Automated Service Negotiation Between Autonomous Computational Agents

Peyman Faratin

A dissertation submitted in partial fulfilment

of the requirements for the degree of

Doctor of Philosophy

of the

University of London.

Department of Electronic Engineering

Queen Mary & Westfield College

December 11, 2000

Acknowledgements

I’d like to thank all those people who have supported me throughout this PhD. Firstly, I’d like to thank my sister and brother as well as my guardians in the UK, the Powell family. Thanks also to both of my supervisors Nick Jennings and Carles Sierra for all their time, support, friendship and intellectual inspirations covering the last five years in the UK and Catalonia.

I’d also like to thank all the people of the Artificial Intelligence Research Institute (IIIA), Barcelona. Your kindness, hospitality and wealth of expertise in AI proved invaluable. I had an opportunity to work with great people (Carles Sierra, Luíse Godo, Pedro Meseguer and Pere Garcia).

Last, but not least, special thankyou to all my friends who have helped and supported me over these years in both the UK and Catalonia. I’ve tried hard not to make this a list but the truth is that I’ve been lucky in having good friends who have individually helped me in one way or another throughout the years. Many thanks to David Wheat, Janet Hague, Shayne Hembrow and Clio Potter, Farhang, Kay and Leyla Morady, Dim, Damian and Suzie, Liz, Greg and Gail Mann, Simon Parsons, Phil Turner, Heloise Kirby, Lisa Hogg, Andy Martin and Ho. Thanks also to all the great friends I made in Barcelona. Carles and Juama, Judy, Fransesc, Josep, Nadine, Robin, Alison, Luise, Mireia, Arturo, Juanma, Niki, The Girls, Miguel, Ben and Nina and Sonia. Thanks also to Juan Antonio Rodríguez, Noyda Matos, Marc Esteva, Maite Lopez (and Tony), Lola Canamero, Adriana Zapico, Francisco and Cristina Martin and the people at iSOCO.

Thankyou.

Contents

List of Figures

Abstract

Multi-agent systems are a new computational approach for solving real world, dynamic and open system problems. Problems are conceptualized as a collection of decentralised autonomous agents that collaborate to reach the overall solution. Because of the agents autonomy, their limited rationality, and the distributed nature of most real world problems, the key issue in multi-agent system research is how to model interactions between agents. Negotiation models have emerged as suitable candidates to solve this interaction problem due to their decentralised nature, emphasis on mutual selection of an action, and the prevalence of negotiation in real social systems.

The central problem addressed in this thesis is the design and engineering of a negotiation model for autonomous agents for sharing tasks and/or resources. To solve this problem a *negotiation protocol* and a set of *deliberation mechanisms* are presented which together coordinate the actions of a multiple agent system.

In more detail, the negotiation protocol constrains the action selection problem solving of the agents through the use of normative rules of interaction. These rules temporally order, according to the agents' roles, communication utterances by specifying both who can say what, as well as when. Specifically, the presented protocol is a repeated, sequential model where offers are iteratively exchanged. Under this protocol, agents are assumed to be fully committed to their utterances and utterances are private between the two agents. The protocol is distributed, symmetric, supports bi and/or multi-agent negotiation as well as distributive and integrative negotiation.

In addition to coordinating the agent interactions through normative rules, a set of mechanisms are presented that coordinate the deliberation process of the agents during the ongoing negotiation. Whereas the protocol normatively describes the orderings of actions, the mechanisms describe the possible set of agent strategies in using the protocol. These strategies are captured by a negotiation architecture that is composed of responsive and deliberative decision mechanisms. Decision making with the former mechanism is based on a linear combination of simple functions called *tactics*, which manipulate the utility of deals. The latter mechanisms are subdivided into *trade-off* and *issue manipulation* mechanisms. The trade-off mechanism generates offers that manipulate the value, rather than the overall utility, of the offer. The issue manipulation mechanism aims to increase the likelihood of an agreement by adding and removing issues into the negotiation set. When taken together, these mechanisms represent a continuum of possible decision making capabilities: ranging from behaviours that exhibit greater awareness of environmental resources and less to solution quality, to behaviours that attempt to acquire a given solution quality independently of the resource consumption.

The protocol and mechanisms are empirically evaluated and have been applied to real world task distribution problems in the domains of business process management and telecommunication management.

The main contribution and novelty of this research are: i) a domain independent computational model of negotiation that agents can use to support a wide variety of decision making strategies, ii) an empirical evaluation of the negotiation model for a given agent architecture in a number of different negotiation environments, and iii) the application of the developed model to a number of target domains. An increased strategy set is needed because the developed protocol is less restrictive and less constrained than the traditional ones, thus supporting development of strategic interaction models that belong more to open systems. Furthermore, because of the combination of the large number of environmental possibilities and the size of the set of possible strategies, the model has been empirically investigated to evaluate the success of strategies in different environments. These experiments have facilitated the development of general guidelines that can be used by designers interested in developing strategic negotiating agents. The developed model is grounded from the requirement considerations from both the business process management and telecommunication application domains. It has also been successfully applied to five other real world scenarios.

Chapter 1

Introduction

The topic of this thesis is *interaction*, a temporary or permanent coupling between deliberating entities in a distributed system. The entities of interest in this thesis are digital and inhabit a digital system. The focus of attention is how to computationally model interactions among these digital entities. The need for such models is seen in the current explosion of auction portals (AuctionBot, eBay, Amazon, i2, Rodríguez *et al.*), which together with standardized communication enabling infrastructures such as the WWW, Java and the Knowledge Query Manipulation Language (KQML, (Neches *et al.* 1991, Finin & Fritzson 1994)), allow multiple buyers and sellers, across organizations (business-to-business), as well as individuals (customer-to-customer or business-to-customer), to enter electronic institutions and trade with one another for goods, resources or services, in open and real time electronic market places. In particular, the subject of this thesis is an extension of the current e-commerce technology to bi-lateral interactions/tradings between *autonomous computational units* called *agents* that represent buyers and sellers. Specifically, this work engineers an electronic *negotiation* framework for interactions in electronic commerce between autonomous agents that *bargain* for multi-dimensional goods called *services*. Here this computational-based trading is referred to as agent based electronic commerce of services.

Electronic commerce is just one exemplar of a system that incorporates interaction between computational components. The problem of modeling such interactions in a distributed computational system was first framed within the Distributed Artificial Intelligence (DAI) community. DAI is concerned with understanding and modeling action and knowledge in a collaborative and distributed enterprise consisting of a number of agents (Gasser 1991). Distribution of intelligence among a set of agents is seen as necessary when (Bond & Gasser 1988):

- knowledge or activities are inherently distributed (e.g medical diagnosis or traffic control)
- there is a need for fail-soft degradation through distribution of control
- there is a need to compute solutions to large scale problems given bounded computational resources

- there is a need for reliability, a distributed system can provide cross-checking of solutions and triangulation of results
- there is a need for the integration of existing legacy systems
- there is a need for expert development of separate units through modular knowledge acquisition and management
- the design of a monolithic system is too problematic and costly and instead the costs involved in the development of a large number of simple communicating units is more effective
- there is a need for a greater adaptive power by allowing alternative solutions to be formed from units which have different logical, semantical, temporal or spatial perspectives
- central processing may be too slow compared to enhanced speed through parallel computation

These benefits have been observed in the wide variety of real world problems to which DAI solutions have been applied. These include: problems in manufacturing (*YAMS* (Parunak 1987)), process control (electricity transportation, *ARCHON* (Jennings *et al.* 1996d), nuclear industry (Wang & Wang 1997), spacecraft control (Schwuttke & Quan 1993), (Ingrand, Georgeff, & Rao 1992), climate control (Clearwater *et al.* 1996)), telecommunication systems (feature interaction (Griffeth & Velthuijsen 1994), service management (Faratin *et al.* 2000), (Busuoic & Griffits 1994), network management (Adler *et al.* 1989),(Rao & Georgeff 1990)), air traffic control (Ljungberg & Lucas 1992), traffic and transport management ((Burmeister, Haddadi, & Matylis 1997), (Fischer, Mủller, & Pischel 1996)), information filtering and gathering ((Sycara *et al.* 1996), (Chen & Sycara 1998), (Etzioni 1996), (Liberman 1995), (Kautz, Selman, & Shah 1997)), electronic commerce ((Chavez & Maes 1996), (Krulwich 1996), (Doorenbos, Etzioni, & Weld 1997), (Tsvetovatyy *et al.* 1997)), business process management ((Faratin, Sierra, & Jennings 1998), (Jennings *et al.* 2000a), (Jennings *et al.* 2000b), (Huhns & Singh 1998)), entertainment (Grand & Cliff 1998), and medical care ((Hayes-Roth *et al.* 1989), (Decker & Li 1998)).

These problem domains are suitable for DAI technology (also known as agent technology (Bond & Gasser 1988)) because they exhibit one or more of the above features. For example, a manufacturing process is inherently a distributed system where production chains, or its components, can be represented as computational agents whose capabilities are captured using plans, and who share these capabilities through negotiation. Similarly, control systems can detect, diagnose and remedy problems if control subprocesses are delegated to agents that not only provide cross checking of results, but also form solutions to problems from different and novel perspectives and exhibit graceful degradation in case of node(s) failure(s).

Although distribution can be beneficial, it gives rise to the following questions that need to be addressed (Bond & Gasser 1988):

1. How to formulate, represent, decompose and allocate the problem and how to synthesis the results among a group of intelligent agents.

2. Sub-problems may interact which requires the agents to communicate and interact. If interaction is required then the problem arises of how to model the language and the protocol of this interaction.

3. how to achieve global coherency from local processing. That is, how to ensure that agents act coherently in making decisions or taking actions, reasoning about the non-local effects of local decisions and avoiding harmful interactions.

4. If there is a need for interaction and coordination, then how should agents represent and reason about the actions, plans and knowledge of other agents.

5. How are agents to recognize and resolve and/or synthesize disparate view points on a sub–problem. These conflicts can be caused either by uncertainty in the world, different reasoning procedures or limited resources.

6. How to actually engineer and constrain practical DAI systems through the design of platforms and methodologies.

Each of the above problems emphasize different facets and perspectives of a DAI system. The first problem is the central problem in DAI and is centered on the *problem* the system is designed to solve in a distributed manner. In addition, distributed problem solvers need coordination (the third problem), agent communication languages (the second problem), and agent reasoning mechanisms (fourth and fifth problems). Finally, there is a need to engineer a distributed system that implements the solutions to the above problems. As Gasser notes, the solutions to these problems are not independent:

> ...different procedures for communication and interaction have implications for coordination and coherent behaviour. Different problem solving and task decompositions may yield different interactions or agent-modeling requirements. Coherent, coordinated behaviour depends on how knowledge disparities are resolved, which agent resolves them, etc (Gasser 1991).

Given this, it can be seen that the coordination issue is a quintessential problem in DAI (Decker 1995). To this end, the contribution of this thesis is the development of a formal model of agent reasoning that attempts to address the coordination problem.

1.1 Aims of the Research

The central aim of this thesis is a formal specification and evaluation of a coordination framework for computational units, called agents, that buy and sell services from one another and operate in either open or closed distributed systems (defined below). Here a coordination framework is defined as a collection of three components:

1. the public rules of behaviour specifying the permissible actions agents can take in the course of interactions

2. the subject of interactions

3. the deliberation mechanism that assists agents in making decisions

These components roughly specify when to interact, what to interact over and how to interact, respectively. The major contribution of this work s a formal model of the third component. This component will be referred to as a *wrapper* layer because it is seen as *supplementing* an asocial domain problem solver with additional functionality that the domain problem solver was not designed for in the first place, i.e. to interact. The wrapper can also be thought of as a “plug and interact” module of systems that need to interact with other systems.

The subject of agent interactions are services. Services capture and represent in an abstract way, similar to methods in object oriented paradigm (Coad & Yourdon 1991), the local capability of agents in performing tasks. There are numerous examples of services in the real world which individuals need. Database validation, financial forecasts, medical diagnosis, fault prediction are but a few examples where the capability of an agent is represented as services it can provide to others who need it. Services, in a similar manner to methods, are reusable for other types of problems that require the expertise of that agent. However, agents differ from objects in that their services can not be invoked by a simple procedure call because, as will be shown below, they are assumed to be autonomous. Therefore, the agent must be persuaded to perform its service(s). Access to services in real social systems is gained through various means such as long term contracts (for example, companies often have long term contracts with companies that provide fiscal forecast information) or conventions of organizations (for example, access to shared and public services such as medical expertise, is still determined not by who can pay most, but on need basis). However, the type of persuasion considered in this thesis is *negotiation*:

Definition 1 *a process by which a joint decision is made by two or more parties. The parties first verbalize contradictory demands and then move towards agreement by a process of concession making or search for new alternatives (Pruitt 1981).*

In summary, the aim of this thesis is the development of a coordination framework that specifies: i) the public rules of behaviour during the negotiation, ii) the services which agents “produce” and “consume” and iii) the deliberation mechanisms that the agents use during negotiation. This coordination framework is designed for both closed and open systems. In this thesis, a closed agent system (also referred to as a

Distributed Problem Solving (DPS) system (Yang & Zhang 1995, Durfee & Rosenschein 1994)) is characterized by a *central* designer(s) undertaking the following steps in the system design methodology:

1. definition of the global problem(s)

2. mapping and assigning subproblems and resources, either dynamically at run-time or statically at design-time, to agents

3. central configuration of *all* the agents, specifying their agent's behaviour in the course of interactions

4. using an agent communication language to allow the agents to solve the problems in step1

This methodology is problem centered (step 1); a central designer creates a fixed and static society of computational agents (step 2), who interact repeatedly (exchanging goals, plans or information) using a communication language (step 4), to collectively solve a well structured and objective global problem. Agents are often homogeneous in architecture, languages and reasoning (step 3), and are cooperatively motivated to help one another to solve the global problem at hand. This benevolent agent attitude directly follows from the assumption in closed systems that agents share a common goal. Thus agents cooperate with one another because they are aware of the fact that they share a common goal. Any conflicts are subjective, arising as a consequence of an incomplete or incorrect local view of the world, rather than objective contradictory interests.

Conversely, an open agent system (Hewitt & de Jong 1984) (also referred to as a *Multi-Agent System*, MAS (Bond & Gasser 1988, Durfee & Rosenschein 1994, Durfee & Lesser 1989)) is characterized by a *number* of designers undertaking the following steps in the system design methodology:

1. either defining the global problem or allowing the problem to dynamically emerge

2. nominating/selecting (pre-existing) autonomous agents to enter interactions

3. configurating of *your* agent(s)

4. using an agent communication language to allow the agents to identify conflicting issues and solve problems in step1

Open environments are better characterized as *encounters*, where pre-existing agents come together infrequently to solve a problem, trade goods, or, alternatively, where problems emerge dynamically "on the fly" in the course of interactions. This interaction centered, as opposed to problem centered, stance means that the agent society is more dynamic. Agents can come and go. There is no globally shared goal(s), hence the motivations in interactions are more selfish. There is a large degree of uncertainty about the other agents.

