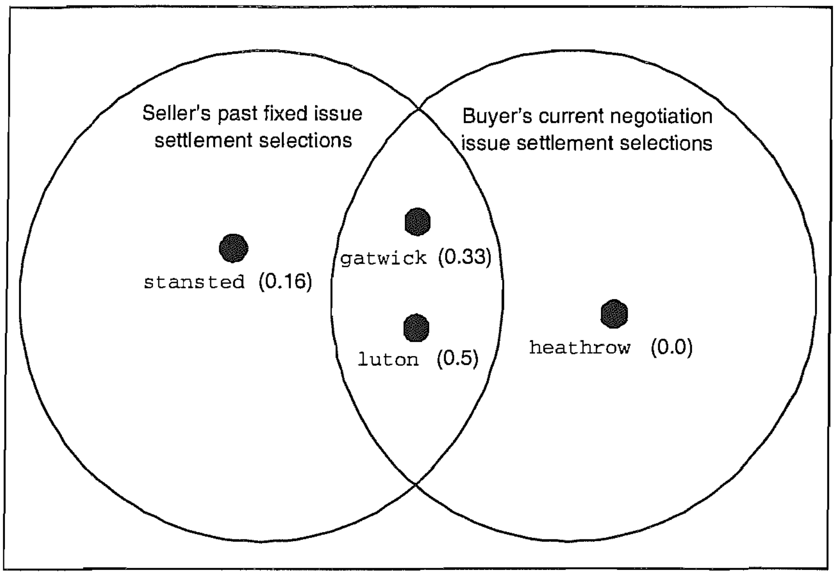


FIGURE 5.3: Buyer's negotiation settlement preferences and seller's previous fixed settlement preferences

Finally, the expectation of a congruent issue is given simply by determining the frequency of the buyer's aspiration settlement in terms of prior selection by the seller, and multiplying this frequency by the fixed classification frequency. Thus, if `gatwick` is the aspiration settlement of the buyer, then the expectation of congruence is simply the seller's selection frequency of 0.33 multiplied by 0.5 = 0.16.

5.3.4.2 Fixed Buyers

If instead, however, the buyer classifies the issue as fixed, there are two possible outcomes: a deadlocked issue or a congruent issue.

In this case, the expectation of deadlock is the sum of the frequencies associated with settlements not in the buyer's current preference set multiplied by the fixed classification frequency, as shown in Figure 5.4, where this is $(0.16 + 0.33) \times 0.5 = 0.24$. The expectation that a congruent issue type will result is simply the relative frequency of the seller selecting the same settlement as the buyer which, here, is 0.5 for `luton` multiplied by the fixed classification frequency of the seller, i.e. $0.5 \times 0.5 = 0.25$.

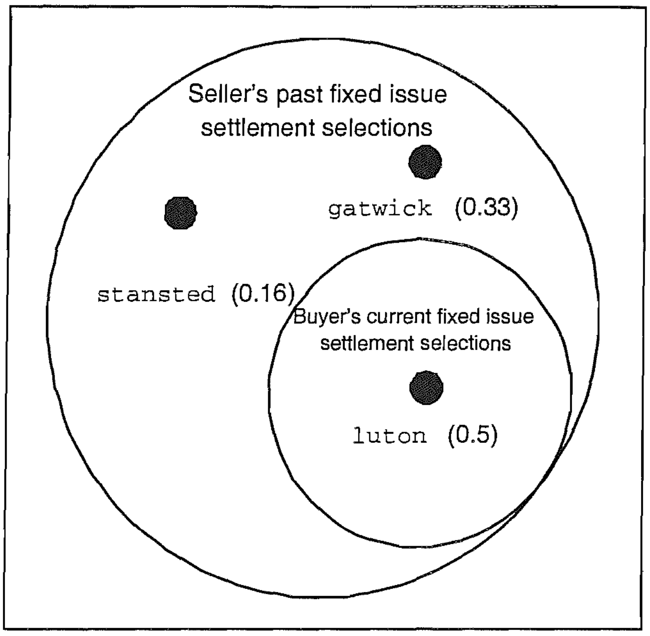


FIGURE 5.4: Buyer's fixed settlement preference and seller's previous fixed settlement preferences

TABLE 5.6: Relative expectation for bilateral issues types for a fixed seller

`departure_airport`	**Bilateral Issue types**		
	Deadlocked	**One-way negotiable**	**Congruent**
Negotiable buyer	0.08	0.42	0.16
Fixed buyer	0.24	-	0.25
Slack buyer	-	-	0.5

5.3.4.3 Slack Buyers

Finally, if the buyer classifies the issue as slack, then the bilateral issue that arises must be congruent, since there is no conflict (as the expectation is the same as the sum of frequencies of all the seller's previous settlements, i.e. 1), as shown in Figure 5.5. The expectation for this is then simply the seller's fixed classification frequency, of 0.5.

With this analysis we can establish the relative expectation that the issue will be either deadlocked, one-way negotiable or congruent. Table 5.6 summarises the expectation values for each bilateral issue type. It is clear from the table that: for a negotiable buyer the issue is most likely to result in a one-way negotiable bilateral issue type (as indicated by the high expectation value for this type); for a fixed buyer the expectation is for a congruent issue; and for a slack buyer the expectation is similarly for a congruent issue.

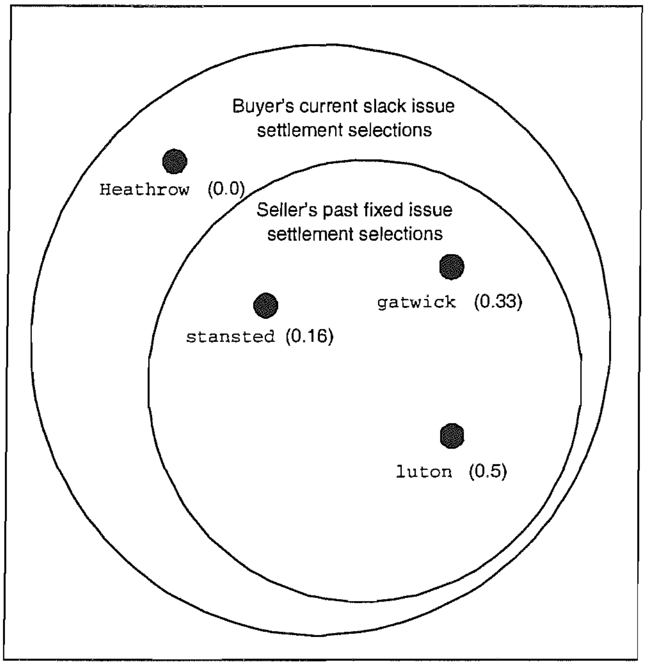


FIGURE 5.5: Buyer's slack settlement preferences and seller's previous fixed settlement preferences

5.3.5 Expected Negotiable Issues

When an issue is classified by a seller as negotiable, different settlement offers are made in sequence, from the reservation to the aspiration settlement. This gives us a range of values to consider, rather than just one fixed settlement, as previously. However, we can still perform the same analysis on the settlements in these sequences by considering the selection frequency of any given settlement over different negotiation instances.

Suppose the seller settlement profile contains the offer sequences shown in Table 5.7, over six different negotiations for the `departure_airport` issue. Across all negotiations, `gatwick` has been offered three times out of six negotiations (in negotiations 1, 3 and 4), giving it a selection frequency of 0.5. Similarly, `manchester` occurs four times, giving a frequency of 0.66. Now, suppose that the buyer's preferences for the issue (or its settlement offer sequence) is as follows, with the aspiration settlement being `stansted`, followed by `gatwick`, and then `heathrow` as the reservation.

$$\langle \texttt{heathrow, gatwick, stansted} \rangle$$

TABLE 5.7: Offer series of a negotiable seller over six different negotiations for the same goal

Negotiation instance	Offer Series
Negotiation 1:	⟨`gatwick, luton, heathrow`⟩
Negotiation 2:	⟨`bournemouth, southampton, manchester`⟩
Negotiation 3:	⟨`gatwick, manchester, heathrow, luton`⟩
Negotiation 4:	⟨`gatwick, edinburgh, heathrow, liverpool`⟩
Negotiation 5:	⟨`edinburgh, manchester, liverpool`⟩
Negotiation 6:	⟨`edinburgh, luton, stansted`⟩

TABLE 5.8: Negotiable seller preferences not in the preference set of the buyer

Negotiation instance	Settlement
2	⟨`bournemouth, southampton, manchester`⟩
5	⟨`edinburgh, manchester, liverpool`⟩

TABLE 5.9: Negotiable seller preferences in the preference set of the buyer

Negotiation instance	Settlement
1	⟨`gatwick, luton, heathrow`⟩
3	⟨`edinburgh, manchester, liverpool`⟩
4	⟨`edinburgh, manchester, luton`⟩
6	⟨`edinburgh, luton, stansted`⟩

If a seller classifies the issue as negotiable, and a buyer does the same, then it can produce a deadlocked issue, a two-way negotiable issue or a congruent issue.

Given the above settlement profiles of the buyer and seller, the expectation of deadlock is given by the frequencies with which the seller's settlement sequences *do not* contain any of the current preferences of the buyer, as shown in Table 5.8. Thus, from six negotiations, the frequency with which the seller's preferences are distinct from the buyer's is 0.33. By multiplying this with the seller's negotiable classification frequency for the issue, we obtain the expectation of $0.33 \times 0.3 = 0.09$ that the issue will be deadlocked.

Similarly, the frequency that the issue will be two-way negotiable is the frequency with which the seller's offer sequences in previous negotiations contain *at least one* of the buyer's current preferences (as shown in Table 5.9) multiplied by the seller's negotiable classification frequency. Thus, in the previous six negotiations, the seller made four offer sequences that contain one or more of the buyer's current preferences, giving a frequency of 0.66, and the expectation for a two-way negotiable bilateral issue type is therefore $0.66 \times 0.3 = 0.19$.

TABLE 5.10: Relative expectation for bilateral issues types for a negotiable seller and negotiable buyer

`departure_airport`	**Bilateral Issue types**		
	Deadlocked	**Two-way negotiable**	**Congruent**
Negotiable buyer	0.09	0.19	0.04

TABLE 5.11: Relative expectation of bilateral issues types for a slack seller

`departure_airport`	**Bilateral Issue types**		
	Deadlocked	**Two-way negotiable**	**Congruent**
Negotiable buyer	-	-	0.2
Fixed buyer	-	-	0.2
Slack buyer	-	-	0.2

Finally, the expectation of a congruent bilateral issue is the likelihood that the seller's most preferred settlement is the same as the buyer's. We can simply examine the seller's settlement profile to determine the number of times the buyer's most preferred settlement was offered by the seller as its first offer. Then, since `stansted` is the buyer's most preferred settlement, and there is only one instance of prior seller negotiations with `stansted` as the most preferred settlement (Negotiation 6), we get a selection frequency of 0.16, and an expectation for a congruent issue of $0.16 \times 0.3 = 0.04$.

Since the bilateral issues that result if the buyer makes a fixed classification while the seller makes a negotiable classification are the same (albeit reversed) as when the seller fixes the issue and the buyer classifies it as negotiable, we do not repeat the analysis. Similarly, when the buyer classifies the issue as slack in the face of a negotiation classification of the seller, the analysis is again the same as the reversed situation described above regarding congruent issues, and so again we do not repeat it.

The relative expectation for each bilateral issue type, given the above analysis for a negotiable buyer is shown in Table 5.10.

5.3.6 Expected Slack Issues

When the seller is expected to classify an issue as slack, there is no need for further analysis as all possible settlements are acceptable, regardless of the buyer's issue classification. The expectation of a slack classification by a seller is therefore simply the issue's slack classification frequency, as shown in Table 5.11.

TABLE 5.12: Issue classification for buyer and seller

Issue	**Buyer**		
	Fixed	**Slack**	**Negotiable**
`departure_date`	✓		
`seat_class`			✓
`departure_airport`			✓

Issue	**Seller**		
	Fixed	**Slack**	**Negotiable**
`departure_date`			✓
`seat_class`		✓	
`departure_airport`			✓

5.3.7 Candidate Negotiation Opponents

By performing such an analysis on all the issues of a negotiation, we obtain the relative expectation of each for the likely bilateral issues types that may arise. To finish the analysis, we now need to consider how this can be used in the current negotiation. In particular, we must determine the likely bilateral issue types across all issues of the current negotiation, given the current unilateral issue classification of both participants. To make this clear, let us assume that, in the current negotiation (ignoring the extensive analysis until now) the unilateral issue classifications of both the buyer and seller are as shown in Table 5.12 for our example goal of buying a flight ticket. Then, after an analysis of the zones of agreement as above, we arrive at the relative expectation for each issue's resulting bilateral issue type.

Table 5.13 brings all these values from the historical analysis together. From Table 5.12, we can see that the `departure_airport` issue has been classified as negotiable by the buyer and negotiable by the seller. This means that the issue may give rise to either a deadlocked, two-way negotiable or congruent issue. After examining the relative selection frequencies of the seller (which we have worked through in the previous discussion in Section 5.3.5 and which are also shown in Table 5.10) we obtain the expectation for each bilateral issue type (shown in the `departure_airport` row of Table 5.13). The `seat_class` issue has been classified as negotiable by the buyer and slack by the seller, so that this issue will give rise to a congruent bilateral issue type with an expectation equal to the frequency that the seller has selected this issue to be slack in prior negotiations (which we show in the table, but have not worked through previously, for reasons of brevity). Finally, the `departure_date` issue has been classified as fixed by the buyer and negotiable by the seller, so that the issue can give rise to either a deadlocked, one-way negotiable or a congruent bilateral issue type.

TABLE 5.13: Relative expectancy for bilateral issues types for a slack seller

Issue	Bilateral Issue types			
	Deadlocked	One-way neg	Two-way neg	Congruent
`departure_date`	0.4	0.3		0.02
`seat_class`				0.2
`departure_airport`	0.09		0.19	0.04

TABLE 5.14: Rating sellers with regard to expected bilateral issue types

Seller	Deadlock proportion	Competitive proportion	Congruent proportion
$Seller_1$	0.33	0.33	0.33
$Seller_2$	0.0	0.33	0.66
$Seller_3$	0.0	0.0	1.0
$Seller_4$	0.66	0.0	0.33

We can use this information to estimate the likely behaviour of negotiation opponents in relation to specific issues, and make comparisons between different opponents.