The agents themselves are heterogeneous in architecture, languages and reasoning procedures. The problem structure itself is ill defined, no objectively correct solution exists and instead preferences are given more importance. Under these circumstances, assumptions about the system (such as agents, resources, information and goals) are not only difficult to make, but may also often be invalid.

The characterization of agents as selfishly and autonomously pursuing multiple goals has a number of important implications. The pursuit of individual goals is beneficial in that it decouples agents from one another. Thus, self interest, as a behaviour guideline, encourages separation between individual and group problem solving. This is useful when an agent is vulnerable to the malicious behaviour of others, or when there is a need to reduce the influence of agents who have erroneous information or deliberation models. Also the assumption in MAS that agents may have multiple, and at least partially, conflicting goals produces social dilemmas or real conflict, which cannot be resolved simply by increasing the awareness of an agent through information exchange. Finally, the autonomy assumption means that agents can create and pursue their own goals in a self-interested manner. The decision of whether to adopt the goals of others is based on whether these adopted goals contribute to changing the current world state into a *personal* desired and motivated state.

This thesis aims to develop a specification of a coordination framework (the rules, objects and deliberation components of interactions) that can operate in both closed and open systems; usable by both a closed system designer, to define each agent's interaction capabilities (step 3 in the closed system design), or, alternatively, by an open system participant who would like his/her agent to interact with other pre-existing agents, designed by other designers (step 3, in open system design). Thus, the coordination framework should be easily configurable and applicable to different types of systems. This configurability is motivated by the principles of re-usability and flexibility. Re-usability is achieved by i) making as few commitments to the agent architecture as possible, ii) dissociating interaction decisions from the protocol of interactions and iii) emphasizing the notion of services. Flexibility, in turn, is sought by avoiding unreasonable or strong assumptions that limit the applicability of the framework to a single domain or agent architecture. Specifically, this requirement amounts to the design of a framework that does not assume the agent is unbounded in computational resources or information (Bond & Gasser 1988). This is because real world environments are often characterized by uncertainty and limited computational resources which need to be devoted to solving the domain problem the agent was actually designed for in the first place. In fact, interaction is an added cost to the agent in not only computation, but also communication. Additionally, not only can communication be expensive, but it can also be unreliable. Prolonged communication may also cause non-terminal chains of beliefs and goals updates because as the length of communication increases so does the chain of beliefs and goals that support the deliberation in the course of interactions (Huhns & Stephens 1999).

Therefore, the aim is to *design and engineer a re-usable and flexible computational coordination framework for both open and closed distributed and multi-agent systems.* Like computational auctions (Varian 1992, Vulkan & Jennings 1998, Sandholm 1999), where agents interact and trade with one another according to normative rules of an electronic institution (Rosenschein & Zlotkin 1994), a computational negotiation framework is sought that permits individual agent designers to specify negotiation strategies for the trading of services, for both closed and open systems, given the rules of interactions. As will be shown, auctions are computationally different to negotiation and a different framework of negotiated interactions is necessary (sections 3.1.8, 3.2.8). The stance adopted in this thesis is that the framework should formally, and minimally, represent:

- the set of agents involved in negotiation
- the conflict object(s)
- the public rules of interaction
- the strategic resolution decisions available to an agent

Note the last aim—*specification* of the strategic choices an agent has in conflict resolution. This relates to the "configuration" step in both the open and closed agent system design methodology (step 3). A framework, as opposed to a unique solution, is sought that makes *available* to agent designers different types of negotiation decision strategies. In this sense, the framework is descriptive and the designer is free to "configure" the agent according to some objective. However, in order to assist the designer, the developed resolution strategies are empirically evaluated in a number of environments (see chapter 5).

1.2 Functional Architecture of the Coordination Framework

The above requirements are captured in the functional architecture of the coordination framework/*system* shown in figure 1.1.[1] The coordination system consists of:

- the coordination deliberation module (the coordination model, the service description and the agent knowledge bases AM (Acquaintance Model) & SM (Self Model), defined below, in figure 1.1)—together these modules are referred to as the *negotiation wrapper*.
- the communication protocol (agent communication protocol).

The communication functionality of the coordination system is supported by the interaction enabling infrastructure (labelled middle-ware (Coulouris, Dollimore, & Kindberg 1994, Brenner, Zarnekow, & Wittig 1998) in figure 1.1). The negotiation wrapper is seen as assisting the domain problem solver in interactions.

[1] The terms interaction and coordination will be used interchangeably throughout the thesis.

The domain problem solver is informally defined as an autonomous entity that has knowledge (represented as the domain information model in figure 1.1) about the domain in which it operates, but that needs the assistance of others (as services) in solving its problems. The coordination architecture, based on ARCHON

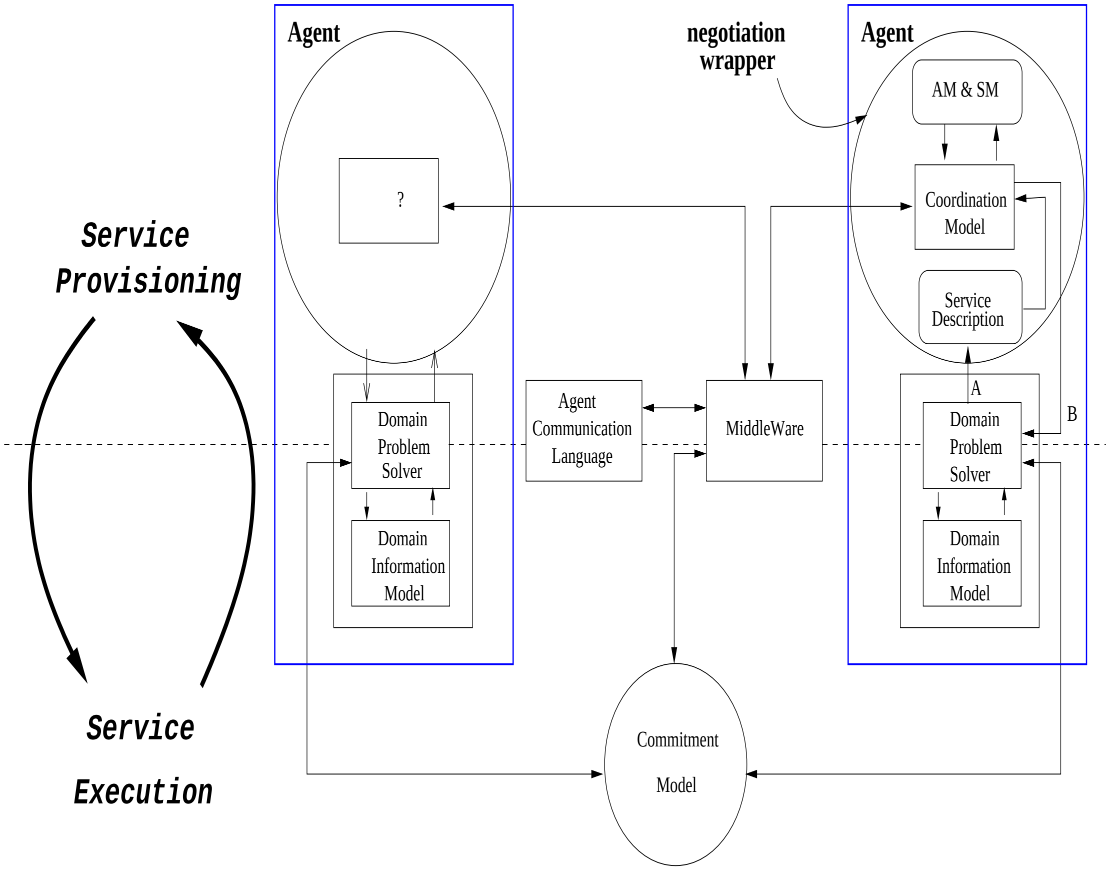

Figure 1.1: Functional Specification of the Interaction System

(Jennings *et al.* 1996d), is divided into two parts, representing the service provisioning and service execution phases of agent activities (shown as the division marked by the dotted line in figure 1.1). Service provisioning is defined as the processes involved in procuring the necessary resources required to perform an activity. Service execution, in turn, is defined as the actual performance of the provisioned activity. This division expresses the differences between the processes involved in provisioning a service from those involved in its execution. The processes involved in provisioning are procurement processes involving scheduling local actions, identifying those actions/tasks that can not be performed locally, contacting the appropriate service provider(s), followed by negotiating the required service. The processes involved in service execution are more like management activities involving monitoring the agreed service execution plan (circle marked *Commitment Model* in figure 1.1) and initiating recovery procedures when execution has failed or is predicted to fail. The division between these two types of processes is informally captured as the *service life-cycle* (depicted as the service provisioning service execution cycle in figure 1.1). The service life-cycle consists of firstly provisioning and then executing a service. Another episode of service provisioning may be initiated if the current execution fails. The focus of this work is on a negotiation model for the service provisioning phase. Therefore, the subsequent exposition will concentrate solely on the service provisioning phase of the life-cycle.

Figure 1.1 shows two domain problem solvers, and their associated domain information models (the boxes labelled, *Domain Problem Solver* and *Domain Information Model* respectively). The negotiation wrapper is depicted as an oval that is connected to the domain problem solver. The exposition of the negotiation wrapper will concentrate on the internal processes and structures of the agent on the right hand side (the circle containing three boxes labelled *AM & SM*, *Coordination Model* and *Service Description*). Assume for now that this agent is the client of a service. Only one agent will be discussed because the negotiation deliberation component of the wrapper does not make any assumptions about the architecture of the other interacting agent. Thus, heterogeneous agents can inter-operate, as long as they obey the rules of the protocol specified by the Agent Communication Language. In fact, from the perspective of a very simple agent (unable to model others), the other agent can simply be viewed as a black box (box labelled with a question mark) that receives inputs, in the form of messages, and generates outputs, again in the form of messages.

Furthermore, note that the domain problem solver is separated from the wrapper layer by a *Service Description* layer. A service description is defined as an enumeration of the dimensions of a service (or identification of the issues involved in the provisioning of a service) and the specification of preferences the domain problem solver has over each of these identified dimensions. This description of a service is then “handed” to the wrapper to provision. This design philosophy is also shared by the work of Kraus:

> There are two aspects to the development of agent architectures: what is the architecture of each agent and how do they interconnect, coordinate their activities and cooperate. There are many approaches to the development of a single agent. ... We provide a separate module for the strategic negotiation, and thus, we are willing to adopt any definition or model of a single agent. Our only assumption is that the agents can communicate with each other and that our negotiation module can be added to the agents (Kraus 2000).

The domain problem solver initiates service requests with the wrapper via this service description layer (link labelled *A*) during the service’s provisioning phase, describing the issues involved in negotiation as well as the domain problem solver’s preferences over these issues. Successful negotiation with the other server agent will result in a contract that is then passed back to the domain problem solver from the wrapper (link labelled *B*). During, or previously, to the service request, both the domain problem solver and the coordination module read and write to their information models, labelled *Domain Information Model* and *AM & SM* respectively. The *AM & SM* are the wrapper's repositories for knowledge about itself and others in its environment respectively (Jennings *et al.* 2000b). The *SM* maintains information such as the services it can provide, the resources available to perform it, and its current schedule of activities. In its acquaintance model (AM), the agent stores information about the existence and known capability of other agents.

The above view of provisioning is agent-centric, concentrating on the internals of the agent. However, there are also inter-agent processes and structures involved. All inter-agent communication is physically routed via a suite of middleware services that assist distributed computation (box labelled *MiddleWare*). These services, possibly provided by other agents, may include: yellow and white page directory services, assisting agents in locating one another; platforms for message routing services (such as DAIS (DAIS 1984) or ORBIX (orbix 2000)); authentication services; security services; mediation services and brokerage services (see (Vogel 1996) for a full description of middleware services). The implemented middleware architecture for communication of this research has been a combination of DAIS (DAIS 1984) and the FIPA Open Source routing platforms (FIPA-OS 2000).

Finally, the syntax and pragmatics of messages are checked against the normative rules of the communication protocol, stored in the agent communication language component of the coordination system, and correct messages are sent via the middleware to the intended recipient. Otherwise an error is flagged and the sender is notified of the divergence from the rules of the protocol.

The details of the negotiation wrapper (the coordination module and its associated information models and service description), and the agent communication language modules of the architecture are revisited in more depth in chapters two, three and four. What constitutes an agent is discussed next, prior to an in-depth discussion of focused concepts such as coordination, interaction and negotiation.

1.3 Agents and the Coordination Problem

An agent definition is presented in this section followed by an in-depth examination of the problem of coordination, its definitions, rationale, properties and types.

1.3.1 Agent Definition

Agents, rather than a *group* of agents, are the kernel of the investigation reported in this work. The term agent, however, has been the subject of much debate recently, ranging from definitions that allow the inclusion of almost all possible objects, to definitions which only permit a very closed set of possibilities as candidates for agency (see (Russell & Norvig 1995), (Maes 1995), (Hayes-Roth 1995), (Wooldridge & Jennings 1995) for some definitions).

In this work, an agent is defined as a *combination* of the domain problem solver and the wrapper (where the latter component is concerned with providing interaction capabilities and communication knowledge for the former):

$$agent = domain\, problem\, solver + wrapper$$

The domain problem solver is assumed to be capable of symbolically representing and reasoning about its internal state utilizing its domain knowledge. Reactive agents (Brooks 1991) are therefore excluded from this research. The domain problem solver is also assumed to be autonomous. Stated simply, autonomy means that the agents operate without the direct intervention of humans or others, and that they have some kind of control over their actions and internal state (Castlefranchi 1995). In this work, autonomy amounts to the wrapper having local control in selecting its strategies in negotiation. Indeed, autonomy is a necessary condition for negotiation since agents cannot be made or ordered to perform task(s) by other peer agents.[2]

Finally, agents are assumed to be capable of being both self or group motivated when making decisions at the interaction phase of their problem solving. In this thesis selfishness is informally defined as the achievement of one's goal(s) independently of the other(s) goals. On the other hand, group motivated decisions are defined as achievement of one's own goal(s), but in a manner that is helpful to others' goal(s). This local and global goal motivational stances of an agent are given more concrete definitions in terms of maximization of individual and social welfare in proceeding chapters when quantitative models of negotiation are introduced. The choice of which attitude to adopt is not hardwired into the agent architecture, rather it is a function of the agent's environment. As was seen in section 1.1, the motivations of agents have been one of the key features that has been used in order to differentiate DPS from MAS.