We can also use the information to estimate, for each seller, the relative proportion of the different bilateral issues that will arise. Since each of the issues in Table 5.13 has three distinct expectation values for different bilateral issue types, we can determine the most likely in the current situation by simply assuming that the bilateral issue with the highest expectation is most likely in the current negotiation. Thus, examining Table 5.13, we estimate that the `departure_date` issue will be deadlocked (since this bilateral issue type has the highest expectation), the `seat_class` issue will be congruent (since it is the only bilateral issue type with an expectation value) and the `departure_airport` issue will result in a two-way negotiable bilateral issue. We can then determine the expected proportions of each bilateral issue type for each prospective seller. For example, in Table 5.14 we can see four sellers, each with the expected proportion of each of the bilateral issue types indicated, with $seller_1$ representing the seller from the above analysis. To rank sellers, we can simply look through the table, and order sellers based on their relative proportions of congruent, competitive and deadlocked issues, to give an assessment of the kind shown in Figure 5.2. Although this loses information, it provides a coarse analysis that may be useful in some situations.

5.4 Conclusion

Agents approach negotiation with preferences for issue settlements determined according to their current needs and circumstances. This leads to the unilateral classification of issues into one of the three types defined and described in Chapter 4. However, when two agents meet to negotiate, their unilateral issue classifications combine to produce different types of bilateral negotiation issues. These issues determine the difficulty or ease with which agreement will be reached, depending on the existence and number of deadlocked and competitive bilateral issue types. In this chapter we have provided a taxonomy of bilateral issue types and described how the types arise from the interactions between each participants' set of preferred settlements. We do this by examining the zone of agreement on issues and highlight the properties that define each type of bilateral issue type.

Making information about the bilateral issue types that are likely to arise in a negotiation available to agent negotiators can help to improve the chances of entering into successful negotiations. This is because, by understanding what the likely bilateral issue types will be for a given negotiation, we can enable agents to avoid those negotiations containing high risk of deadlock or those for which finding agreements of issue settlements is likely to take too long in relation to any existing negotiation deadlines.

In the second half of this chapter we have described a way to make information on likely bilateral issue types available to agent negotiators. Our approach involves examining the relative frequencies with which agents make unilateral issue classifications and the settlement choices they make for those issues. This enables a model of the likely future classifications and issue settlement choices of an opponent to be constructed, which can then be used to make reasoned choices between different opponents.

The analysis provided is a simple heuristic method that is merely intended as an illustration of the ways in which the models can be used, and many other methods are possible. In particular, it may be advantageous to adopt more sophisticated analysis techniques such as case-based reasoning [149] or a more advanced probabilistic model, in which information regarding the context within which the negotiation takes place could be exploited in a more efficient manner than the method we propose.

Chapter 6

Environmental Constraints on Pre-Negotiation

6.1 Introduction

The model introduced so far provides a consideration of goals, issues and preferences in relation to an agent's existing goals, and bilateral issue combinations in relation to bargaining. Yet these are not the only factors to consider when determining how best to prepare for negotiation. In particular, environmental factors can have a significant impact on negotiation, especially through *time* available, and *opportunity* in terms of negotiation partners. When time is short and an agreement must be reached, extra constraints may be imposed in order to avoid a limitless or excessive series of offers and counter-offers. Similarly, when there are limited numbers of sellers, the risk of failure in one negotiation may not be mitigated by the opportunity to negotiate with another seller for the same good. Such factors have been examined from a number of different perspectives. For example, with regard to time, Fatima *et al.* examine the effects of deadlines on the ability of agents to reach agreements [36] and, with regard to the availability of other opportunities for satisfying a goal in addition to the current negotiation, Muthoo examines how such alternatives, or *outside options*, affect the way agents negotiate [101]. Our concerns are somewhat different, since we focus on the effects of such factors on the ways that agents prepare in the *pre-negotiation* stage, rather than their effects on the negotiation process itself.

The key points to consider are how these environmental factors impact on the model constructed previously. More specifically, given the model defined in terms of goals, attributes, issues and preferences, how is it possible to ensure that negotiation is tailored

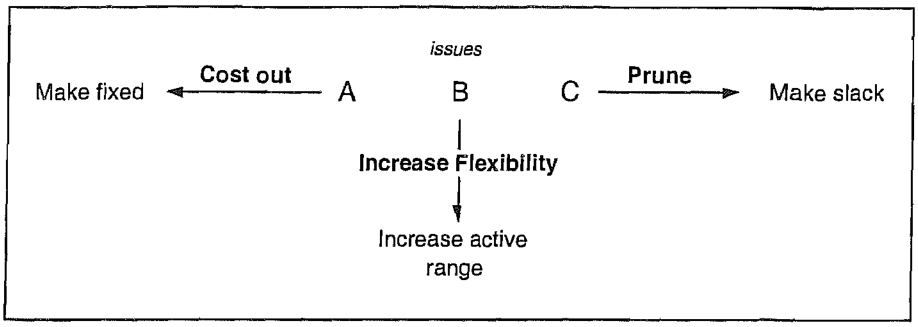


FIGURE 6.1: The three pre-negotiation issue modification mechanisms

to the situation at hand by taking into account time and opportunity constraints? In this chapter, we show how this may be achieved.

In our treatment of time constraints, we examine how time impacts on the classification of negotiable issues and the *competitive* bilateral issues that result. This emphasis on *competitive* issues reflects the fact that it is these issue types that have most impact upon the duration of negotiation (since it is these that form the focus of negotiation, while other issue types do not require negotiation for agreement).

In brief, our approach is to develop mechanisms to enable a *buyer* to alter its negotiation stance in response to environmental constraints. There are three distinct ways of doing this, as shown in Figure 6.1. First, buyers can *cost-out* issues through a process of committing more resources to negotiation in an attempt to influence sellers to make the necessary re-classifications of issues. This enables a buyer to fix its preference for an issue in return for resource. Second, when costing-out is not an option, the buyer can perform its own selective re-classifications of issues to be slack. This effectively reduces, or *prunes* the number of competitive issues in a negotiation. Both of these techniques provide ways for an agent to decrease the duration of a negotiation to meet deadlines. Finally, we provide a way for buyers to increase their flexibility in negotiation, when there are limited numbers of available negotiation partners, by increasing the number of acceptable settlements (within the active range) for an issue. This helps to mitigate the risk of failure for a particular negotiation, which increases as a result of the limited opportunities for negotiation.

The chapter proceeds in the following manner. In Section 6.2 we describe our model of time and how this can be used to determine the deadline and expected duration of negotiation. Section 6.3 describes the process by which the buyer agent can cost out issues in situations of time constraints, and in Section 6.4 we show how, when resources cannot be used, the buyer can reduce the duration of negotiation by selectively removing issues of low worth. In Section 6.5, we turn to constraints on the number of opportunities the buyer has for negotiation, and we describe a method to alter the level of flexibility over issues to address opportunity constraints. Section 6.6 provides some empirical evaluation of the pruning mechanisms, and finally, in Section 6.7 we conclude.

6.2 Negotiation and Time

Typically, time is relevant to negotiation in that a negotiation outcome may be needed by a particular deadline. While there has been some research into examining how time affects the ability to reach negotiated agreements (e.g. [36, 73]), the effects of time on *pre-negotiation*, in which negotiable issues are determined, have been much less explored. Time is also important here, since the classification of issues and the ways these classifications combine to produce bilateral issue types determine the number of issues for which agreement must be found, and hence the time needed.

Consider a situation in which an agent is attempting to negotiate the satisfaction of a goal to book an airplane ticket at short notice. In such a situation, concerns over peripheral issues, such as *seat type* or *airline company*, may not be important enough to warrant being a part of the negotiation. The central objective of getting a ticket in time should form the focus of the negotiation and other, less important issues, should not be allowed to distract from this objective. In this case, it makes sense to discard such peripheral issues, since they do not contribute much to the main objective but could, if included, increase the chance that an agreement will not be found in time. Thus, the time available with which to pursue the achievement of a goal through negotiation is relevant to determining *how* it is pursued. With little time, and when goals must be achieved, a situation in which there are fewer issues to negotiate is clearly more desirable than one in which many issues must be negotiated, since the duration of the negotiation decreases and increases in line with the number of issues. Coming to a quick result, therefore, requires that we identify ways to limit this duration to avoid exceeding any time constraints.

It is this relationship between the time available and the time necessary to find an agreement that must be explored if the classification of issues in the pre-negotiation stage is to proceed effectively. Specifically, the time available (or the *deadline* by which the negotiation must end) and the time required (or the *duration* of the negotiation) should both be factored into the process of classifying issues in order to minimise the risk of missing the deadline.

6.2.1 Preliminary Assumptions

In order to begin the development of our model, we must make some preliminary assumptions about the nature of issues. In particular, we must establish the relationship between different issues, specifically their *dependence* (or otherwise) on each other. Independent issues are those that can be settled without the need to consider the settlement of other issues. For example, when buying an airplane ticket it is possible to settle the issue of seat class (for example, business or economy) in isolation from the issue of flight routes (for example, direct or indirect). Conversely, dependent issues are those for which the settlements *are* affected by the settlement of other issues. For example, price often depends on other characteristics of the good being bought: the price paid for an airplane ticket depends on other issues such as seat class, flight routes, and so on. In general, negotiations with independent issues are simpler to represent and reason about, since each issue can be treated in isolation.

In this thesis, we limit ourselves to dealing only with independent issues that must be settled sequentially. This simplifies the presentation of our model and enables us to focus on the more central concerns of adjusting issue classifications in the context of time constraints. However, we recognise that other, more complex, scenarios can be considered, but leave that to future work.

6.2.2 Time, Deadlines and Negotiation Duration

In understanding negotiation deadlines and duration, the central concept is *time*, yet modelling time can be achieved in many different ways. Time can be considered as a continuous interval between some starting point and some end point extending into the future to infinity. Such a representation has the benefits of being flexible, but increases the complexity of any subsequent development of a computational model. Since our only requirement is that we can make some distinction about points in time and durations between any two points in time, we are able to adopt a simplified representation, in which time points are represented as integers; that is, we discretise time into unique points. The present moment is, therefore, represented as 0, and time points extending into the future are represented as integers that extend to infinity, i.e. $[0,1,2...\infty]$.

In terms of the formal representation of time, we use the non-negative integers.

$$Time == \mathbb{N}_1$$

We can now define both *deadline* and *expected duration* as they relate to a specific negotiation instance. This is shown in the *NegotiationTimeFactors* schema, which includes the *Negotiation* schema, the deadline within which the goal must be satisfied, and the expected duration of the negotiation.

NegotiationTimeFactors
Negotiation
$deadline : Time$
$expectedduration : Time$

6.2.3 Issues and Negotiation Duration

Recall that negotiations are comprised of bilateral issue combinations of the unilateral issue classifications of participants. As briefly discussed above, only *competitive* issue combinations have consequences for the duration of negotiation, since only they *require* negotiation. When determining the duration of a negotiation, therefore, it is these types of issues that we must consider.

Competitive issues are defined as those for which there is a *zone of agreement* (ZoA) (representing the space of acceptable settlements common to both participants) that contains more than one potential settlement. Agreeing on a settlement thus involves a search through the space of settlements for one that is mutually acceptable. Clearly, therefore, the size of this space can affect the time taken to find such a settlement. Although ultimately this depends on bargaining strategies and tactics (which we we do not address), the size of the ZoA provides a heuristic means to determine time to find a mutually agreeable settlement. Thus, in general, the larger the ZoA for an issue, the more time it will take to find a settlement agreeable to both participants. In the worst case, the time needed to find an agreement is that required to propose and consider every settlement in the set.

Whichever method is adopted is in some sense irrelevant, since both are proportional to the size of the set, and we can abstract out the detail simply by providing an axiomatic function, *expduration*, which takes a set of competitive bilateral issues and returns the expected duration of the negotiation.

$$expduration : \mathbb{P}\, Competitive \rightarrow Time$$

We can now state the relationship between a negotiation's deadline and its expected duration. The *ExceedDeadline* schema describes the condition that holds for negotiations that exceed their deadline, which is simply those in which the duration is greater than the deadline. Negotiations that do not exceed their deadline can be defined similarly.

ExceedDeadline

NegotiationTimeFactors

$expectedduration = expduration\ competitive$

$expectedduration > deadline$

This model of negotiations and their deadlines, though simple, provides us with enough detail to reason about the consequences of issue classifications on the ability to complete a negotiation within deadlines. In what follows we describe two mechanisms by which the buyer can manipulate the classifications given to issues, either by influencing the seller to make such re-classifications, or by doing so itself. We begin by describing how sellers can be influenced into re-classifying issues by increasing the amount the buyer is willing to pay to have its goal satisfied. Then, we examine the method by which the buyer can make such re-classifications itself.

6.3 Costing Out Issues

Payment-based negotiations involve the exchange of resources, typically money, in return for satisfaction of a goal. Such negotiations do not preclude the passing backwards and forwards of offers over issues other than price, but by increasing the amount paid, better settlements for other issues can be obtained. This technique is known as *costing-out* issues [116], and involves reasoning about how much a given settlement of an issue is worth in terms of price.

For example, when considering buying an airplane ticket, we may choose to pay more to obtain a desired seat class (for example business, or first class). To do so, we must understand how much the desired seat class is worth in terms of monetary value. By costing-out issues in such a way, more resources are used, but negotiation duration is shortened, since issues that are costed-out in such a manner are effectively re-classified by the seller as slack, which enables the buyer to obtain its most preferred settlement without the need to negotiate. Note, however, that since we are dealing only with pre-negotiation, we focus only on how the buyer can *identify* the extra resources (in this case, money) that it can commit to a negotiation in order that costing out can proceed. That is, we do not consider how the process of advancing extra resources is conducted during the negotiation itself, since this is not a part of pre-negotiation.

In order to be able to reason about the costing-out of issues in a principled way, we must consider the *worth* of issues in terms of resources. In turn, this requires determining how much resource an issue would be worth if it were to be costed-out. In the following sections we tackle these issues by first introducing our resource model and describing how it allows agents to dynamically evaluate their worth depending on availability. We then describe how the worth of both resources and goals is used in reasoning about the costing-out of issues which, in turn, affects the amount of resources that can be committed to a given negotiation.

6.3.1 Resources

Resources are important in our model because negotiation is a resource-constrained activity. For those negotiations that involve the exchange of payments, *how much* payment an agent can afford depends upon the value it places on its monetary resource. Similarly, the same applies to the agent receiving payment. How much payment it finds acceptable in return for satisfying the negotiation goal is determined by how much the agent needs the monetary resource and how much cost is involved in satisfying the goal.

Many different types of resources could be considered, but two common examples are energy and money. However, we can abstract out much unnecessary detail by simply declaring that resources are things an agent possesses whose *quantity* changes when certain actions are performed. Thus, the main property of resources that we need to model is that they are represented as *countable quantities*. For example, money can be represented in a currency of, say, unit quantities of \$1, and energy has unit-quantities of calorific units, and so on. Thus, an agent may possess a higher or lower quantity of resource. To represent this formally, we define the type *Quantity* to be a positive natural.

$$Quantity == \mathbb{N}_1$$

Formally, we define a resource as a set of *symbolic attributes* describing the properties of the resource, together with its quantity, in the *Resource* schema below.