1.3.2 The Coordination Problem

In this section, the concept of coordination is examined from a DAI perspective (see chapter 3 and (Decker 1995), (Kraus 1997b), (Walton & Krabbe 1995) for a more detailed treatment from other related fields). This exposition will lay the foundations for introducing different models of coordination in subsequent chapters.

1.3.2.1 Definitions of Coordination

Holt informally defines coordination as "a kind of dynamic glue that binds tasks together into a larger meaningful whole" (Holt 1988). More specific definitions place the main emphasis on the *outcome* of coordination in creating collective actions. For example, Bond and Gasser define coordination to be:

> ... a property of interaction among some set of agents performing some collective activity (Bond & Gasser 1988).

[2] Autonomy is often a feature of the organizational structure of the society. Thus, whereas a peer can not order other peers to perform a task, in a master-slave relationship orders are permitted, and often practiced in real social systems, to ensure coordinated actions that incur little or no communication and deliberation load (Scott 1987).

This definition is centered on the outcome of coordination. However, it is too abstract to be of any use operationally. For example, the notion of collective activity alludes to the existence of a shared goal to act collectively, since for collective activity agents must share the goal to collaborate with one another in the first instance (Bratman 1990). Such goals are explicitly included in definitions by Singh and Malone:

> The integration and harmonious adjustment of individual work efforts towards the accomplishment of a larger goal (Singh 1994).
>
> The act of managing interdependencies between activities performed to achieve a goal (Malone & Crowston 1990).

That is, with these views, coordination is the *process* of aligning and adjusting agents' actions to manage interdependencies, where success leads to achieving some global system-goal. Although the concept is given a more concrete definition in terms of both outcome ("goal") and the processes involved, terms such as "work efforts" or "integration" or "management" do not constrain different interpretations. For example, which entity is responsible for managing the interdependencies—the individuals or a centralized controller? Likewise, it is not clear what is the object of "work effort"; an agent's goals, plans or desires, or some other construct? The following two definitions offer an alternative perspective on coordination, emphasizing a *local*, rather than a central, locus of coordination:

> Coordination, the process by which an agent reasons about its local actions and the (anticipated) actions of others to try and ensure the community acts in a coherent manner,. . . (Jennings 1996), p.187.

and additionally, a process whose domain of operation is the satisfaction of *preferences*:

> ... a solution to a coordination problem constitutes an *equilibrium*, a compromise that assures somehow "maximal" attainment of different interests of all involved individuals (Ossowski 1999).

The process of coordination is also central to Jennings' definition. However, whereas the previous definitions were ambiguous about how it was achieved, in this definition, coordination is actively brought about via *local*, rather than some centralized, explicit *reasoning* process of each agent. Likewise, Ossowski's definition emphasizes the local locus of control in coordination. However, in addition to this, the "work effort" is the *conflicting interests* of individuals that need to be resolved in coordination. As will be shown in later chapters, Ossowski's definition belongs to game theoretic models of coordination that emphasize notions of *solutions* and *equilibrium* (an emergent property that is coordination).

Finally, whereas all the above definitions are based on achieving collective actions, Huhns argues that although coordination is a property of collective actions, it is not an all or nothing property. Rather it can exhibit *degrees* of satisfaction:

> ... a property of the system of agents performing some activity in a shared environment. The degree of coordination is the extent to which they avoid extraneous activity by reducing resource contention, avoiding live-lock and deadlock, and maintaining applicable safety conditions (Huhns & Stephens 1999), p.83.

The definition of coordination is made more complex because the perspective of the definition needs to be unambiguously determined. Generally, when the system of agents is viewed from a behaviouristic perspective (by observing the behaviour of the system only), then it is difficult to assess whether agents have engaged in coordinated action (Jennings 1996). Agents may have indeed coordinated their actions, but the resulting system behaviour may be incoherent, due to erroneous models, lack of information or insufficient resources. Conversely, the system may exhibit coherent collective actions, but the agents did not actually intend to coordinate their actions (see (Searle 1990) for a description of the problem). For these reasons, some researchers in the field have proposed that a satisfactory definition of coordination cannot be based on behaviourism alone (Castlefranchi & Conte 1997). Instead, a satisfactory theory of coordination must account for and be based on *intentional attitudes* such as beliefs as well as higher order attitudes (or pro-attitudes) such as intentions and desires of the agents (Dennett 1987, Castlefranchi & Conte 1997, Wooldridge & Jennings 1995).

In general, the definitions all share the point that the outcome of coordination is coherent, collective actions. However, there is no consensus over how, and by whom, coordination is achieved, nor what is the object of coordination. The proposed definitions are informally summarized as:

> the *coordination problem* consists of *composing* (relating, harmonizing, adjusting, integrating) some *coordination objects* (tasks, goals, decisions, plans) with respect to some *coordination process*, which solves the coordination problem by composing co-ordination objects in line with the coordination direction (Ossowski 1999).

This view of coordination will be used as the working definition throughout this work. Finally, in this work a distinction is made between processes that help bring about coordinated action and the processes that maintain coordination. This distinction is reflected in real social systems where the processes that bring about “signing of a deal” are separate from processes that maintain “honouring of deals” (Scott 1987). The work reported here is primarily an attempt to address the processes necessary for achieving coordination, although structures are provided to assist the second stage of coordination.

1.3.2.2 Rationale for Coordination

Coordination is needed when there are interdependencies between agents’ actions, between local actions and some global criteria that needs to be satisfied, or when there are differences in expertise or levels of resources (Bond & Gasser 1988, Huhns & Stephens 1999). Action dependencies (Bond & Gasser 1988) occur when the local actions of one agent directly or indirectly have an effect on the actions and plans of others. (Jennings 1996) gives the following examples to illustrate interdependency between agents. Action dependencies arise when the local activities of agents contribute to the solution of a larger problem (e.g. building a house), there is a need to coordinate each individual action, since the local decision of one agent directly impacts actions of other community members. Interdependencies in activities may also arise when there is contention for resources in problem solving (e.g. a hammer may be needed by two agents simultaneously to perform their tasks or a bridge that must be used by two convoys of trucks traveling in opposite directions). Likewise, local actions may need to satisfy some global criteria (e.g. the budget for building a house cannot exceed $\pounds 30000$). Furthermore, in many types of problems no one agent has sufficient competence, resources or information to achieves its goal(s) (e.g. successful diagnosis of a disease often involves many different sources of expertise, information and equipment). Generally, coordination in most of these contexts closely resembles a distributed optimization problem used for ordering individual tasks, selecting who and how to accomplish them, as well as the resources needed for their satisfaction (Decker 1995, Ossowski 1999). Another view is that the outcome of coordination can be divided into three basic classes, reflecting decisions at three levels: specification of what goals or objectives to achieve (creating shared goals); planning of how to achieve them (expressing potential sets of tasks to achieve goals); scheduling of when to perform the actions (task assignment, shared schedules and resource allocation) (Decker 1995).

In the above cases, coordination functions to *inform* local activities. Coordination is an informing process for the types of problems that have concerned the classic distributed planning community, where interdependencies exists among agents' activities (Durfee 1998, Durfee & Lesser 1989, Georgeff 1983, Corkill & Lesser 1983, Durfee, Lesser, & Corkill 1988). Thus the source of conflict is the lack of knowledge in producing effective local actions. In such cases, coordination is used as a method of informing individual agents of the plans of others, who then integrate their partial plans into a coherent global plan. Furthermore, agents are assumed to be helpful and the informing process assists agents in cooperatively synthesizing a solution to the given problem.

However, agents may not always cooperatively agree to perform a task when asked by other agents. They may need *convincing*. This is necessary when the helpful assumption is dropped and the object of coordination is the individual preferences of agents. For example, agents may no longer share the same goal, and instead they may have goals that are mutually exclusive. For example, a buyer wants to buy a good at a low price, whereas a seller wants to sell at a high price. Alternatively consider the example of two trucks wanting to simultaneously cross a bridge that can only support one truck crossing at a time. In both examples there are no shared goals. In fact, the goals of the agents are mutually exclusive. The goals of an *individual* may also be mutually exclusive (e.g. company A wants to increase wages to satisfy its workers, but also wants to cut down on expenditure). In such cases, coordination may involve more than informing others of plans or goals (one truck driver can not simply state its intention that it intends to use the bridge first. It must convince the other driver of this schedule). Indeed, under the non-cooperative assumption even the validity of information can not be taken at face value since agents may be untruthful (Rosenschein & Zlotkin 1994).

In such contexts, coordination is needed because of *conflicts* of interests. In the case of helpful agents, coordination resembles a distributed optimization problem (optimally ordering tasks, resourcing, assigning and scheduling of tasks to agents). In the case of selfish agents, a coordination mechanism is needed that more closely resembles a distributed conflict resolution problem because optimization of activities and resources may be an intractable problem given that information may be incorrect (selfish agents may be untruthful about the information they communicate), uncertain (information is not publicly available hence agents have to make uncertain decisions about actions of others) and partial (no one agent has a complete view of the overall problem). Therefore, optimization of the overall problem becomes intractable. The problem then becomes how to resolve each individual's preferences in the collective activity.

Finally, even if coordination is not needed (actions are independent and resources are plentiful) it may still be beneficial if agents coordinated. For example, information discovered independently by one agent can be transmitted to others which can be used to reduce the complexity of their search (Decker 1995). As will be shown in section 3.2.1, negotiation based on this assumption has been popular with the work of Rosenchein and Zlotkin.

1.3.2.3 Properties of Coordination

The properties, or characteristics, of coordination are closely related to the definition of coordination from section 1.3.2.1, and are meant to capture, in some objective way, what the system as a whole should exhibit for it to be considered coherent. Operational definitions of what is a coherent action have yielded several criteria along measurable objectives such as solution quality, efficiency, clarity and graceful degradation (Bond & Gasser 1988). Specifically, a coordinated system must (Corkill & Lesser 1983):

- ensure all the necessary overall problems are included in the activities of at least one agent—*coverage*
- permit interactions between activities to be developed and integrated into an overall solution—*connectivity*
- ensure the above objectives are achieved within the available computational and resource limitation—*capability*

Malone, in addition to the above, proposes *flexibility* and *efficiency* tradeoff criteria for evaluating the success of coordination (Malone 1990). This criteria can be used to differentiate one type of system that is highly structured, with formalized procedures for all possible eventualities, to systems that are loosely coupled structures that depend on massive amounts of informal communication and mutual adjustments to adapt to rapidly changing and complex environments.

Finally, quantitative models of coordination specify properties for both the outcome and the process of coordination. In these models, which will be described in more detail in chapter three, satisfactory coordination should be efficient (either in the speed of convergence to coordinated behaviours or in the quality of the coordinated outcome, or both) and stable (where the individual's strategy of interaction is self enforcing and deviations from this are irrational (Binmore 1992)). Additionally, the coordinating process itself should not treat individuals differently. This symmetric treatment of agents is a desirable property because a coordination solution that treats one agent more preferentially than another is unlikely to be adopted by the agent who fares worse. Furthermore, to maintain the benefits of the distribution (section 1), it should be distributed, requiring no central decision maker (Rosenschein & Zlotkin 1994). These properties are then used as a benchmark to evaluate different coordination solutions (Rosenschein & Zlotkin 1994).

1.3.3 Types of Coordination

There are numerous different types of coordination techniques (where each type differs in its rationale, methodology and effects). Therefore, for comparison purposes, Walton and Krabbe defined the following interaction set based on the initial context and the joint and individual aims of the concerned parties: (Walton & Krabbe 1995) [3]

- Persuasion—Persuasion begins with the identification of a conflict and a mutual adoption of the goal to resolve this conflict. The primary motivation of each agent is to modify the belief of the opponent while avoiding revision of the agent's own beliefs. However, each agent implicitly acknowledges the willingness to modify its own beliefs.

- Inquiry—In inquiry the aim of each agent is the shared aim of all agents, which is to substantiate or derive a proof for a claim.

- Deliberation—Deliberation is not initiated from a conflict, but is rather directed from a need for action. The aim of deliberation is to jointly arrive at a decision or form a plan of action. Like negotiation and persuasion, deliberation is a non-cooperative interaction in that agents attempt to reach a plan of action or decision which benefits themselves.

- Negotiation—The interaction type used for the problems addressed in this research is negotiation which, like persuasion, but unlike deliberation, is initiated from a conflict of interests. Furthermore, similarly to persuasion, negotiation is motivated by a need to make a deal while selfishly maximizing

[3] Only the relevant classes of interactions are included here. See (Walton & Krabbe 1995) for a more formal treatment of these and other types of interactions.

> personal goals. However, whereas the aim of a persuasion dialogue is to reach an agreement, in negotiation dialogue it is not a necessary condition to reach a settlement—other than agreeing to a particular deal. Thus the beliefs of each agent may still remain diametrically opposite at the end of negotiation. It is in this sense that negotiation is viewed throughout this thesis.

The object of interactions, in this research, over which agents have conflicts is called a *service*. In service-oriented negotiation, one agent (the client) requires a service to be performed on its behalf by some other agent (the server).

A service is a solution to a problem. It is formulated and assigned to agents who then act as experts in solving that type of problem. Examples include diagnosing a fault (performing a task), buying a group of shares in the stock market or allocating bandwidth to transmit a video-conference (gaining access to a resource). Agents that then require that expertise must interact (or negotiate) with agents who own the expertise. Thus solutions to problems are accessed via a computational economy, where the activities of interest are described in terms of the production and consumption of services (Mullen & Wellman 1995). Services partially capture what Malone calls the *"fundamental components of coordination"*, the allocation of scarce resources and the communication of intermediate results (Malone 1990). In this thesis, a service is an abstraction of an agent's *capabilities* to perform *both* tasks and provide resources. As will be shown in proceeding sections, a considerable number of models of negotiation have been developed for either the problem of task allocation (for example, the Contract Net Protocol, see section 3.2.3), where negotiation is viewed as connecting and gaining access to capabilities of other agents (such as security expertise), or resource allocation, where negotiation is establishing usage rights to a shared resource that is owned *mutually* (such as a bridge). This dichotomy is principally due to the process that maps the given problem into a MAS (this process will be referred to as *agentification*). Generally, although tasks are assigned to agents, the associated resources necessary to perform the tasks can either be mutually or privately owned. In either case, agents must interact with one another and establish usage rights of tasks as well as of mutually or privately owned resources. Note, that the choice of agentification (assignment of services and resources to achieve these services) directly influences the coordination wrapper, in terms of coverage, connectivity and capability of the agents to the problem (see section 1.3.2.3). For example, an inappropriate assignment of resources to an agent to perform the service will reduce the effectiveness of the negotiation wrapper. This is because if the resources to perform a service s are provided by several other agents, then the agent that wants to provide s to another agent must engage in a number of other negotiations with providers of resources for s.

To achieve one of the aims of this research (a domain independent negotiation wrapper) the process of agentifying the problem must not only assign individual tasks to agents, but must also assign the resources necessary to perform the tasks. Thus, ownership is assigned over both tasks and resources and specifies the roles of an individual over a service, specifying whether the agent is a provider or consumer of a service. Access to these services is then achieved through trading/bargaining over the service and its multiple features, such as its price, quality, start-time, as well as other service features.

Moreover, in service-oriented contexts, negotiation involves determining a *contract* under certain terms and conditions. A contract is informally defined as:

> a statement of the rights and obligations of each party to a transaction or transactions. A contract, familiarly envisaged, is a formal written statement of the terms of the transaction or relationship: a house purchase or a pop star's deal with a record company (Bannock, Baxter, & Davis 1992).

Thus, agents negotiate for services, defined as multi-dimensional goods, and successful negotiation results in agreements in the form of contracts.

As will be shown in later chapters, the characterization of objects of interaction as services permits abstraction and decoupling of coordination reasoning from the problem domain at hand. The latter problem is handled by the domain expert who then specifies the service(s) it requires and its preferences over the service(s) to the wrapper. Contracts, in turn, *explicitly* model commitments made at the end of successful interactions.

An agent's motivation was a central classification criteria in the above coordination taxonomy. As was shown previously, this attribute has been instrumental in classifying DAI approaches and their techniques into closed (DPS) and open (MAS) system paradigms. Two application domains, one an example of a closed system and the other of an open system, are presented next. The domain problems of these two applications have been instrumental in grounding the research direction of this thesis and have been fully implemented as systems of multiple interacting agents.

1.4 Exemplar Problem Domains

This section presents two application domains, business process management (section 1.4.1) and telecommunication service management (section 1.4.2), that have jointly motivated and grounded the design of the interaction wrapper. These two application domains can be viewed as typical real-world exemplars of applications that are well suited to an agent-based approach (i.e. they exhibit a number of the features described in section 1). See (Jennings *et al.* 2000a), (Jennings *et al.* 2000b), (Jennings, Norman, & Faratin 1998), (Faratin, Sierra, & Jennings 1998), (Sierra, Faratin, & Jennings 1997), (Norman *et al.* 1996), (Jennings *et al.* 1996c), (Jennings *et al.* 1996b), for publications on the business process management (ADEPT) project and (Faratin, Sierra, & Jennings 2000), (Faratin *et al.* 1999b), (Faratin *et al.* 2000), (Sierra, Faratin, & Jennings 1999) and (Faratin *et al.* 1999a) for publications on the telecommunication service management project. In addition to these application domains, the developed wrapper:

1. has been deployed in a European Union project (ESPRIT 27064), called CASBA (Competitive Agents for Secure Business Applications) (CASBA 2000). CASBA is an e-commerce marketplace where agents buy and sell items (travel packages for example, as well as business to business applications). Here the wrapper has been used to model the decision making functionalities of the agents.

2. has been used to demonstrate negotiation within Service Impact Analysis and Service Level re-negotiation within Nortel Networks (property of Nortel Networks, hence no public document exists for referencing). Service impact negotiation relates to network level negotiation for the provisioning of resources for the network to recover from the impact of a failure. Agents representing different nodes within the network negotiate using the wrapper to recover from the network failure. The wrapper has also been used to dynamically re-provision telecommunication service failures with the affected customer at the service level. Agents representing the service provider and effected customers utilize the negotiation wrapper to re-negotiate the committed Service Level Agreement to enable a continued service.

3. has been incorporated as a generic component into the agent framework used within Nortel for developing multi-agent systems. The wrapper technology within the agent framework has been used to construct a number of concept demonstrators, including:

   (a) Security Negotiation: utilizing the negotiation wrapper to enable the required security level to be established between calling parties depending on their individual requirements.

   (b) Shuffle project (Shuffle 2000). The wrapper is also intended to be used in the European Union's Fifth Framework Project Shuffle (*An agent based approach to controlling resources in UMTS networks*). The aim of the project is to use negotiating agents in a resource configuration system that dynamically allocates radio and associated fixed network resources in third generation mobile communication systems. Third generation mobile systems are seen as being the technology to bring the new broadband services being developed for the Internet (and for broadband networks in general) to the mobile user. However, providing flexible, higher bandwidth services in a mobile environment leads to increased complexity in resource control and resource management because of the variable bandwidth requirements of the applications, the new radio architecture and the varying demands on the fixed part of the infrastructure. Such complexity requires the use of sophisticated control and management techniques. Negotiating agent technology is intended to be used to manage this complexity.

Together these seven applications of the wrapper to diverse domains from business process management, to security levels for telecommunication management, to travel agency, procedurally demonstrate the flexibility and re-usability aims of this research. The expertise of agents (management of sub-processes of a business or management of a telecommunication infrastructure or network security) is bought and sold as services to and by agents, to satisfy either individual goals (for example, buying any commodity, such as security expertise for personal purposes) or some joint goal (for example, to collectively manage, through buying and selling of services, sub-processes of a business). In all these cases, the negotiation wrapper can be "configured" to "connect" a buyer to a server of a service independently of what is being bought and sold. The details of how it is configured are deferred until later chapters, but, informally, agents are configured by specifying the issues over which they negotiate, their preferences over these issues, and the behaviours the designer wants the agents to exhibit in the course of negotiation in order to achieve these preferences. A protocol is then used to allow agents to communicate and solve (or "connect") either their individual or their joint problems.

1.4.1 Business Process Management—ADEPT

The initial scenario is the British Telecom (BT) business process of providing a quotation for designing a network to provide particular services to a customer (figure 1.2)[4]. The *overall process* receives a customer service request as its input and generates as its output a quote specifying how much it would cost to build a network to realize that service. It involves up to six agent types: the sales department agent, the customer service division agent, the legal department agent, the design division agent, the surveyor department agent, and the various agents who provide the out-sourced service of vetting customers. All negotiations are centered on a multi-attribute object, where attributes are, for instance, price, quality, duration of a service (see (Jennings *et al.* 1996a) and section 2.2.1 for more details). The process is initiated by the sales agent which negotiates with the CSD agent (mainly over time, but also over the number of invocations and the form in which the final result should be delivered) for the service of providing a customer quote. The first stages of the $Provide\_Customer\_Quote$ service involve the CSD agent capturing the customer's details and vetting the customer in terms of their credit worthiness. The latter sub-service is actually performed by one of the VC agents. Negotiation is used to determine which VC agent should be selected—the main attributes negotiated over are the price of the service, the penalty for contract violation, the desired quality of the service and the time by which the service should be performed. If the customer fails the vetting procedure, then the quote process terminates. Assuming the customer is satisfactory, the CSD agent maps their requirements against a service portfolio. If the requirements can be met by a standard off-the-shelf portfolio item, then an immediate quote can be offered based on previous examples. In the case of bespoke services, however, the process is more complex. The CSD agent negotiates with the DD agent (over time and quality) for the service of designing and costing the desired network service. In order for the DD agent to provide this service, it must negotiate with the LD agent (over time) and perhaps with the SD agent. The LD

[4] The negotiations between the agents are denoted by arrows (arrow head toward client) and the service involved in the negotiation is juxtaposed to the respective arrow.

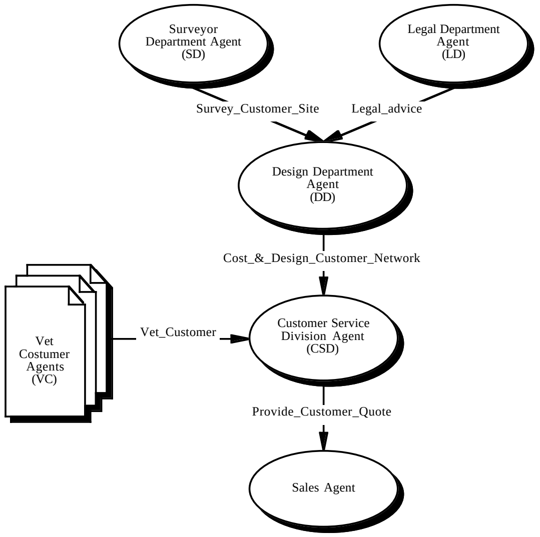

Figure 1.2: Agent system for BT's provide customer quote business process

agent checks the design to ensure the legality of the proposed service (e.g. it is illegal to send unauthorized encrypted messages across France). If the desired service is illegal, then the entire quote process terminates and the customer is informed. If the requested service is legal, then the design phase can start. To prepare a network design, it is usually necessary to have a detailed plan of the existing equipment at the customer's premises. Sometimes such plans might not exist and sometimes they may be out of date. In either case, the DD agent determines whether the customer site(s) should be surveyed. If such a survey is warranted, the DD agent negotiates with the SD agent (over price and time) for the $Survey\_Customer\_Site$ service. On completion of the network design and costing, the DD agent informs the CSD agent, which informs the customer of the service quote. The business process then terminates.

1.4.2 Telecommunication Service Management

The FIPA Agent Communication Technologies and Services (FACTS) telecommunication management problem was part of the ACTS programme of the Fourth framework of the European Community (FACTS 1998). The problem scenario is based on the use of negotiation to coordinate the dynamic provisioning of resources for a Virtual Private Network (VPN) used for meeting scheduling by end users. A VPN refers to the use of a public network (such as the Internet) in a private manner. This service is provided to the users by service and network providers. The scenario is composed of a number of agents that represent the users, the service providers and the network providers (see figure 1.3).

Individuals using the system are represented by user agents that are collectively referred to as Personal Communication Agents or PCAs. PCA agents are composed of *IPCA* and *RPCAs*; the Initiating *PCA* represents the user who wants to initiate the meeting and the Receiving *PCAs* represent the party/parties

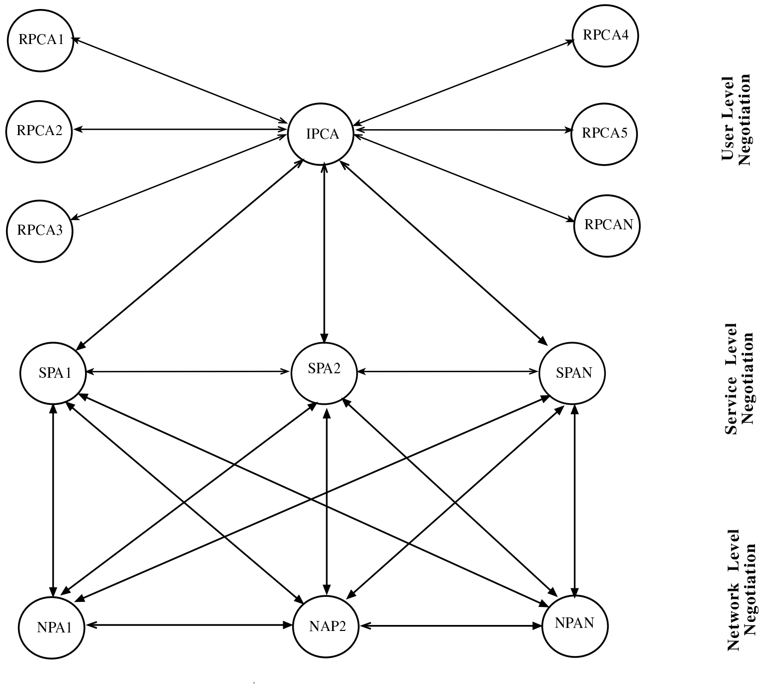

Figure 1.3: Nortel Network's FACTS Scenario

that are required to attend the meeting. Interactions between PCAs can be multilateral (involving one *IPCA* and multiple *RPCAs*) and are centered around negotiation over meeting scheduling. Each agent negotiates on behalf of their user and has the goal of establishing the most appropriate time and security level (see below) for the service requested by the *IPCA*. The set of issues over which PCAs negotiate are: $[Service\_Type, Security, Price, Start\_Time, Duration]$. $Service\_Type$ denotes the choice of the service (e.g. video, audio or mixture of both). $Price$ is the share of the price the agents should pay for the service. $Start\_Time$ and $Duration$ are the time the service will commence and its length, respectively. $Security$ encodes the privacy of the meeting and is represented by both the method of security (e.g. in the order of value to PCAs: Entrust, Verisign or Microsoft) and the level of the security method (again in the order of value: confidentiality, integrity and authentication).

The requirements of the *IPCA* and the *RPCAs* are constrained by what resources are available at the network level. For example, the network may be heavily loaded at the time the service is required by the *PCAs*. Since the network is only visible to the *IPCA* through the Service Provider Agents (*SPAs*), the threads of *IPCA* and *RPCAs* negotiation are executed in parallel with negotiations between *IPCA* and *SPAs*. Note however that the interactions between *IPCA* and *SPA* directly influence the meeting scheduling negotiations between *IPCA* and *RPCAs*. In fact, *PCAs* agents often have to make trade-off between issues given the constraints at the network level. For example, in cases of high network load the *SPA* may offer *PCAs* a later

$Start\_Time$ for a longer $Duration$. Furthermore, only bilateral negotiation is assumed between *IPCA* and *SPAs*. However, each *SPA* can make agreements with *IPCA* for services and then out-source these commitments by initiating negotiation with other *SPAs*. The set of issues in the negotiation between *IPCA* and *SPAs* is the same as that between *IPCA* and *RPCAs* except there is the additional element $Participants$ (the list of users, represented by *RPCAs*, specified to be included in the requested service).

Either concurrently, or after the service is provisioned between *IPCA* and *SPA*, multiple threads of negotiation are initiated between the *SPA* and the Network Provider Agents, *NPAs*, that manage the infrastructure and low level aspects of the IP network. These threads of interaction are multilateral since each *NPA* manages only a subset of the IP network. Therefore, the *SPA* must negotiate with a number of *NPAs* in order to secure resources for the services it provides to *IPCA*. The set of issues in the thread of negotiation between *SPA* and *NPAs* includes: $[Quality\_of\_Service, Security, Participants, Price, Start\_Time, Duration]$. Here $Quality\_of\_Service$, or QoS, represents the "goodness" of the service from an agent's perspective. QoS is often discussed as if it were composed of a number of sub issues such as, the $Bandwidth$ (capacity of the link), the $latency$ (delay imposed by the network on packets), the $jitter$ (maximum time deviation acceptable during transmission), the $availability$ (percentage of time over which the service is required) and $packetloss$ (percentage of total packets lost during lifetime of the provisioned service). Additionally, the sub issues that represent the QoS can change in the course of negotiation. For example, negotiation over QoS may begin with concerns over only the $Bandwidth$ capacity of the link, but later include $packetloss$ if the client of a service requires a higher service quality. Alternatively, sub issues may be removed from the set of issues that define QoS. For example, the *SPA* may remove $jitter$ from the set of QoS negotiation issues with *NPAs* if the end users have agreed to hold a video-conference at a geographically close location (since $jitter$ will no longer be a concern).