Resource

properties : $\mathbb{P}$ *SymAttribute*
quantity : *Quantity*

6.3.2 Resource Availability and Worth

In negotiations that involve the transfer of resources, such as paying for a good or service, it is necessary to understand how much resource can be committed. However, using resources reduces the quantities available and, as resources are depleted, it becomes important to focus more on optimising their use. For example, committing an amount of money to a negotiation should take into account the current availability of money, since if money is scarce, it is prudent to commit less than if it were abundant. This requires the dynamic evaluation of resources so that, for different quantities of resource, different evaluations can be placed on them. Before considering how to determine how much resource to commit to a negotiation, therefore, we examine the dynamic evaluation of resources.

It is most generally the case that agents have access only to limited amounts of resources and, as a consequence, the use of those resources must be managed efficiently if an agent is to be effective in its role. In the real world, the scarcity of a resource tends to increase its value; for example, the price of oil can increase dramatically if production is interrupted by war or natural catastrophe. Thus, when resources are low, the constraints imposed upon their use are increased but, in situation when resource levels are high, these constraints can be decreased so that other priorities are able to take precedence.

In what follows, we present a resource model that enables agents to assess the worth of their resources based on the amount they possess. This allows the dynamic determination of reservation settlements for price, in the context of negotiation goals and costed-out issues.

First, we associate to every resource a *base unit worth* that represents the standard worth of one unit of the resource. Then, as the quantity of a resource available to an agent changes over time, the worth of the resource should also change in relation to how much an agent currently has access to. Of course, there are many ways in which this might be achieved. Figure 6.2 shows an example for a monetary resource composed of \$1 units, for which we give an example base unit worth of 0.5. The line represents the *worth modification curve* that is used to determine the current modification of unit

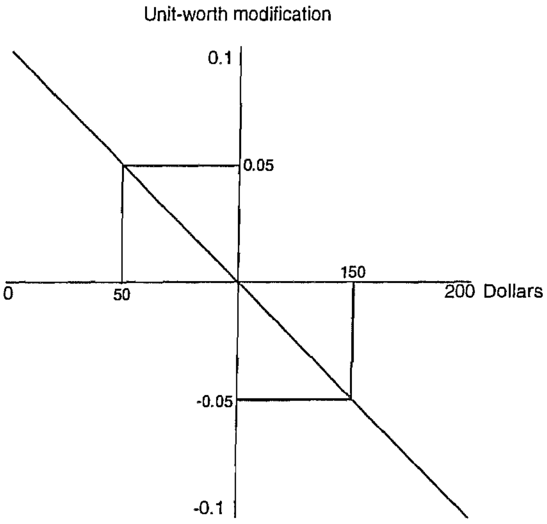


FIGURE 6.2: An example base unit-worth modifier function

worth depending on the amount of resource that is available. Note that many functions could be used to model different worth modification curves of varying complexity. Much research has been done in animal behaviour examining the relationship between resources and their use in satisfying different needs (e.g. [89, 140]). However, the worth of the resource is evaluated in relation to its availability, and the analysis can become very complex [90]; since we are not concerned with exploring these technicalities, it is sufficient to adopt a simple linear function:

$$y = 1.10^{-3}x + 0.1$$

In the figure we can see that if an agent has \$100 (represented by the origin of the graph), then the base unit-worth of \$1 is unchanged. However, if the agent's money falls to \$50, then the base unit-worth of \$1 is increased by 0.05 to 0.55 (as indicated by reading off the y-axis at the point of \$50 on the x-axis). Conversely, if the available money increases to \$150 then the worth is lowered by decreasing the base unit-worth by 0.05 to 0.45.

Formally, the above intuitions are modelled as follows. First we provide three functions: *baseunitworth*, *resourceworthmodifier* and *currentresourceworth*. The first function simply returns the base-unit worth of a resource. The second, *resourceworthmodifier*, modifies the base unit worth based on how much resource the agent currently has access to, as described above. We provide a predicate that specifies how the modification is calculated using the function shown in Figure 6.2. Finally, we provide another function, *currentresourceworth*, which takes a resource and returns its *current worth*, calculated by simply adding the value returned by the *resourceworthmodifier* function to the base unit-worth of the resource.

$$\begin{array}{|l}
baseunitworth : Resource \rightarrow Worth \\
resourceworthmodifier : Resource \rightarrow Worth \\
currentresourceworth : Resource \rightarrow Worth \\
\hline
\forall\, r : Resource;\ w : Worth \bullet w = 1.10^{-3} \times r.quantity + 0.1 \\
\forall\, r : resource \bullet currentresourceworth\ r = \\
\qquad\qquad (baseunitworth\ r + resourceworthmodifier\ r)
\end{array}$$

Having defined these functions to enable the dynamic evaluation of resource worth, we can now describe the costing-out process that effectively allows the buyer to *buy* slack issues from the seller.

6.3.3 Buying Slack Issues

Resources have worth to an agent in the same way that satisfying goals has worth and, since negotiation involves the exchange of resources in return for the achievement of a goal, it is important that there is an overall gain (or, at minimum, no loss) in worth. (Note that in cases where an agent seemingly accepts an overall loss in worth for reasons of strengthening a relationship with an opponent, the worth gained in strengthening the relationship must in fact balance out the worth lost by accepting a less satisfactory settlement.) The amount of resource committed to a negotiation must therefore not have a worth that exceeds the worth of the negotiation goal.

The key question here is how much resource to use in this situation. As described above, costing-out issues simply involves paying extra money (or other resource) to influence a seller into pruning its own set of negotiable issues by reclassifying them as slack. Since the corresponding issue for the buyer becomes fixed, thus instantiating it to the aspiration settlement, the gain in worth can be easily determined as the difference in worth between the original goal and the new goal (arising from the newly created fixed issue). The buyer can then determine how much resource to commit to costing-out the issue, proportional to the gain in goal worth that would arise.

To express this formally, we require a function that returns the worth of an issue if it is settled with the aspiration settlement. The *aspirationsettlement* function below takes a negotiable issue as argument and returns the worth that results from settling the issue in such a manner.

$$aspirationsettlementworth : NegotiableIssue \rightarrow Worth$$

We can now take the worth gained from an issue with the aspiration settlement and determine the total worth to be gained from the negotiation goal, as specified in the *AdjustGoalWorth* schema.

$$\begin{array}{|l}
\_\_ AdjustGoalWorth \_\_\_\_\_\_\_\_ \\
\Delta NegGoalWorth \\
i? : NegotiableIssue \\
\hline
i? \in negotiableissues \\
totalgoalworth' = totalgoalworth + aspirationsettlementworth\ i? \\
\hline
\end{array}$$

It is now straightforward to calculate the *quantity* of resource that can be used to have a goal satisfied through negotiation, which is the amount of resource with worth equal to the worth gained from the goal's satisfaction. This quantity of resource is calculated by taking the quotient of the worth of the goal and the current unit worth of the resource. So, for example, if a goal has a worth of 10 and the current unit worth of the resource is 0.2, then the quantity of resource that can be committed to a negotiation for that goal is 50.

This method of calculating how much resource to commit to a negotiation is shown formally in the schema, *CommitResource*, which includes the *NegGoalWorth* schema, a resource and a quantity of resource to be committed, *commitamount*. The predicate part of the schema then specifies the calculation, as described above.

$$\begin{array}{|l}
\_\_ CommitResource \_\_\_\_\_\_\_\_ \\
NegGoalWorth \\
r : Resource \\
commitamount : RAT \\
\hline
commitamount = (totalgoalworth / currentresourceworth\ r) \\
\hline
\end{array}$$

We have shown how buyers can dynamically evaluate the worth of their resources to commit the right amount of those resources to a negotiation. This ensures that a buyer is not disadvantaged by overcommitting resources, and incurring an overall loss of worth. Clearly, however, costing-out issues is subjective and determined in relation to the buyer's evaluation of issues; it does not guarantee that issues can actually be bought in exchange for the resource identified, since the seller may have a different evaluation and may be unwilling to agree to make the issues slack.

6.4 Unilateral Issue Pruning

Not all negotiations involve the exchange of resources, but instead involve attempts to reach agreement on specific outcomes or courses of action. Consider Bob and Alice trying to come to an agreement over how they will travel to a meeting. Bob wants to travel either by taxi or train, but Alice wants to travel by bus. Here, although it is possible that Alice could introduce a resource-based incentive to persuade Bob to accept her proposal that they both travel by bus, it is more likely that they will haggle over the various options and try to reach an agreement, without exchanging resources.

In these kinds of negotiations when time is short, costing out issues is not a viable course of action, and other ways to reduce the duration of the negotiation must be considered. Our approach is to enable the buyer to consider the various issues and make selective choices over which issues to drop or *prune* from the negotiation. Thus, in the case of Bob and Alice's travel plans, if they must reach agreement quickly, Alice can decide to *prune* the issue regarding the method of transport, allowing Bob to have the final say on how the issue is to be settled.

In terms of our model, Alice effectively re-classifies the method of travel from a negotiable issue to a slack issue, allowing Bob to similarly re-classify the issue from negotiable to fixed. Bob can now settle the issue to his aspiration settlement, since Alice has indicated that she is indifferent to any settlement.

6.4.1 Issues For Pruning

Determining which issues to prune in the above manner is straightforward when considering the expected duration of issues. Earlier we described how the expected duration of an issue is calculated by the size of the settlement set, or the ZoA. In situations where time is short, therefore, it seems that those issues that can be expected to take the longest to settle should be the candidates for pruning. However, this may not always be the best course of action since, if we only examine the expected duration of an issue, other important factors are ignored, such as the *worth* of the issue to the buyer. Recall that issue-values of an issue are scored in relation to their effects on existing goals those that conflict with or hinder existing goals receive negative scores to reflect the loss of worth that would result if they were to be used as settlements. If we ignore the effects of such settlements on existing goals when considering which issues to prune, we risk incurring a loss of worth if the seller subsequently fixes the issue to a settlement that conflicts with, or hinders, one or more of the buyer's existing goals. For these reasons we focus in addition on examining the *worth* of issues when considering which should be pruned from a negotiation. If we assume that each issue has the same expected duration, it is easy to establish the number of issues that must be pruned. This is the approach we adopt here. (However, if we cannot assume such regularity of issue duration, we need not determine the quantity, but can instead simply apply the procedure described in the next section to prune issues repeatedly until we reach the desired time reduction.)

Initially, therefore, we must determine how many issues require pruning based on their expected durations and by how much they exceed the given deadline. Identifying this quantity is achieved in the *PruningQuantity* schema, which examines a negotiation's time factors and expected issue durations. The variable, *overshoot*, is the amount by which the negotiation is expected to exceed its deadline and *prunequantity* is the number of issues that require pruning. Overshoot is calculated simply as the difference between the deadline and expected duration of the negotiation, and the *numberToPrune* function takes the overshoot value and the issues of the negotiation and calculates how many must be pruned.

PruningQuantity

$ExceedDeadline$

$prunequantity : \mathbb{N}$

$overshoot : Time$

$numberToPrune : Time \rightarrow \mathbb{P}\, NegotiableIssue \rightarrow \mathbb{N}$

$overshoot = deadline - expectedduration$

$prunequantitiy = numberToPrune\ overshoot\ negotiableissues$

6.4.2 Worth-Based Pruning of Issues

Once the number of issues that require pruning is determined, we must establish which of the negotiable issues should be pruned. This is simply achieved by removing those issues whose combined worth loss to the buyer is minimal. First, however, we must determine the expected worths of the negotiable issues. This can be achieved in a variety of ways such as, for each issue, taking the settlement with the highest positive *score* or, as there is uncertainty over what settlement will be eventually agreed, taking an average of the scores of all issue-values in the active range of the issue. Whichever method is used is unimportant for our concerns so long as there is a way to trim the set of negotiable issues by excluding those that ultimately offer the least worth to the buyer.

To determine the *expected worth* of a set of negotiable issues without committing to a particular method, we introduce a function, *expectedissueworth*, which simply returns a value that represents the expected worth of a set of issues.

$$expectedissueworth : \mathbb{P}\, NegotiableIssue \rightarrow Worth$$

We can now identify the set of issues that are to be pruned, as shown by the *PruneSet* schema. The set of issues pruned are those whose expected durations are enough to lower the overall duration of the negotiation under its deadline, and are, at the same time, those whose expected worth is lower than any other set of issues in the set of negotiable issues.

$$\begin{array}{l}
\textit{PruneSet} \\
\hline
PruningQuantity \\
pruningissues : \mathbb{P}\, NegotiableIssue \\
\hline
\forall nis : \mathbb{P}\, NegotiableIssue \mid nis \subseteq negotiableissues \\
pruningissues = nis \Leftrightarrow expectedNegIssueDuration\ nis \geq overshoot \land \\
\qquad (\forall nis1 : \mathbb{P}\, NegotableIssue \mid nis1 \subseteq negotiableissues \land \\
\qquad nis1 \neq nis) \bullet expectedissueworth\ nis < expectedissueworth\ nis1 \\
\hline
\end{array}$$

6.4.3 Pruning Issues of Low Worth

Once the set of prunable issues has been identified, they can be discarded so that the total number of negotiable issues is reduced, thereby offering an increased chance of concluding the negotiation within the time constraints.

As an example, consider Table 6.1, which shows three issues from which we determine that two must be pruned. Examining the pairwise combinations of worths of the three issues it is clear that the `departure_airport` and the `direct_flight` issues provide the minimal loss of worth, so are selected as the two issues to prune. This

TABLE 6.1: Pruning issues that offer the minimal loss of worth

Negotiable Issue	**Expected worth**	**Slack**	**Negotiable**
`airline_company`	7		✓
`departure_airport`	5	✓	
`direct_flight`	6	✓	

transforms their status from negotiable to slack as indicated by the ✓ symbols in the slack column.