Additionally, these agents operate in a highly dynamic environment: services need to be updated, new ones come on line, old services are removed and currently agreed services fail. Customer's requirements may also change: new services may be required, services may be required sooner or later than initially anticipated or higher quality may become more important. In all of these cases, negotiation is the means of managing this complexity. New services become candidates of provisioning, those effected by the failed services can be re-provisioned, and service conditions can be dynamically configured or reconfigured.

1.4.3 Characteristics and Assumptions of Problem Domains

The following negotiation characteristics can be observed in the scenarios above. These characteristics form part of the requirements that need to be adequately modeled and which will be used as a benchmark for analysis of other related approaches to similar problems (chapter 3). Moreover, it is believed that these characteristics are likely to be common to a wide range of service-oriented negotiations between autonomous agents because these features are identified at a sufficiently abstract level (such as presence or absence of time limits or organizational structure) to be applicable to most complex and real-time interaction problems.

The main feature of the above scenarios relates to the design of open and closed systems, mentioned in section 1.1. A distributed system is either formed centrally by a designer, or else created dynamically through encounters. In the above two scenarios, the set of BT agents in the ADEPT system and the *SPAs* and *NPAs* in the FACTS scenario, represent a closed system. These agents have been created centrally by designer(s) according to some MAS design methodology (see (Jennings *et al.* 2000b) for the methodology for creating ADEPT agents). On the other hand, the design of, and the interactions between, the *VC* agents and the BT agents in the ADEPT system, and the *IPCAs* with the *SPAs*, in the FACTS scenario, is not a centralized process. In fact these agents can, and do, freely enter and leave interactions (for example, in a deregulated telecommunication industry where customers can choose amongst a wide range of service providers, *SPA* agents are unlikely to encounter the same *PCA* agents). As will be shown below, this open versus closed design directly influences agent interactions along a number of dimensions such as: different agent architectures, languages and reasoning procedures, varying certainty levels, autonomy, motivations and conflict types, different patterns of temporal persistancy (or the period an agent is “alive” in a negotiation), and different frequency of encounters. It is precisely for these reasons that no single coordination mechanism can be designed that solves this type of problem. Rather, the emphasis of this thesis is on a configurable negotiation framework.

In more detail, what can be said about the two domains above are:

- There are roles. Individual agents can be both clients and servers for different services in different negotiation contexts.

- Interactions can be either amongst group members (e.g. the BT agents or the PCAs) or individuals from different organizations (e.g. VC and CSD agents). The organization of agents has four closely related implications:

  – **conflict types**: The conflict between individual and system goals determines the style of interaction. Three types of conflict can be identified within the above two domains. Some negotiations involve entities within the same organization (e.g. between the *CSD* and *DD* agents) where agents share the goal of the organization. Hence, the types of interactions are generally cooperative in nature. Other negotiations are inter-organizational and purely competitive—involving self interested, utility maximizing agents (e.g. between the VC agents and the CSD agent, or between the PCA and the SPA agents). Finally, agents may share the same system goal but have different individual preferences (e.g. the scheduling of meetings by the PCAs requires resolution of different preferences even though individuals all agree that they want to meet).

  - **motivation types**: Note also that a single agent may enter different types of conflict scenarios. For example, the style of negotiation between the *CSD* agent (or *IPCA*) against *DD* (or *RPCAs*) is cooperative in nature, whereas the *CSD* (or *IPCA*) negotiations with *VC* (or *SPAs*) may be more selfish. Therefore the attitude of the agents is not fixed.
  - **autonomy**: The solution to problems, especially in inter-organizational contexts, is based on *mutual* selection of outcomes. Therefore no single agent has control over the other in terms of the selection of the final choice.
  - **uncertainty types**: Some groups of agents often negotiate with one another for the same service (e.g. the CSD and DD agents), whereas other negotiations are more open in nature (for example, the set of VC agents changes frequently and hence the CSD agent often negotiates with unknown agents).

- Negotiations can range over a number of issues (e.g. price, duration and start time). Each successful negotiation requires a range of such issues to be resolved to the satisfaction of both parties. Agents may be required to make trade-offs between issues (e.g. faster completion time for lower quality) in order to come to an agreement or dynamically change the set of issues involved in negotiation.

- As the agents are autonomous, the factors which influence their negotiation stance and behaviour are private and not available to their opponents (especially in inter-organizational and open settings). Thus, agents do not know what utilities their opponents place on various outcomes, they do not know what reasoning models they employ, they do not know their opponent's constraints and they do not know whether an agreement is even possible at the outset (i.e. the participants may have non-intersecting ranges of acceptability).

- Since negotiation takes place within a highly intertwined web of activity (the business process or a video-conference schedule), time is a critical factor. Timings are important on two distinct levels: (i) the time it takes to reach an agreement must be reasonable; and (ii) the time by which the negotiated service must be executed is important in most cases and crucial in others. The former means that the agents should not become involved in unnecessarily complex and time consuming negotiations—the time spent negotiating should be reasonable with respect to the value of the service agreement. The latter means that the agents sometimes have hard deadlines by which agreements must be in place (this occurs mainly when multiple services need to be combined or closely coordinated).

- The quantity of a particular resource has a strong and direct influence on the behaviour of agents, and, moreover, the correct appreciation of the remaining resources is an essential characteristic of good negotiators. Resources from the client's point of view relate directly to the number of servers engaged in the ongoing negotiation; likewise from the server's point of view. Thus, the quantity of resource has a similar effect on the agents' behaviour as time.

These features (or characteristics) will be used as the basis for a critical evaluation of related approaches and finally for the design of the negotiation wrapper itself.

1.5 Contributions of the Research

The work reported here is a formalization and engineering of an interaction wrapper that can be configured for use by asocial agents that need to interact with other agents in a number of different environments. It is an *engineering* endeavor because the wrapper's coordination model utilizes and integrates models from artificial intelligence and economics. Techniques from these disciplines have been used to design a strategic negotiation framework in environments characterized by *direct* and structured interactions between two agents, who have conflicting preferences over multiple issues, and where time and computation are bounded and information is uncertain. The majority of current multi-agent systems have tended to model *indirect* interactions between one to many (auctions) or many to many (markets), where the agents are simple and the institution, as the mediator, controls and, at times, specifies the strategies of interactions.

More specifically, this shift in emphasis towards *direct* and *strategic* interactions between autonomous agents has necessitated:

- employing extant communication knowledge so that agents can understand and interact with the rules of the protocol. This knowledge is modeled as an agent communication language which normatively specifies the syntax, semantics and pragmatics of possible utterances.

- developing a novel coordination architecture for strategically selecting actions given the normative rules of the protocol. The communication language above is knowledge "poor", leaving the decisions about when to use the protocol and what information to transmit to the designer. However, the currently available decision models that the designer could use to guide decision making in such situations often make unrealistic assumptions about the agent (such as perfect information or unlimited computational resources). In contrast, the developed coordination wrapper is based on the realistic assumptions that agents have limited information about their world and their reasoning capability is constrained by time and computational limitations. This relaxation of the strong assumptions has meant that the developed model only aims to compute satisfiable, rather than optimal, solutions.

  The major contributions of this thesis, implemented as a decision architecture within the wrapper, are:

  1. A more in-depth description of the environment of multi issue negotiation that agents can use for decision making. This description represents: the negotiation issues, their importance, their

reservations, the agent's preference over the issues, time deadlines and conversation threads. The presented model incorporates more negotiation concepts than previously proposed systems, thereby allowing richer reasoning mechanisms.

2. Two fully developed and novel offer generation algorithms, called *responsive* and *trade-off* mechanisms, which together search the space of possible negotiation outcomes. Another novel, but as to date undeveloped, mechanism is the *issue-set manipulation* mechanism which performs a different type of search.

   The responsive mechanism is the computationally simplest algorithm. It generates offers based on the negotiation context such as the time remaining in negotiation, the current resource usage levels in negotiation or the behaviour of the other agent. The mechanism generates offers solely on these factors and independently of the benefits that can be gained by both parties. In this sense it can be seen as a selfish mechanism.

   The trade-off and issue-set manipulation algorithms are computationally more complex and demand relatively more information about the other agent in generating offers than the responsive algorithm. The trade-off algorithm generates, unlike the responsive algorithm, offers that have the same benefit to the agent as previously, but that *may be* more beneficial to the other agent than the previous offer. This decision is uncertain because an agent does not know the evaluation function of its opponent. Fuzzy decision techniques are provided that support uncertain decision making during trade-off negotiations. Since the search for mutually more beneficial outcomes is computationally more complex than its responsive algorithm counterpart, the trade-off algorithm is considered as a more cooperative process. This is because an agent that implements such an algorithm will have to dedicate more computational resources to decision making than it would for the corresponding responsive algorithm.

   The issue-set manipulation model is also computationally more complex than the responsive mechanism (because of this increased computational complexity, this algorithm, together with the trade-off algorithm, constitute what is termed as the *deliberative* components of the wrapper). Issue-manipulation operates by dynamically changing the set of negotiation issues by adding and/or removing issues at negotiation time. The model has been developed to escape negotiation deadlocks by removing "noisy" issues that are obstructing the progress of negotiations, or by adding new issues into the negotiation that may increase the benefit to both parties. Again these evaluations are uncertain and are supported by fuzzy decision making techniques. The issue-set manipulation is the least developed component of the wrapper architecture and, unlike the responsive and trade-off algorithms, still requires the specification of an algorithm given the developed formal model.

In summary, all three mechanisms are decentralized. The responsive mechanism is novel because it formally models a concession protocol based on the environment of the agent. This allows agents to explicitly reason about how to concede in negotiation. The novelty of the trade-off mechanism is that, for the first time, it formally models this important negotiation mechanism. Furthermore, although the trade-off mechanism is computationally more complex than the responsive mechanism, it is nonetheless tractable. Finally, the issue-set manipulation mechanism formally models another type of negotiation decision mechanism that has to date not been addressed elsewhere.

3. A meta-strategy model that guides the decision making about which of the available negotiation algorithms to use. Given that there are three choices of methods to generate offers in negotiation, another level of decision making is required to make the choice about which technique to apply. This level of decision making is referred to as a *meta-strategy*. Decisions about which algorithm to use in generating an offer can be based on a number of internal or external factors to the agent, for example, the history of interactions, the time limits and so on. An important decision criteria is based on the fact that the responsive and deliberative components of the wrapper can implement both selfish or cooperative behaviours, respectively. Whereas in traditional DPS the attitude of the agent is hardwired into the protocol, moving towards open environments requires decoupling this strategic decision from the protocol. In some environments it may be beneficial to be selfish and follow the agent's own goals, whereas in other cases being cooperative is more beneficial. This novel way of coupling strategies of interactions to environments and goals via meta-strategies, rather than the protocol itself, also results in a wrapper that is more domain independent than other traditional DPS protocols.

The requirement that the wrapper is operational in both open and closed environments has resulted in a need to develop a coordination framework that is reusable. Re-usability is achieved by separating the wrapper from the domain problem solver layer of an agent through a *service* layer. The domain expert can then develop domain dependent code for the problem at hand, but use this service layer to achieve effective coordination when problems interact with other autonomous entities. Designers can then build agents without significant expertise in the development of coordination strategies.

Furthermore, the designer is provided with not only a coordination framework, but also a preliminary empirical evaluation of its components. This evaluation can be used to guide the selection of strategies in a wide range of environments. Such evaluation is needed because the wrapper is only a formal description of *possible* strategic negotiation behaviour and there is no way to predict which strategy is best for a given environment. This can only be achieved by empirically evaluating the developed coordination framework in a number of environments.

1.6 Structure of the Thesis

The remainder of the thesis is structured as follows. The requirements defined in section 1.4.3, as well as additional considerations, are given a more detailed treatment in the next chapter. These requirements are considered, and introduced, as elements that need to be modeled in negotiation which then serve as *inputs* into the wrapper layer. This chapter also elaborates on some of the assumptions made in the main body of this work. Economically and computationally motivated coordination models are then introduced in chapter three and critically evaluated for their appropriateness for the problems and requirements mentioned above. Next, in chapter four, the developed negotiation model is presented, followed by an empirical evaluation of its behaviour in a number of different environments in the penultimate chapter. Finally, chapter six presents the conclusions reached and outlines some potential future directions of this research.

Chapter 2

Components of a Negotiation Wrapper

The aim of this chapter is to define the scope of the research and justify the working assumptions. The scope of the research is presented through a description of the elements of interaction that need to be captured in the negotiation wrapper, as well as the dependency relationship(s) between these elements. This analysis and specification is in part grounded in the two application domains described in the previous chapter, and in part from the re-usability and flexibility requirements. The activities involved in the design of a wrapper are divided into: i) the identification of the important elements of negotiation that need to be captured, followed by ii) the formal or informal modeling of the identified issues. This chapter expands on the first phase of the design process. The following chapter (chapter 3) is a review of attempts to model them.

The choice of which negotiation factor(s) to model and which to omit has a direct impact on the applicability of the wrapper, in terms of not only the adequacy of the computed solution, but also the computability of the solution itself. In real world interactions, there are a large number of factors that directly influence the process and outcome of negotiation, including:

- the symmetry of agents in information and resources. Agents are in a symmetric context when they *both* have the same information and resources (Gibbons 1992). When this symmetry is broken, the relationship between the agents is often qualitatively transformed (Raiffa 1982)—the agent that has more information and/or resources can exert a larger influence on the direction the final outcome will take; the agent is said to have more "power" (Corfman & Gupta 1993).

- whether there are hard or soft deadlines. As was discussed in the previous chapter, time deadlines are important in a negotiated settlement. Hard deadlines represent absolute and inflexible time schedules by which some activity must be completed by. On the other hand, the achievement of an activity within a soft deadline is less absolute and more flexible.

- the protocol of interaction. The protocol of the interaction defines the language and rules of interaction between the agents. Negotiation protocols will be expanded on in more depth in this chapter.

- the strategies of interactions. A strategy is informally defined as an individually (or locally) chosen action of an agent given the rules of group (or global) behaviour. It is strategic because the agent can have a number of choices of the actions that will result in the achievement of a goal. This multiple choices of actions leads to agents having preferences and behaving strategically over which action to take.

- the rationality of the agents. The term rationality is informally defined as making appropriate decisions, or "doing the right thing" (Russell & Wefald 1991). The rationality of an agent is defined with respect to the type of agent that is being designed. For example, rationality of a cognitive agent is defined in terms of what actions are legitimate given the agent's current beliefs, desires and intentions (the so called BDI architecture (Bratman 1990, Cohen & Levesque 1990, Rao & Georgeff 1991). The rationality of an economic agent, on the other hand, is defined in terms of maximization of the agent's preferences, modeled as a utility function, over states of the world (Kreps 1990, Gibbons 1992, Binmore 1992). Agents in this thesis are economic and thereby abide by the latter principle of rationality.