Once a negotiable issue is pruned, it must be converted to a slack issue. This is achieved in the *BuyerSidePrune* operation schema, which alters the state of a negotiation goal by reclassifying negotiable issues as slack. The schema takes as input a set of *negotiable issues*, which must be within both the buyer's set of negotiable issues for the goal and the set of bilateral competitive issues. The *ExceedDeadline* schema is also included to indicate that this operation can only be used in this situation. The issues are made slack using the axiomatically defined function, *makenegslack*, added to the set of *slack issues*, and removed from the set of *negotiable issues*. The sets of congruent and competitive issues are modified similarly to be consistent. Finally, the expected duration of the negotiation is also amended to take into account the new issue classifications.

$$makenegslack : \mathbb{P}\, NegotiableIssue \rightarrow \mathbb{P}\, SlackIssue$$

BuyerSidePrune

$$\begin{array}{l} PruningCandidates \\ is? : \mathbb{P}\, NegotiableIssue \\ ExceedDeadline \\ \hline is? \subseteq negotiableissues \wedge is? \subseteq competitive \\ slackissues' = slackissues \cup \{makenegslack\ is?\} \\ negotiableissues' = negotiableissues \setminus \{is?\} \\ congruent' = congruent \cup \{makenegslack\ is?\} \\ competitiveissues' = competitiveissues \setminus \{is?\} \\ duration' = expduration\ negotiableissues' \end{array}$$

6.5 Opportunities for Negotiation

When there are many opportunities for negotiation, in that there are several sellers to negotiate with, failure to reach an agreement in one negotiation does not prevent success in others. In these situations, we can be more demanding over what settlements are considered acceptable. If there are few negotiation opportunities, however, then we must try to ensure that a given negotiation has a higher chance of succeeding; to do this we need greater flexibility to consider negotiation settlements that might otherwise be rejected. Figure 6.3, in which the rectangle represents the space of possible settlements for a negotiable issue, shows how the opportunity for negotiation affects considerations of possible settlements. If there are few opportunities for negotiation, then a larger part of the settlement space should be considered.

Previously, we have shown how agents can prune issues to speed up the process of reaching an agreement when time constraints are important. Effectively, the process of pruning an issue alters issue preferences so that the issue becomes slack and there is maximal flexibility over how it may be settled. However, we may instead want to *gradually* increase issue flexibility in response to the *consequences* and *likelihood* of failing to reach an agreement.

The consequences of failing to reach an agreement relate to the level of worth expected from the negotiation goal, so that goals with higher levels of worth incur more severe losses if a negotiation fails. Conversely, the likelihood of a goal failing is related to the number of opportunities to negotiate its satisfaction; when there are few opportunities, the risk that the goal will remain unsatisfied increases.

To increase flexibility for an issue, there are two problems to solve. First, we must specify how to identify the issues that require more flexibility by establishing the existing level of flexibility of issues and determining if a change is necessary. In order to do this we must examine both the amount of opportunity that exists for negotiation and the worth of the negotiation goal, since we must ensure that the increase in flexibility over issues satisfies the need to avoid failure in the context of the worth loss that may arise. Second, once a need for an increase in flexibility is identified, we must quantify the degree of flexibility change that is required. This relates again to the degree of opportunity; in situations where opportunity is low, more flexibility over issues is required than in situations where opportunity is higher.

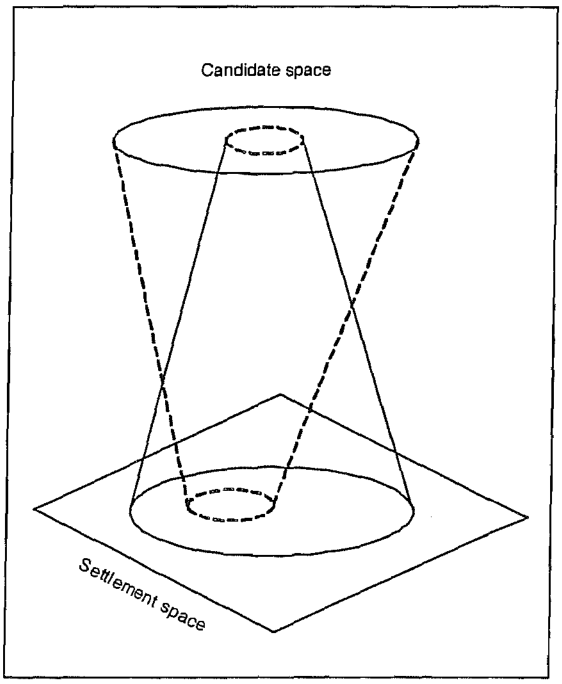


FIGURE 6.3: Negotiation opportunities and their effect on consideration of the space of negotiated agreements

6.5.1 Available Negotiation Opponents

When an agent needs to negotiate, it must attempt to discover which other agents are able and willing to enter into negotiation. Normally, this is achieved by requesting participation of agents offering the required service, with those agreeing to negotiate forming the candidate set of sellers. Since we are not interested in the mechanics of determining this set (which can be achieved through standard methods such as a call for bids), we simply provide the *responseset* function which, given a negotiation goal and a set of agents, returns those sellers that are willing to negotiate over the satisfaction of the goal.

$$responseset : NegotiationGoal \rightarrow \mathbb{P} Agent \rightarrow \mathbb{P} Agent$$

The constraints that the degree of opportunity for negotiation imposes on a particular goal are given in the *NegOpportunityConstraints* schema below, which contains a negotiation goal and variable representing the *opportunity* that exists for negotiation. The predicate part simply states that *opportunity* is equal to the size of the set of agents returned by the *responseset* function.

NegOpportunityConstraints

$$\begin{array}{l} ng : NegotiationGoal \\ opportunity : \mathbb{N}_1 \\ \hline \forall ng : NegotiationGoal; \ ags : \mathbb{P} Agent \bullet \\ opportunity = \#responseset \ ng \ ags \end{array}$$

6.5.2 Goal Worth and Within-Issue Flexibility

For goals with high levels of worth, greater degrees of within-issue flexibility should be adopted, and *vice-versa* for goals with low levels of worth. Flexibility over an issue is determined by the proportion of issue-values in the issue's *range* that are also in its *active range*. For example, when considering the different airports from which to take a flight, for which the possibilities include all UK airports, we might limit the active range to the subset of London airports. This provides us with a degree of flexibility that would be reduced if we limited ourselves to just Gatwick and Heathrow, but would be increased if we were to include all airports in the southern region.

This is simply expressed as a function, *flexibilityscore*, which takes a negotiable issue and returns a value expressing the degree of flexibility expressed for the issue, given simply by the ratio of issue-values of the issue's *range* that are in also the *active range*.

$$\begin{array}{l} flexibilityscore : NegotiableIssue \rightarrow RAT_0^1 \\ \hline \forall i : NegotiableIssue; \ q : RAT_0^1 \bullet \ q = \#i.activerange / \#i.range \end{array}$$

Now, to determine if we need to increase flexibility to increase the chances of coming to an agreement in time (given the worth of the goal and the degree of opportunity for its satisfaction), we define a *flexibilitythreshold*, below which the flexibility score of an issue is regarded as sufficiently inflexible that it requires adjustment. The value for the threshold is itself determined in relation to the worth of the goal so that, for goals with high worth, the threshold is set high and vice-versa for goals with low worth. This is defined formally by the *flexibilitythreshold* function, which takes the worth of a negotiation goal and returns the *flexibilitythreshold* value that is then used to determine an issue's flexibility status. Instantiating such a function is domain-dependent, and many are possible.

$$flexibilitythreshold : NegGoalWorth \rightarrow RAT_0^1$$

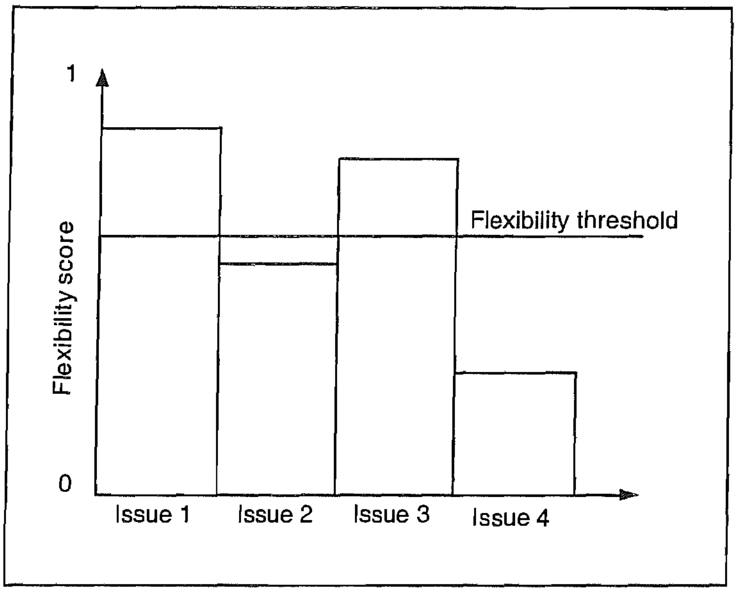


FIGURE 6.4: Issue flexibility

TABLE 6.2: Flexible and inflexible issues

Issue	Flexible	Inflexible
Issue 1	✓	
Issue 2		✓
Issue 3	✓	
Issue 4		✓

To make this clear, consider Figure 6.4, in which we can see four issues, each given a flexibility score (on the y-axis) reflecting the number of issue-values in their range that have been placed by the buyer in the active range. The flexibility threshold is shown as the horizontal line. Those issues whose flexibility falls under the threshold are considered sufficiently inflexible. Table 6.2 gives each issue and its flexibility status.

With the two functions above, we can determine which issues of the negotiation goal are too constrained in relation to the worth of the goal. We show how this is achieved formally in the *InflexibleIssues* schema, which incorporates the *NegGoalWorth* schema and a set of negotiable issues, *inflexibleissues*, which are too constrained, and for which we must consider expanding the set of acceptable settlements to avoid failure. The predicate part states that every issue in the set of negotiable issues with a flexibility score less than the flexibility threshold is placed in the set of inflexible issues.

InflexibleIssues

$\Delta NegGoalWorth$

$inflexibleissues : \mathbb{P}\, NegotiableIssue$

$$inflexibleissues = \{i : NegotiableIssue \mid i \in ng.negotiableissues \bullet \\ flexibilityscore\ i < flexibilitythreshold\ ng\}$$

6.5.3 Expanding the Set of Acceptable Settlements

Now that we have established how to determine when issues are too constrained given the worth of the current goal, we can elaborate the process by which the *degree* of increase to flexibility is determined.

When a negotiation goal is initially generated, the issue-values in the active range are those that give positive benefits to the existing goals of the agent. This ensures that only those beneficial settlements are considered as outcomes of the negotiation. When opportunity for negotiation is low and goal worth is high, we can expand the set of acceptable issue-values to increase the chance of agreement. However, this must be done in a manner that does not negate the worth gained from satisfying the goal. For example, suppose Bob is travelling to Barcelona for a holiday, and wants to stay in a hotel close to the beach. Unfortunately, Bob has left his booking very late in the season, and discovers that there are not many rooms available at this time. In order to increase his chance of obtaining a holiday, Bob reconsiders his options and decides that he will also consider hotels that are near easy travel links to the beach, but will not consider those near the airport because of noise.

In this way, when faced with a limited number of options to obtain the satisfaction of a goal, we are sometimes forced to expand our set of preferences to increase the possibilities for goal satisfaction. However, in general, there are limits to the increase in flexibility that is possible, specifically if this permits settlements that entail a loss in worth greater than the worth gained by the goal's satisfaction.

In terms of our model, we must examine issues and their potential settlements to determine which can be incorporated into a new set of preferences. In particular, for each issue, we need to measure the worth loss of each unacceptable issue-value that is not in the active range (that is, those with a negative worth score) in relation to the overall worth gain of the negotiation goal. Then, those issue-values with a worth loss less than the overall gain arising from the goal can be added to the issue's active range.

This is described formally in the *AddPossibleSettlements* schema, which takes a negotiable issue as input and adds an issue-value to the active range of that issue if and only if its *worthscore* is less than the total worth of the goal.

$\mathit{AddPossibleSettlements}$

$\mathit{i?} : \mathit{NegotiableIssue}$
$\mathit{InflexibleIssues}$
$\mathit{NegOpportunityConstraints}$

$\mathit{i?} \in \mathit{inflexibleissues}$
$\forall \mathit{iv} : \mathit{IssValue} \mid \mathit{iv} \in (\mathit{i?.range} \setminus \mathit{i?.activerange}) \bullet$
$\quad \mathit{i?.activerange}' = \mathit{i?.activerange} \cup \{\mathit{iv}\} \Leftrightarrow$
$\quad -(\mathit{elementscore}\ \mathit{iv}) \leq \mathit{totalgoalworth}$

6.6 Empirical Evaluation

In order to test our model of pre-negotiation decision-making we have developed an empirical testbed, which allows us to assess the effectiveness of a buyer attempting to select optimal sellers for negotiation. Our main focus is on how buyers can ensure that negotiations do not exceed given deadlines, which is achieved by the selective pruning of issues that are expected to increase the duration of the negotiation past its deadline.

Specifically, the testbed implements the goal model defined in Chapter 4, and performs the bilateral classification analysis described in Chapter 5. For the purpose of our experiments, we do not need to examine the effects of fixed issues, since they do not impact on pruning, and we can therefore restrict our analysis to negotiable and slack issues.

Each experiment involves one buyer and a population of sellers. The buyer has a goal that requires satisfaction, for which it must find a suitable seller to negotiate with. Each negotiation episode of the experiment involves the buyer generating a goal, finding a negotiation opponent and negotiating with that opponent. To ensure our results are robust, we simulate repeated negotiation episodes with different parameters. The complete set of repeated episodes is an experimental run; different runs are conducted to test different strategies or characteristics of our pre-negotiation model, to provide a comparative evaluation. Note that each run uses the same randomly generated values that provide variability in the behaviours of the agents to ensure that the results across strategies or characteristics are indeed appropriate to compare. A complete experiment involves a set of such different runs to investigate the effects of these different properties on the ability of the buyer agent to engage in successful negotiations.

6.6.1 Control Parameters

At the start of each experiment, several control parameters are fixed, and some of which are automatically initialised, some of which can be set by the user. These are as follows.

- The goal over which agents negotiate is composed of a predetermined number of issue attributes, each of which can be settled in one of 10 ways. The number of issues in a goal is is set by the user.
- Each goal is assigned a random deadline in a restricted interval.
- Each agent has a preference model that determines the kinds of issue classifications made and the preferences adopted for the settlement of issues. Preference models can be parameterised to make the agent more competitive or cooperative and are specified by the user. Agents can be made to have completely fixed preferences or to display preferences that vary probabilistically.
- The buyer models each seller's preferences over issue classifications and settlement preferences, as described in Chapter 5. The buyer's model of the sellers can be more or less accurate, as specified by the user; to ensure that we use sensible models, assuming some variability in the accuracy of these models and introducing some noise at the same time, in our experiments, the buyer's model of sellers assumes a normal distribution.

6.6.2 Experimental Process

During each negotiation episode in an experiment, several different processes are undertaken, in two parts. In the first part, issues are classified by all agents, as follows. First, as shown in Algorithm 1, a goal is generated that contains a pre-determined number of issue attributes, each of which is associated with ten different settlements. For each agent, the issues of the goal are classified into either slack or negotiable issues according to preferences. Then, for each issue that is classified as negotiable, a subset of the associated settlements are selected as acceptable settlements and assigned a random worth value in the interval [0,1]. The first part ends by ordering the set of acceptable settlements according to worth, with the last settlement in the set becoming the aspiration settlement and the first becoming the reservation settlement.

Algorithm 1 Issue Classification

Inputs: goals, agents

Algorithm:

```
g = generateGoal(goals)
for all a ∈ agents do
    issues = classifyIssues(g)
    for all i ∈ issues do
        if i = Negotiable then
            acceptablesettlements = selectSettlements(i)
        else if i = Slack then
            acceptablesettlements = i.range
        end if
        orderSettlements(assignWorth(settlements))
    end for
end for
```

In the second part, shown in Algorithm 2, the buyer chooses a seller and negotiates with it. First, the buyer examines its models of seller preferences and forms expectations about their issue classifications and settlement selections. The buyer then compares the expected preferences of each seller against its own preferences and determines the likely bilateral issues that will arise. Given this analysis, the buyer ranks sellers and selects the highest ranked; sellers with which competitive issues are minimised and congruent issues are maximised are ranked more highly. If the duration of the negotiation with the selected seller is expected to exceed the deadline, the buyer prunes its issues. Finally, when these steps are completed we simulate negotiation. This is done by taking a system-level view and comparing the buyer's preferences against the *actual* preferences of the seller. Then, where possible, settlements that *maximise the product* of the worths gained by each agent are selected. Thus, we do not consider the effects that negotiation strategies might have on the outcome of the negotiation since our concern is to ensure that each negotiator is equally capable and does not have strategic advantages over the opponent. This ensures that the experimental results are due only to issue classifications and not strategy interactions.

6.6.3 Seller Population and Buyer Success

The first set of experiments examines the effects of the number of sellers in the environment on the ability of the buyer to engage in successful negotiations. We consider the

Algorithm 2 Opponent Selection

```
Inputs: goal, sellers, prunestrategy
Algorithm:
for all $s \in sellers$ do
    $s.class = determineExpectedIssueClassifications(goal, s)$
    $s.bilaterals = determineBilaterals(goal, class)$
end for
$partner = selectNegPartner(ranksellers(sellers))$
if $determineExpDuration(partner, goal, partner.bilaterals) > goal.deadline$ then
    $applyPrune(prunestrategy, goal)$
    $negotiate(partner, goal)$
end if
```

effects of the seller population on four different negotiation partner selection strategies, as follows.

Info/no prune The buyer uses its models of seller preferences to make selections, but no pruning of issues is performed.

Prune duration The buyer makes its selection as above, but for negotiations that are expected to exceed their deadlines the buyer selectively prunes issues according to their expected duration.

Prune worth The buyer makes selections as above, but prunes issues according to their expected worth.

No info/no prune The buyer selects sellers randomly and performs no pruning.

In these experiments, each seller is equally likely to classify issues as negotiable or slack, but the buyer has competitive preferences so that it has a tendency to select more negotiable issues than slack issues. Figure 6.5 shows the results for the four selection strategies described above for different numbers of sellers. Since the buyer does not learn over time, we present the results as a bar chart where each run is distinct and not comparable. Thus, we are only concerned with comparing the strategies within each run and not between runs. We can observe that the *pruning* strategies outperform the *information only* strategy which itself performs better than the *no information* strategy in all cases. This is what is expected since they combine the benefits of the selection strategy that uses the seller preference models to infer likely issue classifications, and can make further refinements due to pruning. The *Prune duration* strategy outperforms the *Prune worth* strategy in all but one run (the last run), which can be explained by the fact that simply pruning for worth cannot guarantee that the negotiation duration is reduced to the required degree, so may lead to more failed negotiations.

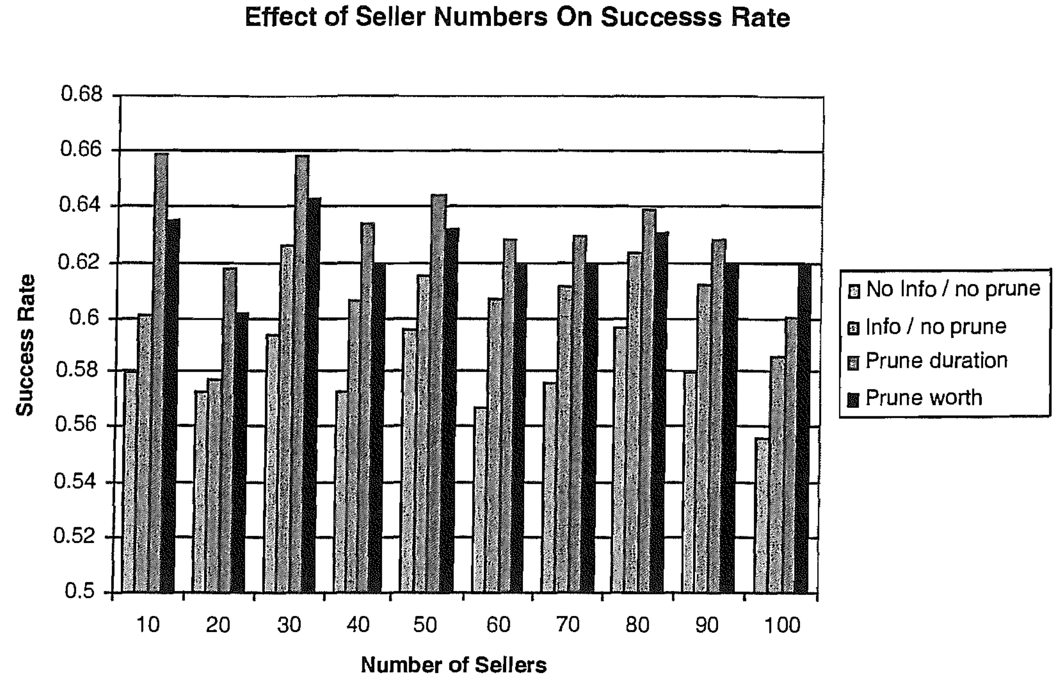


FIGURE 6.5: The effect of seller numbers on buyer success

6.6.4 Issue Quantity and Buyer Success

The number of issues that must be negotiated can be expected to affect performance of negotiation since, as the number of issues increase, the more difficult it will be to ensure that agreements can be found for all. In this experiment, we again consider the four selection strategies and compare their performance when we steadily increase the numbers of issues that must be negotiated. The population of sellers is fixed to 20, and all sellers are equally likely to classify issues as negotiable or slack. The buyer is again competitive, so that its own classifications of issues tend to be negotiable.

The results of this experiment are shown in Figure 6.6, and it is immediately clear that increasing the number of issues that require settlement greatly affects buyer success. Again, the *pruning strategies* outperform the others, but all strategies perform equally well when there are few issues. As the number of issues increases, however, the strategies differentiate, with the two pruning strategies remaining close in their level of performance.

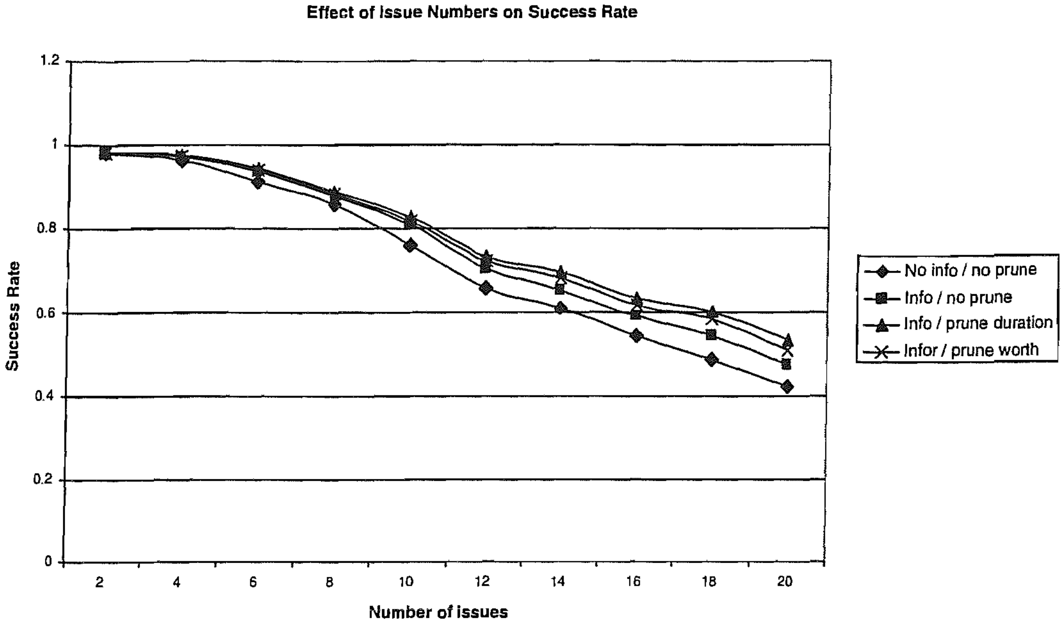


FIGURE 6.6: The effect of issue numbers on buyer success

6.6.5 Issue Quantity and Buyer Worth Gain

As we have seen above, the number of issues that require settlement can affect the ability of participants to successfully conclude a negotiation within time limits. As well as affecting the number of successful negotiations, this can also affect the amount of worth gained from negotiation. However, by using *pruning strategies*, we would expect, along with an increase in the number of successful negotiations, that the amount of worth gained would also be increased. To explore this, we examined the worth gained from negotiation as the number of issues increases from 2 to 20. Again, we use the same four selection strategies described above, and the same buyer and sellers. The results are shown in Figure 6.7, which raises several points to note. First, when the number of issues is low, all strategies give high amounts of extra worth, and the strategies using information about sellers all perform similarly. However, as the number of issues increases, the extra worth gained decreases, but at a lower rate for the pruning strategies in comparison to the *information only* strategy and the *no information* strategy. It can also be seen that the *prune for duration* strategy performs better than the *prune for worth* strategy. This is because the worth-based pruning strategy cannot always ensure that the required amount of time is saved and so more of these negotiations will fail. Note that the worth gain displayed in the figure is that amount of worth gained *over and*

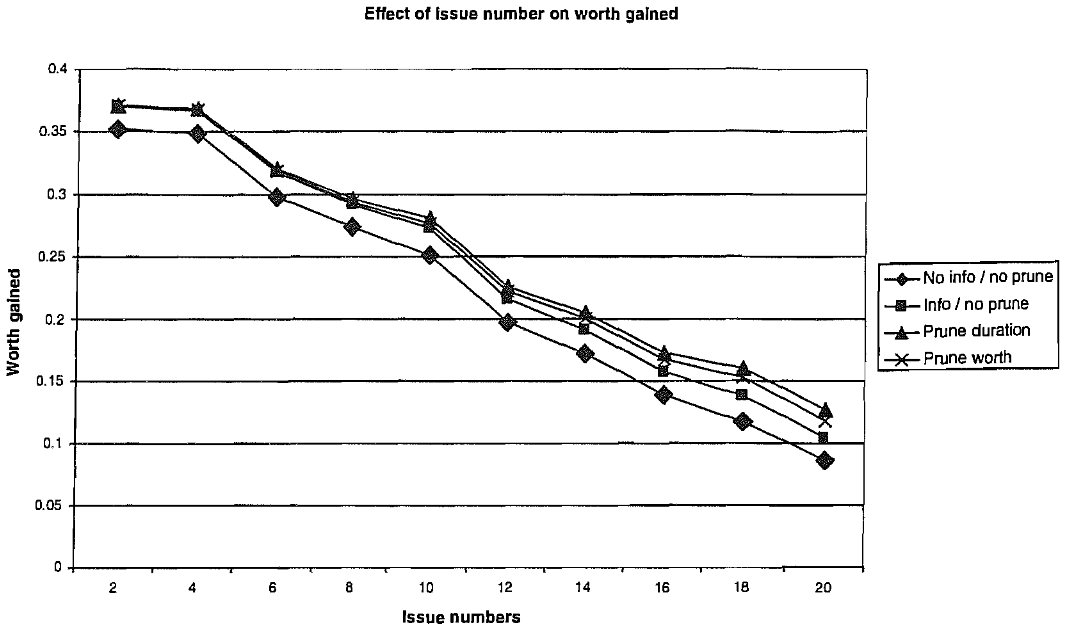


FIGURE 6.7: The effect of issue numbers on worth gained from negotiation

above the worth for the goal being negotiated, i.e. it is the extra worth gained from the benefits arising from the successfully settled issues.

6.6.6 Discussion

In the experiments reported above, we have tested the performance of a number of different negotiation selection strategies for a buyer agent. We have shown how the strategies that employ pruning can improve the chance that negotiations will be successful, by enabling agents to avoid exceeding deadlines associated with the negotiation goal. The pruning strategies perform well even when the number of sellers increases, and outperform other strategies when the number of negotiation issues is large. The disadvantage of using such strategies is in the worth that is lost from the issues that are pruned. Pruning issues transforms them from negotiable issues to slack issues, so that any worth that might have been gained by negotiating an acceptable settlement is lost in favour of the time gained (by not negotiating over the issue). However, since the worth given by issues is additional to the worth gained from the satisfaction of the goal, the loss of such worth is still beneficial if the negotiation ends within its time constraints.

TABLE 6.3: Comparison of pruning for worth and information only strategies.

Statistical Measure	Pruning (worth)	Information only
Mean	0.60654	0.62403
Variance	0.000240869	0.000133376
Observations	10	10
Hypothesised Mean Difference	0.0	
$P(T <= t)$	0.005214064	

6.6.7 Experimental Analysis

Our experiments involve examining the performance of the buyer in negotiations with different sellers. In particular, we examine how different *opponent selection strategies* affect the success of the buyer in engaging in successful negotiations. For each experimental run we validate the results by performing statistical analyses, our main concern being to show that the difference in performance produced by using the different strategies is statistically significant. To achieve this we use the *t-test: Two sample Assuming Equal Variance* analysis, which is applied to pairwise comparisons of each selection strategy. The data used by the test comprises the means observed for each strategy generated from each experimental run. Thus, to compare the *Prune worth* strategy against the *Info/no prune* strategy for different numbers of sellers, we compare the means generated by 100 negotiation episodes over 100 runs, which gives the analysis presented in Table 6.3.

For each experiment we use a significance level of 0.05 giving a confidence level of 95%. The important value to note is $P(T <= t)$ which, if under the significance value means that the result is statistically significant. We have analysed each experiment conducted, but present only those for the experiment concerned with the effect of seller numbers on the ability of the buyer to successfully conclude negotiations. Table 6.4 presents the $P(T <= t)$ values for a subset of pairwise comparison of the different strategies.

TABLE 6.4: Comparison of pruning for worth and information only strategies.

Strategy pair	$P(T <= t)$
No info / Info	0.000223874
Info / prune (duration)	0.000870899
Info / prune (worth)	0.005214064
Prune (worth) / prune (duration)	0.079908423

All strategy pairs, except for one, perform significantly differently. The exception is the two pruning strategies, which have have comparable performance.