- the possibility of coalitions. Coalition refers to interactions between different *groups* of agents (Sandholm 1999), as opposed to "monolithic" agents that only represent themselves and not others (Raiffa 1982).

- the risks and uncertainty. Uncertainty arises because agents seldom have full access to the entire information about their world. This lack of information can be due to either "laziness" (too much to be known in the world), declarative ignorance (limited knowledge of the domain—for example, chemical science has no complete theory of the science), or procedural ignorance (consequences of effects of actions are unknown) (Russell & Norvig 1995). Risks, in turn, characterize the attitude of the decision maker in choices (or what is called lotteries (Neumann & Morgernstern 1944)) between a sure outcome and an expected (or uncertain) outcome (Neumann & Morgernstern 1944).

The benefit of formalizing *all* the issues involved in negotiation is that the behaviour of the system is likely to be more predictable. However, the object of consideration of this research is only a subset of the aforementioned issues. This is because the benefit gained from formalizing all of the issues is offset by the computational difficulties they incur on coordination (for example, the information required or the amount of time it takes to reach a solution). Therefore, the first stage of the design of the negotiation wrapper (which issues to model) has been constrained by the inclusion and consideration of only the most important negotiation issues. In the main, these have been derived from the *general*, as opposed to problem/domain specific, properties of the two scenarios described previously and the configurability requirement of the wrapper for use in different types of domains. These issues can be roughly categorized into cognitive (or informational), affective (or choice) and conative (or action) (Kiss 1992). The chapter can also be viewed as a description of the following coordination components of figure 1.1:[1]

- the set of possible inputs (motivations, section 2.1.2, issues, section 2.2.1, information, section 2.2.6),
- the set of possible outputs (action and strategies, section 2.1.3, contracts, section 2.2.5)
- the set of possible environments (the agent society, section 2.1.1, protocols, section 2.1.3, time deadlines, section 2.2.7, bounded rationality, section 2.2.8, commitments, section 2.2.5)

To define the scope of this research and justify the working assumptions, the exposition is structured along two dimensions; the characteristics of the society of agents (section 2.1), and its interactions (section 2.2). The former is a description of the issues involved in modeling interactions from a multi-agent perspective, and the latter is the set of issues involved in modeling interaction from an agent-centric perspective.

2.1 Characteristics of the Society

Kraus presents a classification of coordination methods for multi agent systems that is based on i) the size of the society, ii) the motivations of the agents and iii) the presence or absence of a protocol of interaction (Kraus 1997b). These criteria, and additionally the frequency of interactions, are used below to define the scope of the research and the underlying assumptions about the agent society. The frequency of interactions is an important criteria that helps to distinguish a closed from an open system, and, as will be shown below, directly influences other factors in interactions like learning, reputations and trust.

2.1.1 Society Size

One of the aims of this research is to develop a negotiation technology for direct interactions amongst *two* agents (bi-lateral negotiation), as opposed to large scale societies requiring coordination mechanisms such as organizations, markets, auctions, voting or social decision schemes (see (Corfman & Gupta 1993) for an overview of the different decision mechanisms from bargaining, social welfare, organization, marketing and psychological disciplines). Bargaining models, defined and explained in depth in the next chapter, are in the main designed for bi-lateral negotiations (Gibbons 1992) (Nash is an exception (Nash 1950)). These models describe interactions between economically rational agents that attempt to maximize some utility. Market and auction mechanisms also model economically rational agents, but are only adequate for large number of agents (Sandholm 1999). Social decision schemes (e.g. plurality, majority, proportionality), are also inappropriate for bi-lateral negotiations because they need to form a decision based on agreements

[1] Note that nothing will be mentioned about the middleware component of figure 1.1. Issues involved at this level of coordination include synchronicity of the messages and control protocols (Parunak 1999, DAIS 1984, Mowbray & Zahavi 1995, OMG 1996) which themselves are technologies that *facilitate* coordination.

of more than two agents (Laughlin 1980, Laughlin & Earley 1982). Coordination techniques for large groups must also model the possibility of coalition (Kahan & Rapoport 1984, Shehory & Kraus 1995, Sandholm & Lesser 1997) and differential power (Binmore, Shaked, & Sutton 1984) amongst members.

Since the focus of this work is interaction between few agents, bargaining models are the most appropriate candidate mechanism (or at least, as will be shown, its solution criteria, protocols and quantitative modeling tools) for building the coordination model component of the negotiation wrapper in figure 1.1. As will be shown below, the preferences of individuals and the rules of interactions are central in these models. Although the work reported here is exclusively on the design, engineering and evaluation of the framework for bi-lateral negotiation, the framework has nonetheless been designed so that its extension to multi-lateral negotiations should not be problematic. This is achieved via modular design of the negotiation mechanisms that generate offers for bi-lateral negotiations. Multi-lateral negotiation is then achievable through concurrent reasoning over multiple independent *threads* (defined in section 4.2.3) of bi-lateral negotiations. Thus, the stance taken in this work is that bi-lateral negotiation is an appropriate first case assumption, which is extendible to multi-lateral negotiations. In fact, as will be shown in the next chapter, bi-lateral negotiation is a harder problem to solve than multi-lateral negotiation whose solution can be found in the form of auction or market mechanisms.

2.1.2 Society Motivations

Agents act in order to achieve some goal(s). The agent is then said to be motivated to achieve a given set of goals (Russell & Norvig 1995). Individual motivations of agents to achieve their own goals (or local goals) directly influences the nature and outcome of negotiations when local goals of agents interact. The importance of an agent's motivation is best illustrated by an abstract game called the *Prisoner's Dilemma* (figure 2.1).[2] There are two players in this game and each has a choice of defecting or cooperating. Each player receives a payoff, or utility, that determines how good, in some subjective way, the outcome is for the player. This payoff is often taken to mean the degree of satisfaction of the agent's preferences, modeled as a utility function. The combination of individual payoffs then defines the group welfare (also called social or global welfare), according to some combination function. The respective payoffs for each player are shown

	cooperate	*defect*
cooperate	3,3	0,5
defect	5,0	1,1

Figure 2.1: The Prisoner's Dilemma Game

[2] The game is actually a demonstration of the principle of *trust* (Raiffa 1982), and has been applied to a large class of problems in political sciences, biology, computer science, psychology and philosophy. See (Axelrod 1984) for a full description.

as row and column entries. If the agents are cooperative and cared only for the equity of the group then they should both choose to cooperate, since the sum of the individual payoffs (the group welfare) is greatest when they both cooperate ($3 + 3$). However, individually the only rational move is for an agent to defect, resulting in higher *individual* payoffs (5), but a lower group welfare ($5 + 0$ or $0 + 5$). Hence the dilemma.

Thus, motivation is an important element of agent design that strongly affects the outcomes of negotiation. This point was discussed in the previous chapter in the description of the differences between DPS and MAS. This distinction is also acknowledged in the social sciences, where an agent's attitude is a function of whether it belongs to an organization or not. Agents in an organization exist to perform a function that is externally formed and controlled. Agents not belonging to any organizations (primary, as opposed to, institutional agents (Faris 1953)), on the other hand, are more self motivated and are not centrally controlled. Thus a different organizational status in turn motivates the attitude of an agent towards interactions. Members of an organization are more likely to be concerned about the benefit of the group choice than their own preferences. Conversely, an agent participating in negotiation and not belonging to an organization is more likely to place greater emphasis on its own preferences.

As will be shown in more detail in the next chapter, there are two choices of bargaining models that individually model different types of agent motivations. The decisions and processes involved in negotiation when an agent's preferences are important (i.e. self motivated) are better modeled by non-cooperative bargaining models. On the other hand, agents that care about equity (or welfare) of the others are better modeled using cooperative bargaining models.

2.1.3 Protocols: Normative Rules and Languages

Computational agents require ordered and structured interactions (Bond & Gasser 1988). Such structuring is needed because in the absence of any normative rules of public behaviour, interactions lead to chaotic dynamics where agents can send messages that cannot be understood or the message is inappropriate given the history of the current interaction. The term "normative" states prescriptive rules of behaviour (Rosenschein & Zlotkin 1994) (what ought to be), as opposed to descriptive observation of behaviour (what is). Throughout this work, the term "protocol" refers to these high level normative rules of public behaviour. The protocol of interaction (also referred to as the "resolution protocol") must specify three aspects of public behaviour:

- the permissible content of interactions; the objects agents exchange with one another.
- the permissible process of interactions; when and how to exchange the above objects of exchange.
- the language of interaction; the language to use in exchanges.

The choice of a protocol directly influences the uncertainties involved in negotiation (section 2.2.6) and the quality of the outcome. Quality of an outcome is defined in more depth in section 2.2.3, but generally it refers to the degree of satisfaction of either or both agents' aspiration levels. Also shown, in later sections, is the relationship of how quality of an outcome is directly effected by the *content* of negotiation when more than one goal needs to be resolved. In particular, different resolution protocols can be used to differentially specify rules of interactions to reach settlements. For example, in multi-issue negotiations (also called *integrative* negotiations (Raiffa 1982)) the protocol must specify whether agents can generate offers over "packages" of issues, or alternatively over sub-packages, or reach a settlement on the most important issue first, then try and resolve other issues one by one (Raiffa 1982). These different possibilities, each implemented by a different protocol, have a direct influence on the outcome quality. For example, consider bi-lateral negotiation over two issues. In an issue by issue resolution protocol, depending on the strategies of both agents (see section 2.2.4 for a definition of strategies), one agent may gain very little in negotiation on both issues. However, in a package resolution protocol, a loss on the first issue and a simultaneous gain on the second may result in a better outcome for that agent.

Furthermore, there is a need to constrain the *process* of negotiation, otherwise agents may fail to synchronize their utterances, dispatching and receiving utterances randomly. For example, rules must specify who must begin the negotiation round (as will be shown in section 2.2.5 who starts first again directly influences the outcome of negotiation), whether negotiation is a turn taking, sequential alternating round of offers and counter-offers, or whether the resolution mechanism is a mediated one-shot simultaneous offer whose mid point of intersection is chosen by a third party as the final settlement, or a one-shot take it or leave it (divide the pie or ultimatum game (Gibbons 1992)) from one agent to another. The quality of the solution, itself possibly a function of the costs to reach the solution and the number of rounds in negotiation, and the benefits gained either individually or collectively, is directly dependent on the chosen protocol of interaction. For example, if the quality of a solution is a function of the number of messages exchanged between agents, then clearly a single-shot protocol is more "efficient" than the sequential iterated protocol. As will be shown in the next chapter, the majority of game theory models attempt to achieve speed of resolution by constraining agents' choices of strategies through the design of negotiation protocols that, although they can be iterative, are, nonetheless, single-shot (or instant) when agents act rationally.

In addition to normatively specifying the permissible *content* and *process* of interactions, a protocol must specify the *language* of the interaction. The language of interaction is a model of the syntax and semantics of utterances that agents can make during their interactions (Finin & Fritzson 1994). The syntax of the agent communication language functions to distinguish messages based on grammatical forms, and adherence to this syntax assists comprehension of messages. The semantics of utterances, on the other hand, distinguishes messages based on their intrinsic meaning, which can be informing, querying, requesting or ordering (Cohen & Perrault 1979, Werner 1989). Furthermore, there is a need to map the terms and concepts of the individual agents into a shared representation (or a common ontology (Gruber 1994, Huhns & Singh

1997, Guha & Lenat 1990)) for successful coordination and communication.

All of these design choices can be dictated by the designer(s) for a closed system. However, in more open environments there is possibly a need for a pre-negotiation phase where agents come to mutual agreements over not only the rules and language of interactions, but also, as will be shown below in section 2.2.2, the set of issues that need to be resolved (or the content of negotiation) and their resolution protocol.

2.1.4 Frequency of Interactions

When agents interact with one another, they do so either anonymously (as drivers on a highway) or their identity must be known (as dealers in a stock exchange). The issue of identity is particularly important if interactions are repeated. The possibility of repetition of encounters directly influences the type of models that can be used for modeling interactions. A model may be sensitive to whether agents meet again or not. For example, in repeated interactions, models are needed that can capture an agent's ability to learn others' strategies and/or their preferences (Zeng & Sycara 1997, Bui, Kieronska, & Venkatesh 1996). Furthermore, in repeated interactions, reputations become important. Kreps and Wilson have shown that early in the interaction history, agents attempt to acquire a reputation of being "tough" or "benevolent" (Kreps & Wilson 1982). They demonstrated this "reputation effect" where agents take actions that appear individually costly but yield a reputation that is beneficial later. Milgrom and Robert identified information uncertainty and repeated actions with the possibility of observing past behaviours as the two factors that lead to the emergence of a reputation (Milgrom & Roberts 1982).

Although repeated interaction is a realistic possibility, especially in closed systems, the negotiation model developed in this work is for single encounter interactions. This is for two main reasons. Firstly, the number of issues involved in the modeling of interactions is already large. Therefore, as a first case assumption, a model of negotiation is sought that adequately describes and predicts the core elements of negotiation. Once achieved, this assumption can then be relaxed and the developed model updated (possibly with multi-agent learning algorithms, to account for repeated interactions). Secondly, although possible, interactions in open systems are unlikely to be repeated. Agents have an incentive to enter and leave different electronic communities with evolving degree of services provided by each community.[3] For these two reasons, the simplifying assumption that interactions are isolated is made.

2.2 Characteristics of Interactions

Having defined the characteristics of the agent society, this section presents the set of issues involved in modeling interaction from an agent-centric perspective.

[3] For example, although convenient, it is not necessary for an individual to buy weekly groceries from the same store all the time. Better offers, the possibility of Internet shopping and other factors may give sufficient incentive to the consumer to break the routine of going to the same store and buy products from varied vendors.

2.2.1 Object of Negotiation—Issues

The design of the wrapper must firstly include *what* agents exchange with one another in the course of negotiations—that is, the content, or object, of communication. In classical DPS, negotiation objects may be plans, goals or information. In other explicit coordination models, these objects may be other high level constructs such as intentions, arguments or, justifications (Kraus, Nirkhe, & Sycara 1998, Parsons, Sierra, & Jennings 1998). However, since agents in this work are viewed as buyers and sellers of services, the objects of negotiation are offers and counter offers over a set of service issues. Issues represent various dimensions of a service production or consumption. Services are represented as multi-dimensional goods, since complex services in the real world are seldom adequately described in terms of a single feature. Pricing is one method of describing goods using a single issue. However, although useful for describing commodities, a decision maker is presented with a random choice in the face of two or more equally priced services. Other dimensions of a good must be provided to the decision maker in order to differentiate the goods and better support allocation decisions of the good (see section 3.2.9 for arguments against pricing). For example, a banking service is not just defined in terms of the interest rates it offers, but also its loan schemes and/or repayment methods. Likewise, access to a shared resource, such as parallel computers, may be described in terms of features such as job waiting length, speed or memory limits. Issues therefore describe features of a domain, which may be qualitative in nature (e.g. repayment schemes) or quantitative (e.g. waiting length of the que) with discrete or continuous domain values respectively.