6.7 Conclusion

Negotiation is typically a constrained activity. Constraints commonly exist on the amount of time available to reach an agreement, the amount of resources available to secure agreements on issues and the opportunities available to engage in negotiation. When considering entering into negotiation over some goal, it is important that these constraints are recognised and factored into the preparations made. Limited time demands a quick approach, in which case it may be necessary to focus on only those most important issues and discard the less important. Limited resources may hinder the ability to cost out issues and may require that the buyer simply drops those with less worth. Similarly, limits on negotiation opportunity affect the consideration of settlements and their perceived desirability.

In this chapter, we have described mechanisms that can adapt the negotiation stance taken in response to information about environmental constraints. We have shown how these constraints affect a buyer's classification of issues for negotiation and the flexibility it exhibits over issues. In the context of time constraints there are two ways that the buyer can decrease the expected duration of a negotiation. First, if the negotiation involves the exchange of resources, the buyer can attempt to cost out issues by increasing the amount it pays to the seller in return for the identification of more slack issues, and obtain a subsequent reduction in the duration of the negotiation. Second, when resource exchange is not possible or desirable, the buyer can choose to minimise the number of negotiable issues it negotiates over by selectively pruning issues of low worth to achieve the same reduction in duration.

We also examined how goal worth and the degree of opportunity for negotiation can be used by an agent to determine if and by how much a change in flexibility for negotiable issues is necessary. This approach helps to avoid deadlock by increasing the space of possible settlements from which agreements on issues can be found. In situations where there are few opportunities for negotiation, the buyer can take a more flexible stance, leading it to accept less than optimal settlements. This is reflected in the expansion of its preferences for settlements over the various issues.

We limit ourselves to the examination of independent, sequentially settled issues. This makes the analysis and presentation of the ideas simpler to follow. However, we recognise that other more complex situations can be considered. In particular, much recent work is investigating negotiations involving the concurrent settlement of issues [104], and useful extensions to our work could be developed by expanding our issue model to account for such types of issues.

In this chapter we also describe how our model of bilateral issue types can aid an agent in selecting suitable negotiation opponents. However, other work is also currently being undertaken to examine opponent selection from several perspectives.

First, opponent selection has been considered in relation to *trust* and *service reputation* (e.g. [127, 119]), where both refer to the fidelity of the opponent's behaviour with regard to the negotiated outcome. While trust and reputation are of great importance for designers of agent systems, especially systems characterised by openness they are, we argue, only part of the story, and must be supplemented with the kinds of considerations we examine here, such as the expectation of unilateral issue classifications by opponents, and the resultant bilateral issues.

Second, non trust-based opponent selection has also been addressed by a number of researchers. Work by Tesfatsion [145] examines how agents select opponents based upon the amount by which they exceed a fixed performance-based tolerance threshold. Though this work examines similar problems to those in this chapter, it does not address the specific problems of the minimisation of conflict through the selection of negotiation opponents, as we do through the notion of deadlocked issues and the risk that results. Banerjee *et al.* [7] examine the formation of coalitions, and agents must choose partners based on the expected payoffs gained over a period of time. Although the work considers partner selection, it focuses on cooperative encounters and does not deal with the problems of negotiation. Another approach to opponent selection, using cognition-based strategies, is described in [31], in which several heuristic decision-functions facilitate the selection of optimal opponents. However, it not deal with considerations of conflict or expected negotiation duration, but instead focuses on the efficacy of the decision heuristics.

By providing an adaptive approach to pre-negotiation as described in this chapter, we increase the chance that agents will engage in negotiations that end successfully in a deal. In doing so, we decrease the amount of wasted time and resources that can arise from the need to engage in repeated negotiations with different partners over the same goals when a particular negotiation fails. Considering the importance and ubiquity of negotiation in agent systems, reducing the number of times they fail can greatly help increase overall system efficiencies.

Chapter 7

Conclusions

7.1 Introduction

Negotiation represents one of the most important forms of interaction in agent systems, allowing agents with different goals and agendas to find ways to cooperate and solve conflicts. In this context, a major problem involves increasing the effectiveness of agents using negotiation in dynamic domains where human guidance is limited or undesirable. The work developed in this thesis addresses this problem in two main ways.

- First, it provides a model of negotiation goals that allows agents to flexibly determine their negotiation stance in the light of constraints imposed by existing goals and environmental limitations on time, resource and opportunity to negotiate.
- Second, it provides motivational mechanisms to enable an agent to assign worth to goals, so that it can then reason about alternative negotiated settlements in the context of existing goals.

In this chapter we present a summary of the main contributions of the work and discuss its limitations and the opportunities for future research that can build upon and extend the work here.

7.2 Contributions

The work in this thesis makes the following contributions. First, it provides *a model of negotiation goals* to describe and enable the autonomous classification of issues for negotiation. Second, it provides a classification of *unilateral issue types*, determined through deliberations over the effects of issues on existing goals. Third, it provides an analysis and classification of *bilateral issue types* that result from the unilateral issue classifications of individual agents. Finally, it provides *mechanisms* for agents to increase their chance of entering into successful negotiations, exploiting information gained from previous negotiations.

With these contributions, we enable the design of agents displaying characteristics not previously considered in the context of negotiation. In particular, we assist in the construction of agent negotiators that are able to flexibly tailor their approach to negotiation based on several internal and external criteria. This is needed in domains that are highly dynamic; as automated systems in which agents operate increase in complexity, such dynamism will become more prevalent, and there will be a concomitant need to find better and more flexible ways to manage them. The aim of the work in this thesis is to address some of these concerns by increasing the effectiveness and flexibility of negotiation.

In what follows we review the main contributions of the work in this thesis.

7.2.1 A Model of Negotiation Goals

The basic foundation on which this thesis rests (apart from the SMART framework) is a model of deliberative negotiation goals. While most existing work on negotiation has taken an implicit approach to the definition and use of goals (focusing on a utility maximising approach, for example), in this thesis our concern has been to develop a model of pre-negotiation that builds on the deliberative tradition of symbolic agent architectures. As a result, our model of negotiation goals has adopted a symbolic approach, in which goals are explicitly defined, facilitating the consideration and use of the important class of deliberative agents (such as BDI) that have been widely investigated and, in many cases, deployed. In particular, the model facilitates the kind of reasoning about goals that is necessary to try to improve behaviour and results from a deliberative stance, and provides the necessary hooks for it to be integrated with the deliberative approach.

- *An analysis and taxonomy of negotiation issues*

  Our model of negotiation goals is distinct from previous explicit models, including both traditional deliberative BDI-type models, and the SMART model on which our initial work is based, in that it provides more detail of negotiation objectives and issues, allowing them to be analysed and classified. In particular, goals in the model include a set of *potential* negotiation issues, and preferences over them, the latter of which determine the status (or nature) of the former, and how they may be reasoned about. However, the model extends these other models to allow them to incorporate reasoning about negotiation.

- *A deliberative preference determination mechanism*

  Establishing such a classification of negotiation issues is an endogenous process, without the need for external direction. It is achieved simply from analysing preferences which, in turn, are determined by examining existing goals (in the BDI sense) and considering the effects of particular issue settlements on them. The goal model uses relationships between goals and potential settlements to order preferences; in this way, the higher level goals and issues of an agent are consistent and integrated with the base deliberative architecture.

- *A worth-based motivation model applied to negotiation*

  The concept of motivation has been introduced previously into agent architectures as a means of generating goals. In our work, we have provided a simple model of motivation (building on existing work) and have applied it to the problems of pre-negotiation. For negotiation, it is important that alternative settlements for issues can be both represented and reasoned about, yet determining how preferences for different settlements can be generated is not straightforward in deliberative architectures, since goals are symbolic and traditionally do not have numerical worths. We have used our motivation model to assign worth to negotiation goals for analysing and comparing the relative benefit of such goals to an agent. In particular, our motivation model enables the dynamic determination of preferences over issue settlements to ensure that agents take a stance on issues that is in concord with their current activities defined by their existing goals. While the model itself is simple and similar to existing models, our application of it to the problems of pre-negotiation represents a novel contribution.

7.2.2 Bilateral Issue Analysis

The form of a negotiation depends on the approaches of both participants. For example, negotiations can be more or less cooperative or competitive depending on the stance that each participant adopts. By examining the unilateral issue classifications of each negotiation participant, we can determine the resulting bilateral issues that arise when agents negotiate which, in turn, allows buyers to examine different sellers and form expectations regarding likely negotiation outcomes. The key feature of the model is presented below.

- *A taxonomy of bilateral issue types.* Three bilateral issue types are defined. First, deadlocked issues arise when there are no acceptable settlements common to both participants, so that no agreements can be found. Second, congruent issues occur when either one or both agents have no preferences for their settlement, or when both participants specify the same aspiration settlement. Thus, when considering alternative sellers, a buyer should try to find those sellers whose unilateral issue classification are congruent with its own. Finally, competitive issues are those for which there is one or more acceptable settlements common to both participants but for which the aspiration settlement is not jointly preferred. Within this type of bilateral issue, there are two subclassifications. One-way negotiable issues are those for which only one participant has classified the issue as negotiable and two-way negotiable issues are those for which both participants classify the issue as negotiable. The existence of competitive issues entails that a search is made to find a settlement acceptable to both participants. This can increase the duration of the negotiation and may mean that deadlines are exceeded. In this case, it is important for participants to recognise the impact of such issues on the duration of the negotiation and ensure that negotiations containing such issues do not exceed their deadline.

7.2.3 Mechanisms for Dynamic Negotiation Preparation

- *Dynamic resource assessment*

  We have developed a dynamic resource evaluation model that allows agents to determine the amount they commit to negotiations in response to changes in resource availability and the expected worth to be gained from negotiation.

  Resources are simply quantities an agent possesses that are depleted when actions are performed. As resources become scarce, it becomes more important to optimise their use so that they are not expended on objectives with little worth. We have provided mechanisms to enable agents to re-evaluate the worth of resources in relation to how much resource they have access to. In situations where resources are few, the value on resources is increased, and conversely the value is decreased when resources are plentiful. This allows an examination of worth loss in resource expenditure during negotiation in relation to the worth gain from satisfying goals. More specifically, it provides a means for agents to determine their reservation price for negotiation goals, by enabling comparisons between the worth of the goal and the worth of the resource used to secure a negotiated agreement.

- *Modelling negotiation time and opportunity constraints*

  Negotiation is often bounded by time and partner availability. Deadlines create problems when the duration of a negotiation is too long to find settlements on all issues. Similarly, the risk that negotiation will fail is compounded when there are few opportunities to (re-)negotiate in terms of numbers of available sellers.

  The ability to negotiate effectively can be severely compromised if time constraints are imposed. The duration of a negotiation is determined by the existence and size of zones of agreements on the issues of negotiation between participants. With larger sets of potential settlements, duration increases, and the negotiation may exceed the deadline; in such cases, remedial action must be taken. We have developed a model that facilitates reasoning about negotiation duration and deadlines, and have constructed mechanisms to effectively manage the risks that negotiation durations will exceed given deadlines.

  Similarly, when there are few negotiation partners, limits are again imposed, since the possibilities for coming to a successful result are reduced. We have developed a means to incorporate the consequences of failure due to a lack of partners in terms of the worth lost if the negotiation goal is not satisfied. Both the risk of failure and the consequences of failure must be considered; our model uses information on the number of available negotiation partners and the loss (in worth) that would be incurred in the case of failure in providing a way to relax constraints and increase the chance of success.

- *Dynamic issue and preference adjustment*

  Using the information provided by the time, resource and opportunity models, we have developed mechanisms to enable agents to effectively manage the various environmental constraints by modifying their negotiation stance. In situations with strong time constraints, agents can decrease the duration of a negotiation by selectively pruning issues from their negotiation set. First, when resource availability is high, agents can attempt to cost-out issues by increasing the resources committed to securing a deal. The increase in resources influences sellers to drop issues so that the buyer can obtain a settlement without negotiating over the issues. Second, when resources are scarce, buyers can selectively prune issues of low worth to reduce the expected negotiation duration. In addition, when opportunity to negotiate is low, buyers can modify their flexibility over issues by selectively increasing their set of acceptable settlements to include those previously discarded. This improves the chance that of a successful deal with the current negotiation partner, avoiding the need to re-negotiate.

7.3 Limitations and Further Work

The key contributions of our work lie in the models and mechanisms provided to increase the effectiveness of agent negotiators in dynamic domains. Although we have covered a broad range of issues surrounding pre-negotiation, there are inevitably limitations in scope and not all aspects have been covered. In particular, several problems have not been addressed and there are a number of extensions to our work that would increase its value in the future.

- *Consideration of opponent negotiation strategies*

  Although we have explored how buyers can adapt their negotiation stance in anticipation of the issue classifications of sellers, we have not considered how the use of different negotiation strategies can influence the classification of issues. Since the adoption of different strategies can greatly affect the outcome of a negotiation, their consideration during pre-negotiation may further improve the chance that negotiations will be successful. For example, expectations about strategy may inform the decisions about which issues to classify as slack or negotiable since, if it is known that a particular seller will adopt a particular strategy leading to agreement for a less preferred settlement, an agent can change its stance by classifying an issue as fixed, thereby refusing to enter into negotiation over its settlement.

  Such expectations of the influence of likely opponent strategies on issue classifications have not been considered in our work, but incorporating such knowledge could further refine the classifications made to prevent less preferred settlements being forced onto the buyer.

- *Adaptation of issue classifications during negotiation*

  Our models only use expectations about issue classifications of opponents to adapt the negotiation stance before negotiation begins. Once negotiation begins, the actual classifications of sellers become apparent and there is an opportunity to re-evaluate issue classification in the light of this information.

The classification process can therefore be made more adaptive if re-classifications can be made in response to information revealed after negotiation has begun. This could reduce the errors in attempting to anticipate opponent issue classifications and increase the overall effectiveness of tailoring the negotiation to the current situation.

- *Relationships between negotiation participants*

  Issue classification can be used strategically to tailor the approach to negotiation to suit the needs of both participants. In particular, existing relationships between the negotiation participants could influence the classifications made so that, for example, if two agents belonging to the same organisation negotiate, they may both identify more slack issues, producing a more cooperative encounter. We have not considered this kind of information about relationships in this thesis.

  Nevertheless, much work has been done on modelling agent relationships and their effects on interaction. Knowledge of shared aims and goals between agents from the same organisations, can be used to refine the issue classifications made so that more congruent issues arise and negotiations become quicker and easier to find agreements. Incorporating such information about existing relationships into the issue classification process could be a valuable way to further increase the effectiveness of negotiation preparation.

- *Sequential, independent issues*

  Our model assumes that issues are independent and can be settled sequentially. By making such assumptions, we simplify the analysis, but recognise that many negotiations may be more complex. Recent research has explored the complexities of negotiations involving concurrent, dependent issues, and findings within this sphere could usefully extend our model to a broader range of negotiations.