Generally speaking, issues are rarely viewed as equally *important*. For example, a banking service provider may deem the interest rate more important than the repayment scheme or memory usage may be more important than CPU usage on a parallel computer. Issues also have *reservation* values associated with them. For example, there is a maximum amount of memory a user may be permitted to utilize on a parallel computer. Conversely, there is a minimum interest rate that the institution will not consider economically viable for a lending policy. Generally, for autonomous agents, these reservation values can be viewed as constraints associated with the issues that typically represent the limitations placed on:[4] the resources needed to produce a service, together with their usage schedule (e.g. *quality*, *number* or *volume*, *delivery time*); the information required for executing a service and the information produced as the output at the end of the service execution; the penalty for decommiting from an agreed contract; or the price of a service. Issues, importance levels and reservation values are highly domain typed (domain specific in nature), reflecting dimensions of the problem at the level of the domain problem solving. Therefore, these factors are viewed as inputs (originating from the domain problem solver) into the coordination model.

Once formulated, these issues, their relative importance and their satisfaction constraints must be represented to the negotiation wrapper by the domain problem solver. The task of the wrapper is then informally

[4] This is a *possible* set of constraints because issues may vary in different domains.

defined as the goal to achieve the satisfaction of the issues, given their constraints, or the maximization of some satisfaction function when interacting with other agents for service provisioning. More formally, the decision problem P of the negotiation wrapper is described by the tuple:

$$P = \langle I, C, Criteria \rangle \tag{2.1}$$

where I is the set of negotiation issues, C is their associated constraints and $Criteria$ is a set of cost/benefit functions for each issue that the wrapper must minimize/maximize respectively. Negotiation, then, is viewed as a process of settling disputes over each of the issues in the set I when the satisfaction of an agent's goal interact negatively with the satisfaction of the others' goals. As mentioned earlier, goals interact because the fulfillment of one goal has a negative effect on the fulfilment of another agent's goal, due to exclusive goal state desired by two or more agents (e.g. a buyer wants to buy a service at a low price and a seller wants to sell at a high price).

2.2.2 Issue Set Identification and Modification

The above discussion assumed that agents *shared* the same goal set I, and that conflict resolution arises due to a conflict of preferences over goals. However, before goal satisfaction can commence, agents have to identify which goals are actually in conflict:

> ...these (coordination techniques) presuppose that the agents already know what they are "arguing" about, and what remains to be done is to settle the "argument". It is my contention that, in many domains, a substantial part of the negotiation effort is involved in figuring out what needs to be settled. As our computational agents are increasingly applied in dynamically evolving worlds (like on the Internet), capabilities for identifying who needs to negotiate and over what, rather than having these predefined by the system developers or users, will come to the fore (Durfee 1998).

Therefore, in addition to resolving conflicting goals (section 2.1.3), the resolution protocol must generate a unified and mutually agreed upon set of issues for the agents to negotiate over in the first place. This requirement can be captured by a protocol that includes a pre-negotiation phase, where agents enumerate, discuss and select which of their goals are in conflict and need to be resolved. Furthermore, since in an open system the space of possible concerns can evolve continuously, the negotiation protocol must specify whether this mutually agreed upon set of issues is static or can be added to or deleted throughout the negotiation phase. For example, the inclusion of issues into the negotiation set is often permitted and functions as a "side-payment" altering the dynamics of the negotiation (Binmore & Dasgupta 1989). Likewise, "noisy" issues may be removed either because they jeopardize successful negotiations, thus helping escape local minima in the negotiation dynamics, or because "*negotiating over the root causes of numerous disagree-*

ments can sometimes be more cost-effective than negotiating over each individual disagreement separately" (Durfee 1998).

2.2.3 Solution Quality

The quality of an outcome measures how good the outcome is from the perspective of either the individual or the society (individual and joint welfare respectively). Consideration of the quality of the wrapper's output (a contract to the domain problem solver) must be considered in the wrapper design process for two reasons.

Firstly, as was discussed in section 2.1.2, the motivations of the domain problem solver can be either self or group interested (selfish and benevolent respectively), corresponding to increasing the individual or the groups' quality of the final outcome respectively. This motivational stance can then be used by the wrapper as a decision criteria about how to behave in negotiation. For example, in the context of a minimum task load and plentiful computational resources, the domain problem solver may prefer solutions from the wrapper that increases the satisfaction of all parties involved in negotiation (the problem solver is motivated by joint welfare). Alternatively, under time pressures or where there is a large task load, a domain problem solver may be satisfied with a lower individual solution quality (the problem solver is motivated by task completion). Therefore, a notion of solution quality is needed that objectively measures the outcome of negotiations from both a local individual perspective and a global social perspective. As will be shown later, the quality of a solution is closely linked to the boundedness of an agent (see section 2.2.8).

Another justification for having a measure of solution quality, independently of the motivations of the domain problem solver, is that the joint welfare can be increased directly as a consequence of describing services in a multi-dimensional manner. Quantitative models (see chapter three) often distinguish between zero-sum and non-zero sum games (Gibbons 1992) (or distributive and integrative negotiations respectively (Raiffa 1982)). Zero-sum games are defined as games where the addition of the individual payoffs for an outcome sum to zero. More formally. Let I be the set of n agents. Let S be the space of joint strategies, $S = S_1, \ldots, S_n$ (for example, *defect,defect* strategies in the Prisoner's dilemma described in section 2.1.2), each agent choosing from a finite set of individual strategies $S_i = \sigma_{i1}, \ldots, \sigma_{im}$ (again, the strategy choices are *defect, cooperate* in the Prisoner's dilemma). Let P be a set of payoff functions P_i for each player i, each of which is of the form $P_i : S \to I\!R$ (the prison sentences issued in the Prisoner's dilemma described section 2.1.2). Then a zero-sum game is defined as:

$$\forall \sigma \in S. \sum_{i=1}^{n} P_i(\sigma) = 0$$

where the payoffs always sum to zero. Poker is a classic example of a zero-sum game because whatever money is won by one agent is necessarily lost by the others. It follows that in a two player zero-sum game the interests of the agents are in conflict and self interested agents will attempt to maximize their minimum payoff (maximin criteria of rationality—a player takes an action and the opponent reacts with its best action, which due to the nature of the zero-sum game, results in the minimum outcome for the player (Binmore 1992)).

There are also constant sum games where the agents' payoffs always sum to a fixed constant c (Binmore 1992). It can be shown that any constant-sum game can be changed into an equivalent zero-sum game by simply subtracting the constant c from all of one of the player's payoffs (Binmore 1992).

In non-constant sum games (or integrative bargaining), on the other hand, the interests of the agents are not totally antagonistic. A non-constant sum game is defined as:

$$\exists \sigma, \sigma^{'} \in S. \sum_{i=1}^{n} P_i(\sigma) \neq \sum_{i=1}^{n} P_i(\sigma^{'})$$

where at least one strategy combination is better from the view point of the group. This allows agents to search for mutually more satisfactory outcomes (called "win-win" bargaining (Raiffa 1982)). In integrative negotiation involving a number of issues it is no longer true that if one party gets more the other necessarily has to get less; they both can get more (Raiffa 1982).

Therefore, some objective measure(s) of the quality of outcomes can serve as a benchmark in (empirically) analyzing the performance of the developed negotiation reasoning mechanism(s), given that theoretically multi-issue negotiations should result in better global outcomes than purely conflicting single issue negotiations.

2.2.4 Decisions, Actions, Strategies and Rationality

Given the desired goal state, the wrapper's coordination module is then faced with the task of how to transform the current world state to the goal state, in such a way as to not only satisfy, either fully or partially, its own goal(s), but perhaps also the goal(s) of others involved in the interactions. This task can either be viewed as *problem solving* or *decision making* (Laughlin 1980). This distinction expresses a division between coordination tasks that involve the construction of resolution alternatives that are demonstrably correct and tasks that involve decision making when no objectively correct answer exists and the resolution process emphasizes the selection of alternatives based on an agent's preferences. Problem solving coordination tasks are better modeled using an argumentation based mechanism (Walton & Krabbe 1995), requiring explicit communication of high level objects like justifications, arguments and beliefs (see section 2.2.1), where arguments and justifications serve to modify others' beliefs (recall the taxonomy of different types of coordination techniques, such as persuasion, argumentation and negotiation, based on their differential rationale, methodology and effects). Decision making coordination tasks, on the other hand, are better modeled by a negotiation mechanism, where the objects of communication are preferences/demands over goals.

The task of the negotiation wrapper in this body of work is decision making since no objectively correct answer exists, and the object of coordination is an agent's goals and its preferences over these goals. As will be shown in the next chapter, this decision problem has a solution in bargaining models of game theory, where the problem reduces to representing preference relationships quantitatively as utilities, that *satisfy* (rather than *cause*) the preferences (Neumann & Morgernstern 1944). Rational behaviour then consists of acting as though to maximize this utility function.

Furthermore, due to the privacy of information and the uncertainties involved in negotiation (see section 2.2.6), the conflict resolution protocol is likely to be iterative, involving more than one round of negotiations. If agents had perfect information and unlimited computational capabilities, then a resolution could be arrived at immediately (Kraus 1997a). However, resolution may not be immediate in uncertain and computationally bounded environments (Kraus 1997a). Thus agents are faced with a problem of constructing a *sequence* of actions (called a *strategy*). The notion of a strategy is closely tied to the protocol of interaction, where strategies are taken to mean the individual, private, and centrally uncontrolled, usage of permissible actions available given the protocol rules of interaction. The decision problem is further complicated by strict constraints on the decision mechanisms such as computational or informational limitations. This latter point is described in more detail in section 2.2.8. In this sub-section the concept of actions and strategies are described in more detail.

The task of a coordination wrapper is the formulation of individual actions for the agent throughout the negotiation *and* the specification of how to combine these individual actions in the course of negotiation into a coherent strategy that achieves the goal of resolving the conflict, while respecting i) the normative rules of the protocol and ii) the bounded nature of the domain problem solver.

In negotiation, actions can be roughly divided into evaluatory and offer generation decision categories. Specifically, during negotiation the coordination module of the wrapper must make the following decisions:

1. what is the range of acceptable agreements?
2. what initial offers should be sent out?
3. what counter offers should be generated?
4. when should negotiation be abandoned?
5. when is an agreement reached?

The first point represents the set of possible outcomes, determining *individually* acceptable (or individually rational) settlements of the conflict over the issues. Note that these settlement regions are closely linked to the notion of partial and complete fulfillment of goals, represented as utility values. This range of possible agreements is formally represented in section 3.1.4. An important assumption in this work is that this set of acceptable agreements is independent of the existence of outside options, a central assumption of non-cooperative game theory also (see chapter three). An agent is said to have an outside option if in the course of negotiation with one agent it has already established, possibly a tentative, agreement with another agent. The process and outcome of negotiation is directly influenced when agents have outside options, giving greater power to those with more valuable outside options because they can legitimately threaten to leave negotiations (Corfman & Gupta 1993). However, rather than modeling the influence of an agent over decisions (its power), throughout this work the range of acceptable agreements is bounded to zero utility at the minimum (the *conflict outcome* (Zlotkin & Rosenchein 1992)). Thus, all negotiation decisions are made with respect to a failure reference point (no fulfilment) specified by this conflict outcome that determines agents' payoffs in cases of failure to reach a resolution.

Given the range of acceptable agreements and the information history of interaction, the chain of decisions between points two to five above then constitutes an agent's strategy. The set of resolution strategies available can be classified into the following strategies:[5]

- *log-rolling*: where each agent slightly relaxes its constraints (Pruitt 1981). This strategy is also often referred to as a *concession* strategy (Pruitt 1981).

- *bridging*: involving the development of a completely new solution that satisfies only the most important constraints (Pruitt 1981).

- *unlinking*: involving overlooking weak interactions among constraints (Pruitt 1981).

- *pursuing goals independently*: where each agent pursues its goal(s) without taking into consideration the goal(s) of others (Sycara 1987).

- *anti-planning*: where an agent forms a plan to prevent another agent from fulfilling its goal(s) or prevents others from interfering with its own plans (Schank & Abelson 1977). An agent persuading another agent to abandon its goals is an example of one anti-planning strategy.

The above is not an exhaustive list of strategies that an agent can follow throughout negotiation, but rather enumerates a set of likely courses of actions open to an agent. Further resolution strategies can be composed by *combining* individual strategies into what will be referred to as *meta-strategy*, in response to the intrinsic or extrinsic conditions of an agent. For example, due to the lack of an immediate deadline or the perceived importance of the given goal, the negotiation wrapper may select a course of action that implements an anti-planning strategy. However, in the course of negotiation the chosen strategy may lead to a deadlock and necessitate a change of strategy to a log-rolling strategy. Thus, the wrapper is required to

[5] Note that the presented strategy list is for iterative protocols. There are a whole wealth of strategies according to the type of protocol (Binmore 1992).

not only initiate a strategy, but also monitor and, if required, reassess its applicability, given that the agents' tasks and goals may change in the course of negotiation.

2.2.5 Commitments

Once a conflict has been resolved, it is desirable to ensure these resolutions are kept by all parties. Commitments function to provide this stability of resolutions. Commitments are inextricably linked to the notion of trust and different coordination mechanisms model trust differently. For example, in cooperative domains agents implicitly trust one another, since it is common knowledge that agents share a common goal and personal preferences can be overridden. Non-cooperative models of negotiation, on the other hand, implicitly model trust through a notion of *equilibrium* (see next chapter), specifying a strategy for each agent where deviation from these strategies is individually irrational. Hence, trust is self enforcing.

The problem of trust is nicely shown in the simple game shown in figure 2.2 by Raiffa. This game also demonstrates the role of commitments in more quantitative models of negotiation (Raiffa 1982). The game is an abstraction of Camp David negotiations between Israel and Egypt.