- *Costing out issues*

  When costing out issues, we limit ourselves to examining only the worth of aspiration settlements. More realistically, such costing out could be applied to settlements other than the aspiration settlement. In this way, less optimal, but cheaper alternatives could be considered, and could usefully be employed in situations of low resource levels.

7.4 Concluding Remarks

The work presented in this thesis helps to increase the effectiveness of agent negotiators in domains where guidance or direction from human users is difficult or undesirable to apply. In particular, we have provided models to allow agents to autonomously tailor their approach to negotiation in a manner sensitive to the current situation and consistent with existing goals. In elaborating our model of pre-negotiation, we have provided a number of decision-making mechanisms that allow agents to refine their approach to negotiation to take into account both information about opponents and environmental constraints surrounding the negotiation encounter. Underlying all this work is the key concept of motivation, which provides the foundation for agent autonomy by enabling on-the-fly evaluations of goals and negotiation settlements, thus ensuring that decisions made regarding negotiation objectives and acceptable settlements are sensitive to the overall context of an agent at the time the negotiation is required.

Our research addresses the needs of agent-based system designers who, increasingly, are being called upon to design systems that manage larger and more complex domains. Autonomous operation of agents in such systems is increasingly important as their size and complexity grows, and providing the means to enable agents to conduct negotiations more autonomously increases their ability to manage goals and interactions without human direction. For future applications of agent technology, such autonomy will be vital to manage the complexities inherent in the processes and systems of tomorrow's organisations and businesses. In developing the work in this thesis, we believe that we have taken some important steps towards the realisation of such systems.

Bibliography

[1] P. E. Agre and D. Chapman. Pengi: An implementation of a theory of activity. In *Proceedings of the American Association for Artificial Intelligence*, pages 268–272, Seattle, WA, 1987.

[2] A.Newell and H.A. Simon. Computer science as empirical inquiry: symbols and search. *Communications of the ACM*, 19(3):113–126, 1976.

[3] R. Ashri, M. Luck, and M. d'Inverno. On identifying and managing relationships in multi-agent systems. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, pages 743–748, 2003.

[4] R. Ashri, I. Rahwan, and M. Luck. Architectures for negotiating agents. In *The 3rd International/Central and Eastern European Conference on Multi-Agent Systems*, pages 136–146, 2003.

[5] R. Axelrod. *The evolution of cooperation.* New York: Basic Books, 1984.

[6] F. Balbo, G. Scemama, and M. Tendjaoui. Satir: A multi-agent decision support system for bus networks, agent technologies in logistics. In *European Conference on Artificial Intelligence*, pages 2002–07, 2002.

[7] B. Banerjee and S Sen. Selecting partners. In *AGENTS 2000, Proceedings of the fourth international conference on Autonomous agents*, pages 261–262. ACM, 2000.

[8] S. Barber and C. Martin. Agent autonomy: Specification, measurement, and dynamic adjustment. In *Proceedings of the Autonomy Control Software Workshop, Autonomous Agents 1999 (Agents'99)*, pages 8–15. Seattle WA, 1999.

[9] M. Barbuceanu, T. Gray, and S. Mankovski. Role of obligations in multiagent coordination. *Applied Artificial Intelligence*, 13(1/2):11–38, 1999.

[10] C. Bartolini, C. Preist, and N.R. Jennings. A software framework for automated negotiation. In *Software Engineering for Multi-Agent Systems III: Research Issues and Practical Applications*. Springer, 2005.

[11] L.P. Beaudoin and A. Sloman. A study of motive processing and attention. In *Prospects for Artificial Intelligence: Proceedings of AISB93*, 1993.

[12] M. Bratman. *Intentions, Plans, and Practical Reason*. Harvard University Press, 1987.

[13] C. Breazeal, A. Edsinger, P. Fitzpatrick, and B. Scassellati. Active vision for sociable robots. *Socially Intelligent Agents - The Human in the Loop, Special Issue IEEE Transactions on Man, Cybernetics, and Systems, Part A: Systems and Humans*, 31(5):443–453, 2001.

[14] R.A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47(1–3):139–159, 1991.

[15] R.A. Brooks. *Cambrian intelligence: the early history of the new AI*. MIT Press, Cambridge, MA, USA, 1999.

[16] K. Bryson, M. Luck, M. Joy, and D. Jones. Agent interaction for bioinformatics data management. *Applied Artificial Intelligence*, 15(10):917–947, 2001.

[17] C. Castelfranchi. Guarantees for autonomy in cognitive agent architecture. In M. Wooldridge and N. R. Jennings, editors, *Intelligent Agents: Theories, Architecture, and Languages (LNAI Volume 890)*, pages 56–70. Springer, January 1995.

[18] R. Charton, A. Boyer, and F. Charpillet. Learning of mediation strategies for heterogeneous agents cooperation. In *Proceedings. 15th IEEE International Conference on Tools with Artificial Intelligence*, pages 330–333, 2003.

[19] A.M. Coddington and M. Luck. Towards motivation-based plan evaluation. In I. Russell and S. Haller, editors, *Proceedings of Sixteenth International FLAIRS Conference*, pages 298–302, 2003.

[20] R. M. Coehoorn and N. R. Jennings. Learning an opponent's preferences to make effective multi-issue negotiation tradeoffs. In *Proceedings of the 6th International Conference on E-Commerce*, pages 59–68, 2004.

[21] I. Craig. *Formal Specification of Advanced AI Architectures*. Ellis Horwood, 1991.

[22] M. Dastani and L. W. N. van der Torre. Programming boid-plan agents: Deliberating about conflicts among defeasible mental attitudes and plans. In *AAMAS*, pages 706–713, 2004.

[23] K. Dautenhahn and C. Nehaniv. Imitation in animals and animats. *Artificial Intelligence and Simulation of Behaviour*, 1(4):303–304, 2003.

[24] E. David, R. Azoulay-Schwartz, and Kraus S. Protocols and strategies for automated multi-attribute auctions. In *ICMAS-2002 Fourth International Conference on MultiAgent Systems*, 2002.

[25] H. de Garis. Cam-brain : Growing an artificial brain with a million neural net modules inside a trillion cell cellular automata machine. *Journal of the Society of Instrument and Control Engineers (SICE)*, Vol.33(No.2), 1994.

[26] M. d'Inverno, D. Kinny, M. Luck, and M. Wooldridge. A formal specification of dMARS. In Singh, Rao, and Wooldridge, editors, *Intelligent Agents IV: Proceedings of the Fourth International Workshop on Agent Theories, Architectures and Languages*, Lecture Notes in Artificial Intelligence, 1365, pages 155–176. Springer Verlag, 1998.

[27] M. d'Inverno and M. Luck. Engineering AgentSpeak(L): A formal computational model. *Journal of Logic and Computation*, 8(3):233–260, 1998.

[28] M. d'Inverno and M. Luck. *Understanding Agent Systems*. Springer, 2001.

[29] M. d'Inverno, M. Luck, and M. Wooldridge. Cooperation Structures. In Martha E. Pollack, editor, *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)*, pages 600–605, Nagoya, Japan, 1997.

[30] E.H. Durfee. Distributed problem solving and planning. In G. Weiss, editor, *Multiagent Systems:A Modern Aproach to Distributed Artificial Intelligence*, chapter 3, pages 121–164. MIT Press, Cambridge, MASS, 1999.

[31] P. S. Dutta, L. Moreau, and N. R. Jennings. Finding interaction partners using cognition-based decision strategies. In *Proceedings of The IJCAI-2003 workshop on Cognitive Modeling of Agents and Multi-Agent Interactions*, pages 46–55, 2003.

[32] P. Faratin. *Automated Service Negotiation Between Autonomous Computational Agents*. PhD thesis, University of London, Queen Mary College, Department of Electronic Engineering, 2000.

[33] P. Faratin, C. Sierra, and N. R. Jennings. Negotiation decision functions for autonomous agents. *Journal of Robotics and Autonomous Systems*, 24(3-4):159–182, 1998.

[34] P. Faratin, C. Sierra, N. R. Jennings, and P. Buckle. Designing responsive and deliberative automated negotiators. In *Proceedings of the AAAI Workshop on Negotiation: Settling Conflicts and Identifying Opportunities*, pages 12–18, Orlando, FL, 1999.

[35] S. Fatima, M. Wooldridge, and N. R. Jennings. An agenda based framework for multi-issues negotiation. *Artificial Intelligence Journal*, 2004.

[36] S.S. Fatima, M Wooldridge, and N.R. Jennings. Multi-issue negotiation under time constraints. In *ICMAS-2002 Fourth International Conference on MultiAgent Systems*, 2002.

[37] J. Ferber. *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Addison Wesley Longman, 1999.

[38] I.A. Ferguson. Touring machines: Autonomous agents with attitudes. *IEEE Computer*, 25(5):51–55, 1992.

[39] R. E. Fikes and N. Nilsson. STRIPS: A New Approach to the Application Theorem Proving to Problem Solving. *Artificial Intelligence*, 5(2):189–208, 1971.

[40] J. Fox, P. Krause, and S. Ambler. Arguments, contradictions and practical reasoning. In *In proceedings of the 10th European Conference on Artificial Intelligence*, pages 623–627, 1992.

[41] S. Franklin and A. Graesser. Is it an agent, or just a program?: A taxonomy for autonomous agents. In Jörg P. Müller, Michael J. Wooldridge, and Nicholas R. Jennings, editors, *Proceedings of the ECAI'96 Workshop on Agent Theories, Architectures, and Languages: Intelligent Agents III*, volume 1193 of *LNAI*, pages 21–36. Springer, August12–13 1997.

[42] M. P. Georgeff. Distributed multi-agent reasoning system dMARS. Technical report, Australian Artificial Intelligence Institute, Melbourne, Australia, 1994.

[43] M.P. Georgeff and F.F. Ingrand. Monitoring and control of spacecraft systems using procedural reasoning. Technical Report 03, Australian Artificial Intelligence Institute, Melbourne, Australia, 1989.

[44] L. Godo, J. Puyol-Gruart, J. Sabater, V. Torra, P. Barrufet, and X. Fabregas. A multi-agent system approach for monitoring the prescription of restricted use antibiotics. *Artificial Intelligence in Medicine Special Issue: Software Agents in Healthcare*, 2003.

[45] R. Goodwin. Formalizing properties of agents. Technical report, Carnegie Mellon University, 1993.

[46] N. Griffiths. *Motivated Cooperation.* PhD thesis, University of Warwick, 2000.

[47] N. Griffiths and M. Luck. Cooperative plan selection through trust. In *Modelling Autonomous Agents in a Multi-Agent World*, volume 1647 of *LNAI*, pages 162–174. Springer, 1999.

[48] T. Halliday. *Causes and Effects.* Blackwell Scientific, 1983.

[49] N. Haque, N. R. Jennings, and L. Moreau. Resource allocation in communication networks using market-based agents. *Intenational Journal of Knowledge Based Systems*, 18(4-5):163–170, 2005.

[50] M. He and N.R. Jennings. Designing a successful trading agent using fuzzy techniques. *IEEE Trans on Fuzzy Systems*, 12(3):389–410, 2004.

[51] M. He and H. Leung. An agent bidding strategy based on fuzzy logic in a continuous double auction. In *Autonomous Agents*, pages 61–62. ACM Press, 2001.

[52] J. Henrich. Does culture matter in economic behavior? ultimatum game bargaining among the machiguenga of the peruvian amazon. *American Economic Review*, 90(4):973–979, 2000. available at http://ideas.repec.org/a/aea/aecrev/v90y2000i4p973-979.html.

[53] V. Hilaire, O. Simonin, A. Koukam, and J. Ferber. A formal approach to design and reuse agent and multiagent models. In J. Odell, P. Giorgini, and J.P. Muller, editors, *Agent-Oriented Software Engineering V*, pages 142–157. Springer, 2004.

[54] M.G Hinchey and J.P. Bowen, editors. *Applications for Formal Methods.* Prentice Hall International Series In Computer Science, 1995.

[55] E. Horvitz. *Automated reasoning for biology and medicine*, chapter Advances in Computer Methods for Systematic Biology: Artificial Intelligence, Databases, and Computer Vision. Johns Hopkins University Press, Baltimore, 1993.

[56] M. J. Huber. JAM: A BDI-theoretic mobile agent architecture. In *Proceedings of The Third International Conference on Autonomous Agents*, pages 236–243, Seattle, WA, 1999.

[57] M. N. Huhns and L.M. Stephens. Multiagent systems and societies of agents. In G. Weiss, editor, *Multiagent Systems:A Modern Aproach to Distributed Artificial Intelligence*, chapter 2, pages 79–120. MIT Press, Cambridge, MASS, 1999.

[58] M.N. Huhns and M.P. Singh. Workflow agents. *IEEE Internet Computing*, 2(4):94–96, 1998.

[59] N. R. Jennings. Specification and implementation of a belief-desire-joint-intention architecture for collaborative problem solving. *International Journal of Intelligent and Cooperative Information Systems*, 2(3):289–318, 1993.

[60] N. R. Jennings, P. Faratin, M. J. Johnson, T. J. Norman, P. O'Brien, and M. E. Wiegand. Agent-based business process management. *International Journal of Cooperative Information Systems*, 5(2&3):105–130, 1996.

[61] N. R. Jennings, P. Faratin, T. J. Norman, P. O'Brien, B. Odgers, and J. L. Alty. Implementing a business process management system using ADEPT: A real-world case study. *Int. Journal of Applied Artificial Intelligence*, 14(5):421–463, 2000.

[62] N. R. Jennings, T. J. Norman, and P. Faratin. ADEPT: An agent-based approach to business process management. *ACM SIGMOD Record*, 27(4):32–39, 1998.

[63] N. R. Jennings and M. Wooldridge. Applying agent technology. *Journal of Applied Artificial Intelligence*, 9(4):351–361, 1995.

[64] N.R. Jennings, A.G. Cohn, M. Fox, D. Long, M. Luck, D.T. Michaelides, S. Munroe, and M.J. Weal. *Cognitive Systems: Information Processing Meets Brain Science*, chapter 8, pages 163–188. Elsevier Academic Press, 2006.

[65] E. Kalai and M Smorodinsky. Other soultions to nash's bargaining problem. *Econometrica*, 43:413–18, 1975.

[66] M. Karlins and H.I. Abelson. *Persuasion.* Crosby Lockwood and Son, 1970.

[67] J. Krane. Military tests software agents for quick intelligence. Washington Times, October 1 2001.

[68] S. Kraus. *Strategic Negotiation in Multi-Agent Environments.* MIT Press, Cambridge, USA, 2001.

[69] S. Kraus, K. Sycara, and A. Evenchik. Reaching agreement through argumentation: a logical model and implementation. *Artificial Intelligence*, 104(39):1–69, 1981.

[70] V. Krishna. *Auction Theory.* Academic Press, Inc., 2002.

[71] R. Vd. Krogt and M. deWeerdt. Self-interested planning agents using plan repair. In *The 15th International Conference on Automated Planning and Scheduling*, 2005.

[72] Z. Kunda. The case for motivated reasoning. *Psychological Bulletin*, 108(3):480–498, 1990.

[73] K. Kurbel and I. Loutchko. Multi-agent negotiation under time constraints on an agent-based marketplace for personnel acquisition. In *Proceedings of the 3rd International Symposium on Multi-Agent Systems, Large Complex Systems, and E-Business (MALCEB2002)*, Erfurt, Germany, 2002.

[74] P.R. Lawrence and N. Nohria. *Driven: How Human Nature Shapes Our Choices.* Jossey-Bass, 2002.

[75] D. Lightfoot. *Formal Specification in Z.* Palgrave, 2001.

[76] A.R. Lomuscio, M. Wooldridge, and N.R. Jennings. *A Classification Scheme for Negotiation in Electronic Commerce*, pages 19–33. Springer Verlag, 2001.

[77] F. Lopez y Lopez, M Luck, and M. d'Inverno. Constraining autonomy through norms. In *Proceedings of the First International Conference on Autonomous Agents and Multi-Agent Systems*, 2002.

[78] A. Lucas and S. Goss. The potential for intelligent software agents in defence simulation. Technical report, Agent-Oriented Software Group, 1999.

[79] M. Luck and M. d'Inverno. A formal framework for agency and autonomy. In Victor Lesser and Les Gasser, editors, *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pages 254–260. AAAIP, 1995.

[80] M. Luck and M. d'Inverno. Motivated behaviour for goal adoption. In Zhang and Lukose, editors, *Multi-Agent Systems: Theories, Languages and Applications Workshop on Distributed Artificial Intelligence*, Lecture Notes in Artificial Intelligence, 1544, pages 58–73. Springer Verlag, 1998.

[81] M. Luck, S. Munroe, and M. d'Inverno. Autonomy: Variable and generative. In H. Hexmoor, C. Castelfranchi, and R. Falcone, editors, *Agent Autonomy*, pages 9–22. Kluwer, 2003.

[82] X. Luo, N.R. Jennings, N. Shadbolt, H. Loung, and J Lee. A fuzzy constraint based knowledge model for bilateral, mulit-issue negotiations in competative environments. *Artificial Intelligence*, 148(1-2), 2002.

[83] T. Magedanz, K. Rothermel, and S. Kraus. Intelligent agents: An emerging technology for next generation telecommunications? In *INFOCOM'96*, San Francisco, CA, USA, 24-28 1996.

[84] S. Marsh. *Trust in Distributed Artificial Intelligence*, pages 94–112. Springer Verlag, 1994.

[85] A. Maslow. *The farther reaches of human nature.* New York: Penguin Books, 1971.

[86] N. Matos, C. Sierra, and N. R. Jennings. Determining successful negotiation strategies: an evolutionary approach. In Y. Demazeau, editor, *Proceedings of the 3rd International Conference on Multi-Agent Systems (ICMAS-98)*, pages 182–189, Paris, France, 1998. ieeep.

[87] Y. Matsumoto and S. Fujita. An auction agent for bidding on combinations of items,. *Proceedings of the fifth international conference on Autonomous Agents*, 2001.

[88] J. McCarthy, M.L. Minsky, N. Rochester, and C.E. Shannon. A proposal for the dartmouth summer research project on artificial intelligence. Technical report, Dartmouth College, Hanover, NH, USA, 1955.

[89] D. McFarland. *Animal Behaviour.* Longman Scientific and Technical, 1985.

[90] D. McFarland and T. Bosser. *Intelligent Behaviour in Animals and Robots.* The MIT Press, 1993.

[91] D. Moffat and N. Frijda. Where there's a will there's an agent. In M. Wooldridge and N. R. Jennings, editors, *Intelligent Agents (LNAI Volume 890)*, pages 245–260. Springer Verlag, 1995.

[92] H. Moravec. Rise of the robots. *Scientific American*, pages 124–135, December 1999.

[93] P. Morignot and B. Hayes-Roth. Adaptable motivational profiles for autonomous agents. Technical report, Knowledge Systems Laboratory, Stanford University, 1995.

[94] P. Morignot and B. Hayes-Roth. Motivated agents. Technical report, Knowledge Systems Laboratory, Stanford University, 1996.

[95] J. Muller and M. Pischel. The agent architecture InteRRaP: Concept and application. Technical report, Technical Report RR-93-26, DFKI Saarbrucken, 1993., 1993.

[96] S. Munroe and A. Cangelosi. Learning and the evolution of langauge: The role of cultural variation and learning costs in the baldwin effect. *Artificial Life*, 8:311–339, 2003.

[97] S. Munroe and M. Luck. Motivation-based selection of negotiation opponents. In *LNAI 3451: Engineering Societies in the Agents World*. Springer, 2004.

[98] S. Munroe and M. Luck. Agent autonomy through the 3M motivational taxonomy. In *Agents and Computational Autonomy: Potential, Risks, and Solutions*. Lecture Notes in Artificial Intelligence 2969, Springer, 2005.

[99] S. Munroe and M. Luck. Balancing conflict and cost in the selection of negotiation opponents. In *Proceedings of the First International Workshop on Rational, Robust, and Secure Negotiations in Multi-Agent Systems*, 2005.

[100] S. Munroe, M. Luck, and M. d'Inverno. Towards motivation-based decisions for worth goals. In *Proceedings of the 3rd International Central and Eastern European Conference on Multi-Agent Systems*, 2003.

[101] A. Muthoo. *Bargaining Theory with Applications*. Cambridge University Press, 1999.

[102] J. Nash. The bargaining problem. *Econometrica*, 18:155–162, 1950.

[103] BBC News. Windows 2000 bug starts virus war. World Wide Web: http://news.bbc.co.uk/1/hi/technology/4162124.stm, August 2005.

[104] T.D. Nguyen and N.R. Jennings. Managing commitments in multiple concurrent negotiations. *Int J. Electronic Commerce Research and Applications*, 2005.

[105] N.J. Nilsson. *Problem-Solving Methods in Artificial Intelligence*. New York: McGraw-Hill, 1971.

[106] E. Norling. Folk psychology for human modelling: Extending the BDI paradigm. In *Third International Joint Conference on Autonomous Agents and Multiagent Systems*, 2004.

[107] T. J. Norman and D. Long. Goal creation in motivated agents. In M. Wooldridge and N. R. Jennings, editors, *Intelligent Agents (LNAI Volume 890)*, pages 277–290. Springer Verlag, 1995.

[108] T. J. Norman and D. Long. Alarms: An implementation of motivated agency. In M. Wooldridge, J.-P. Müller, and M. Tambe, editors, *ATAL95*, pages 219–234. Springer, 1996.

[109] M.J. Osborne and A. Rubinstein. *Bargaining and Markets (Economic Theory, Econometrics, and Mathematical Economics).* Academic Press, Inc., 1990.

[110] S. Parsons, C. Sierra, and N. R. Jennings. Agents that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.

[111] H. Van Dyke Parunak. Industrial and practical applications of DAI. In G. Weiss, editor, *Multiagent Systems:A Modern Aproach to Distributed Artificial Intelligence*, chapter 9, pages 377–421. MIT Press, Cambridge, MASS, 1999.

[112] J. Patel, W. T. L. Teacy, N. R. Jennings, and M. Luck. A probabilistic trust model for handling inaccurate reputation sources. In P. Herrmann, V. Issarny, and S. Shiu, editors, *Third International Conference on Trust Management*, pages 193–20, 2005.

[113] D. Perugini, D. Lambert, L. Sterling, and A. Pearce. Agents for military logistics planning. In *Agent Technologies in Logistics Workshop, European Conference on Artificial Intelligence*, 2002.

[114] P. Petta. Principled generation of expressive behavior in an interactive exhibit. In Velasquez J.D., editor, *Third International Conference on Autonomous Agents (Agents '99)*, pages 94–98, 1999.

[115] D.G. Pruitt. *Negotiation Behaviour.* Academic Press, 1981.

[116] H. Raiffa, J. Richardson, and D. Metcalfe. *The Art and Science of Negotiation.* Harvard University Press, 2002.

[117] S. D. Ramchurn, D. Huynh, and N. R. Jennings. Trust in multi-agent systems. *The Knowledge Engineering Review*, 19, 2004.

[118] S. D. Ramchurn, C. Sierra, L. Godo, and N. R. Jennings. A computational trust model for multi-agent interactions based on confidence and reputation. In R. Falcone, S. Barber, L. Korba, and M. Singh, editors, *Workshop on Deception, Trust, and Fraud, AAMAS*, pages 69–75, 2003.

[119] S. D. Ramchurn, C. Sierra, L. Godo, and N.R. Jennings. Devising a trust model for multi-agent interactions using confidence and reputation. *Int. J. of Applied Artificial Intelligence*, 18(9-10), 2004. to appear.

[120] S.D. Ramchurn, C. Sierra, L. Godo, and N.R. Jennings. Devising a trust model for multi-agent interactions using confidence and reputation. *International Journal of Applied Artificial Intelligence*, 18(9-10):833–852, 2004.

[121] A. S. Rao. AgentSpeak(L): BDI agents speak out in a logical computable language. In W. van der Velde and J. W. Perram, editors, *Agents Breaking Away (LNAI 1038)*, pages 42–55. Springer, 1996.

[122] A. S. Rao and M.P. Georgeff. BDI agents: from theory to practice. In Victor Lesser, editor, *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS'95)*, pages 312–319, San Francisco, CA, USA, 1995. MITP.

[123] A. Rogers, E. David, and N. Jennings. Self-organized routing for wireless micro-sensor networks. *IEEE Transactions on Systems, Man and Cybernetics*, 35(3):349–359, 2005.

[124] J.S. Rosenschein and G. Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. MIT Press, 1994.

[125] A Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica*, 50:97–109, 1982.

[126] S.J. Russel and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995.

[127] J. Sabater and C Sierra. Social regret, a reputation model based on social relations. *SIGecom Exch.*, 3(1):44–56, 2002.

[128] T.W. Sandholm. Distributed rational decision making. In G. Weiss, editor, *Multiagent Systems:A Modern Aproach to Distributed Artificial Intelligence*, chapter 5, pages 201–258. MIT Press, Cambridge, MASS, 1999.

[129] H. Saunders. *Negotiation Theory and Practice*, chapter We Need a Larger Theory of Negotiation: The Importance of Pre-Negotiation Phases, pages 57–70. Cambridge, 1991.

[130] M.F W. Shen and D.H. Norrie. Metamorph: An adaptive agent-based architecture for intelligent manufacturing. *nternational Journal of Production Research*, 37(10):2159–2174, 1999.

[131] W. Shen and D.H. Norrie. Agent-based systems for intelligent manufacturing: A state-of-the-art survey. *Knowledge and Information Systems, an International Journal*, 1(2):129–156, 1999.

[132] R. Siegwart and I.R. Nourbakhsh. *Introduction to Autonomous Mobile Robots.* The MIT Press, 2004.

[133] C. Sierra, P. Faratin, and N. 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-97)*, pages 17–35, Ronneby, Sweden, 1997.

[134] C. Sierra, P. Faratin, and N. R. Jennings. Deliberative automated negotiators using fuzzy similarities. In *Proceedings of the EUSFLAT-ESTYLF Joint Conference on Fuzzy Logic*, pages 155–158, Palma de Mallorca, Spain, 1999.

[135] C. Sierra, N. R. Jennings, P. Noriega, and S. Parsons. A framework for argumentation-based negotiation. *Lecture Notes in Computer Science*, 1365:1774, 1998.

[136] H.A. Simon. *Models of Thought.* Yale University Press, 1979.

[137] A. Sloman. Motives, mechanisms, and emotions. *Cognition and Emotion*, 1:217–233, 1987.

[138] A. Sloman and M. Croucher. Why robots will have emotions. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pages 197–202. Vancouver, B.C, 1981.

[139] L.K. Soh, T. Blank, L. D. Miller, and Person S. Ilmda: An intelligent learning materials delivery agent and simulation. In *The International Electro-Information Technology Conference (EIT'2005)*, 2005.

[140] E. Spier and D. McFarland. A finer-grained motivational model of behaviour sequencing. In *From Animals to Animats 4: Proceedings of the fourth conference on the Simulation of Adaptive Behavior*, 1996.

[141] J.M. Spivey. *The Z Notiation, 2nd ed.* Prentice Hall, Hemel Hempstead, 1992.

[142] J.M. Spivey. *The fUZZ Manual.* Computing Science Consultancy, Oxford, 1993.

[143] A. Stauffer and M. Sipper. The data and signals cellular automation and its application to growing structures. *Artificial Life*, 10(4):463–477, 2004.

[144] P. Szor. *The Art of Computer Virus Research and Defense*. Addison-Wesley Professional, 2005.

[145] L. Tesfatsion. *Computational Approaches to Economic Problems*, chapter A trade network game with endogenous partner selection, pages 249–269. Kluwer, 1997.

[146] D. Urbig and K Schroter. C-IPS approach to negotiating agents: Specifying dynamic interdependencies between issue, partner, and step. In *3rd International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS 2004)*, 2004.

[147] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton: Princeton University Press, 1944.

[148] K. Warwick. *I, Cyborg*. Century, 2002.

[149] I. Watson. *Applying Case-Based Reasoning : Techniques for Enterprise Systems*. Morgan Kaufman, 1997.

[150] Y.Z. Wei, L. Moreau, and N.R. Jennings. Learning users' interests by quality classification in market-based recommender systems. *IEEE Trans on Knowledge and Data Engineering*, 2005.

[151] M. Wooldridge. Intelligent agents. In G. Weiss, editor, *Multiagent Systems:A Modern Aproach to Distributed Artificial Intelligence*, chapter 1, pages 27–78. MIT Press, Cambridge, MASS, 1999.

[152] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2):115–152, 1995.

[153] M. Wooldridge and S. Parsons. Languages for negotiation. In W. Horn, editor, *Proceedings of the Fourteenth European Conference on Artificial Intelligence (ECAI-2000)*. John Wiley, 2000.

[154] X.F.Wang, X. Yi, K. Lam, and E. Okamoto. Secure information gathering agent for internet trading. In *Selected Papers from the 4th Australian Workshop on Distributed Artificial Intelligence, Multi-Agent Systems*, pages 183–193, London, UK, 1998. Springer-Verlag.

[155] J. Xiaoping. *ZTC: A Type Checker for Z – User's Guide.* Chicago, IL 60604, USA, 1994.

[156] X.Q. Zhang, V. Lesser, and T. Wagner. A proposed approach to sophisticated negotiation. *In AAAI Fall Symposium on Negotiation Methods for Autonomous Cooperative Systems*, 2001.