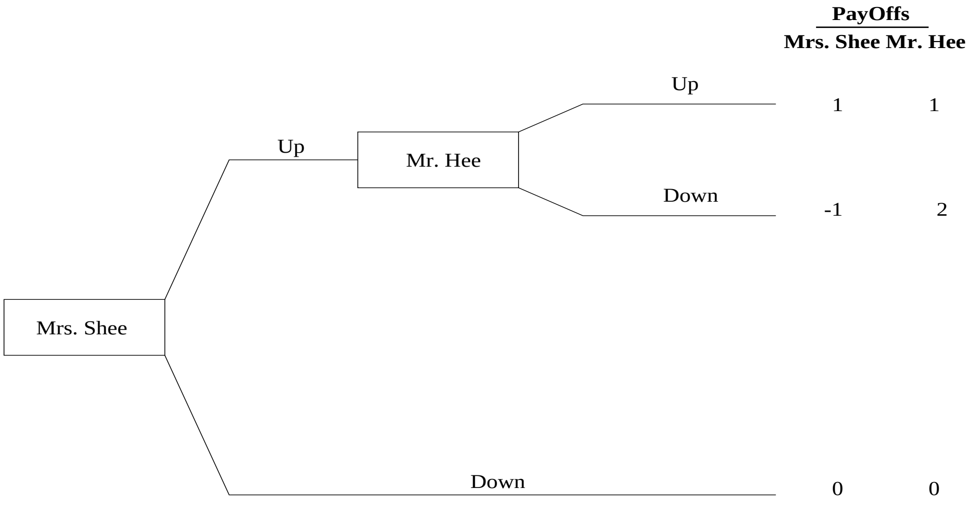

Figure 2.2: Commitment Game

There are two players *Mrs. Shee* and *Mr. Hee*, playing a game that consists of an alternating offer protocol between the two players. The permissible moves in this game are *up* or *down* and *Mrs. Shee* is given the control to move first. Then it is the turn of *Mr. Hee* to move either up or down. The respective payoffs of each player are shown on the right hand side of the figure. Suppose the game is played only once, the players are fully informed of the rules of the game and the outcome scores, and there is no communication. *Mrs. Shee* might think as follows. "If I choose *down* then we both get 0.[6] If I choose *up*,

[6] This line of reasoning assumes that agents can make inter-personal payoff comparison, an assumption that will be returned to in the next chapter.

then he will certainly choose *down*, since he would rather get 2 than 1. Hence if I choose *up* I'll get -1. I'm better off choosing *down* (maximizing the minimum loss, or maximin, strategy). It is too bad we cannot talk to each other and agree that we both should choose *up*". Now assume that the players can communicate, but that now the agreements are non-binding, or non committal. The game might then be played as follows. *Mr. Hee* might say, "it doesn't make sense for you to choose *down*. If we both choose *up* then we'll get 1". She might respond: "True. But how do I know that you won't switch to *down* later on, when I have committed to *up*?". Her problem is whether she can trust him. His intentions may indeed be to commit himself to *up*, now, but later on, due to some unforeseen event, he may be forced to choose *down* when she has chosen *up*.[7] After explaining her fears of his switch she then proposes to *Mr. Hee* that "I'm going to choose the *down* alternative, unless you can take some binding action now to reduce that payoff of 2 units to a value below 1" (called *free disposal* by economists, (Binmore 1992)).

The dynamics of the game are altered if the game is repeated an infinite number of times. She would then know that if his response to her choice of *up* was *down*, then in the next stage she will choose *down*.[8] This outcome also underlines the importance of repeated interactions, described in section 2.1.4. This simple game shows the central role commitments play in joint activity. As Lesser notes:

> The ability to appropriately bound the intentions of agents and to create and sufficiently guarantee the commitments of agents to accomplish certain tasks is at the heart of efficient, organized behaviour (Lesser 1998)

Commitments, in DAI, are viewed as pledges to undertake a certain course of action (Jennings 1996). In classical distributed planning, it provides a certain degree of predictability to the agents, so that they can take the *future* courses of actions of others into account when there are interdependencies, resource conflicts or global constraints.

When proposals are fully binding, agents cannot retract a proposal once it has made it. Therefore agents need to make sure they "look" before they "leap" (Durfee 1998). However, commitments can also be temporally bounded and different coordination mechanisms are based on different time scales where the commitments may be valid. For example, organizations, a coordination mechanism, model commitments via the notion of *roles*, which are static and long term (Carley & Gasser 1999). When agents agree to play a role within an organization they commit themselves to comply with the behaviour that the role and their relationships imply (Ossowski 1999). On the other hand, in the multi-agent planning paradigm, agents

[7] This example nicely shows the role of turn taking in negotiation, since clearly the person that moves first is at a disadvantage. This is another issue which a protocol of interaction must take into account.

[8] However, the game is complicated in cases where there is a finite number of iterations and both players know this number. This can lead to backward induction reasoning resulting in playing *down*. The discussion of this point is a divergence, but details of the game can be found in the actual example by Raiffa (Raiffa 1982), p. 199.

commit to behave in accordance with the generated joint plan of future actions and interactions. However, since plans can change, due to unforeseen events occurring in a dynamic environment, successful execution of a multi-agent plan can not be a priori assumed. Instead agents must re-plan and commitments must be managed. In such contexts, commitments can be managed through a notion of *conventions* (Jennings 1993) which i) constrain the conditions under which commitments should be reassessed and ii) specify the associated actions that should be undertaken. Conversely, a negotiation mechanism for coordination can be based both on short or long term commitments, where the process dynamically generates commitments between agents. In cases of failures, commitments can be re-negotiated; thus either amending the original commitment or generating a new commitment.

Commitments, and their temporal validity, become increasingly important in cases of selfish agents. Commitments in such cases have been modeled quantitatively (from game theory) by conditioning the commitment to a contract (called *contingency contracts* (Sandholm 1999)) on the probabilistically known future events—that is, the obligations of the contract are made contingent on future events (Raiffa 1982). If this approach is adopted, then Sandholm identifies two issues that need to be addressed for modeling commitments for automated and selfish agents (Sandholm 1999). Firstly, contingency contracts may be good for a small number of events, but there may be a potentially combinatorial explosion in the number of events in real world problems that need to be conditioned on. It is often practically impossible to enumerate all possible relevant future events in advance. Secondly, the verification of the occurrence of an event among selfish agents is problematic, because events may only be observable by a single agent which may have an incentive to lie. Thus, to be viable, a contingency contract needs a mechanism to correctly detect and verify events that is not manipulable, complicated or costly.

2.2.6 Information

An essential component of any decision making is information. Information is informally defined as knowledge about all those factors, both intrinsic and extrinsic to the decision maker, which affects the ability of an individual to make choices in any given situation (Young 1975). These factors correspond to the contents of the Self and Acquaintance components of the wrapper in figure 1.1 respectively. As mentioned earlier, in most DPS systems negotiation protocols are used to inform agents of the plans and goals of other agents. Indeed, if agents held complete knowledge of the goals, actions and interactions of other agents then coordination would not be needed (removing the problem mentioned in section 1), since agents would know exactly the current and future state of other agents. However, the perfect knowledge assumption is often invalid in real world contexts. This means it is necessary to include mechanisms within the wrapper for handling sources of *uncertainty* over the plans, goals and actions of other agents during interactions. The aim of this section is to elaborate on the sources and solutions to the uncertainty problem in coordination.

2.2.6.1 Uncertainty and Incomplete Information

The availability of information involving choices among alternatives is central to an individual's choice. However, in negotiation the availability of information about the potential choices of *other agents* introduces a further degree of complexity into an individual's decision making process. The most important source of uncertainty in negotiation is the beliefs of the other agent(s), and, as will be shown below, these uncertainties directly influence the processes and outcomes of interactions. If an agent is economically rational, as modeled in this thesis, then the goal of the agent is to maximize its utility. What is uncertain is *how* (what strategy) agent(s) take to achieve their goal.

A condition for coherence of a multi-agent system and conflict avoidance is reasoning about the non-local effects of local decisions (see section 1.3). However, if the behaviour between two member of a group involving a choice of action is contingent on that individual's estimates of the actions (or choices) of others in the group, then the actions of each of the relevant others are based on a similar estimate of the behaviour of group members other than itself. This is referred to as strategic interaction (SI). As Rapoport notes:

> strategic behaviour will occur whenever two or more individuals all find that the outcome associated with their choices are partially controlled by each other (Rapoport 1964).

Most rational decision making models have often ignored the issue of uncertainty by assuming perfect information (Young 1975). The models therefore assume the environment of the decision maker is fixed or else treat it as if it were fixed. The environment of a decision maker is fixed by assuming that either the values that describe the environmental variables are fixed (e.g. sunny 365 days or a probability distribution) or by appealing to the law of large numbers (e.g. if there are a large number of individuals involved in a given activity, such as the economy, then each individual is perceived as insignificant (Young 1975)). However, whereas the concept of information is reasonably straightforward in choice situations involving a decision making environment which is fixed, or can be treated as such, the concept itself becomes ambiguous under conditions of strategic interactions and consequently negotiation, since negotiation is strategic itself.

2.2.6.2 Single Agent Information Requirements—Fixed Environment

Even if no strategic interactions occur, the rational decision models identify the following information requirements for a decision maker (Young 1975):[9]

1. a set of alternative outcomes

2. a set of preferences over outcomes

3. an attitude towards risk

[9] Much of the following exposition is classic game theory basics and the reader is referred to text books such as (Gibbons 1992) for a more in-depth exposition of the concepts.

4. a set of mechanisms for uncertainty management

The first requirement amounts to the problem of identifying the decision maker's context by specifying a range of distinct alternatives which the individual must choose from. Normally this is solved in deductive models by assuming that these outcomes are given on an a priori basis (Gibbons 1992). However, this assumption leads to two further difficulties. Firstly, in some contexts the set of alternative outcomes can be infinitely large. For example, there can be an infinite division of a cake, or a dollar, or any divisible good. This problem is addressed in more depth in section 2.2.8. Secondly, the assumption abstracts away all the problems associated with shifts (by adding or removing alternatives) in the range of alternatives, a context that is realizable if agents are permitted to alter the set of issues involved in negotiation, thereby modifying the possible set of outcomes.

The second requirement is that the decision maker must also have a complete knowledge of its own preference orderings or utility function. That is, the individual must be able to create a confidence ranking of all the alternatives in its environment in terms of its preference. Furthermore, it is assumed that if each alternative represents a certain outcome, the decision maker needs to: i) only specify its preference ordering in ordinal terms and ii) these preference ordering are transitive and consistent over time (Gibbons 1992). However, the presence of uncertainty makes it impossible to characterize decisions perfectly. Therefore, the decision maker needs information about the probabilities associated with various outcomes in order to make a rational choice. Thus the decision maker describes its environment in terms of fixed probabilities and therefore specifies its preference orderings in cardinal terms.

Finally, in cases where it is not possible to calculate the probabilities in ordinal or cardinal terms, the decision maker requires knowledge of some technique(s) for handling uncertainty. However, problems of this kind are difficult to deal with when the phenomena are intrinsically non-iterative because the decision maker cannot even attempt to calculate probabilities in terms of empirical frequencies (Young 1975). A possible solution is to assume that the individual makes *subjective* probabilities. Subjective probabilities is a distribution that characterize and agent's degree of belief (Russell & Norvig 1995). However, this abstracts away the question of how individuals obtain specific values for subjective probabilities especially with respect to events that are non-iterative. One-off encounters between agents in an open system are likely to be non-iterative, where agents meet, interact and disappear.

2.2.6.3 Dyad Information Requirements—Dynamic Environment

The problem of dealing with and managing information is extensive even when the environment is fixed. However, the introduction of strategic interactions expands the set of information requirements for a decision maker (section 2.2.6.2) to include information describing the probable choices of others. This, in turn, introduces additional problems for an agent in i) identifying others upon whose choice its own choices are contingent and ii) acquiring information about the probable behaviour of these individuals.

One solution to the latter problem is to remove strategic interactions altogether by forming confident expectations through acquiring information (Young 1975). For example, an agent may confidently expect (the derivation of which will be explained below) that the other agent will call back when their call was cut off, so there is no need to call. Then when a decision maker discovers its choices are interdependent, it should, at best, acquire sufficient information about the relevant other(s) to form accurate predictions of their choices, or, at least, form confident expectations concerning their probable behaviour. Then the decision maker's choice problem becomes a game against nature (Young 1975). Complications caused by strategic interactions would no longer exist since the choices of other(s) would no longer be contingent on its choices. Thus the agent would be able to treat its decision making environment as if it were fixed. However, this is only logically possible, since the concept of strategic interaction means, by definition, that the choices of others will depend on the choices of the decision maker. To eliminate strategic interactions, the decision maker is assumed to require to know:

1. the range of alternatives available to others
2. their preference orderings over these alternatives
3. the probability distribution affecting the other individual's choices and attributable to nature rather than the presence of strategic interactions
4. others' reaction to, and techniques for, coping with strategic interactions since they are facing the same prediction problems

Furthermore, it is assumed that the decision maker knows the identity of the others and that they are rational. However, in open digital systems an individual is fortunate if it can identify others yet alone know points one to four above (Cranor & Resnick 2000). Even if rational decision models can cope with the first three points above, the problem still remains that other individual's efforts to cope with strategic interactions will be contingent on the behaviour of other(s) whose efforts in turn depend on the first individual. This is commonly referred to as the out guessing regress problem and its occurrence makes the procedures of forming accurate predictions or confident expectations impossible (Luce & Raiffa 1957). [10]

However, decision makers are capable of making choices under conditions of strategic interactions in the real world—whenever a decision maker does make a choice he automatically eliminates or reduces the strategic aspects of interaction (Young 1975). Therefore, in designing a negotiation wrapper, one can look for models which accurately explain and predict the actual problem solving processes involved in strategic decision making since real social systems have developed solutions to the SI problem.

[10] In fact, if one decision maker is irrational, by ignoring the fact that its choices are dependent on other(s) (i.e behaves in a very stylized fashion), then there exists a *chance* that a rational individual can accurately predict the irrational individual's behaviour and hence escaping out-guessing regress (Young 1975).

There are several methods for handling strategic interactions in the real world that can be implemented by a computational protocol. One such mechanism is to make the decision of all the participants sequential rather than simultaneous (or independent and the encounter is restricted to a single move (Gibbons 1992)). Sequential interactions permit agents to evaluate their beliefs, given an observation. SI can also be eliminated or reduced by formulating subjective estimates of the probable choices of other(s). If successful, then the agent fixes its decision making environment and the SI problem is removed. However, the formulation of subjective estimates raises two other problems. Firstly, as mentioned above, in some contexts it may be inappropriate to assign probabilities to outcomes that are infinitely large, such as division of a dollar. Secondly, formulation of subjective probabilities leads to “silent out-guessing” (Young 1975). A designer of a negotiating agent may use any number of heuristics in making these estimates, but the result will be highly subjective because they will be based on guesses about the probable choices of others, whose choice will, in turn, depend on guesses about the probable choices of the first. Therefore, the process of formulating subjective estimates will involve some silent out-guessing.

Uncertainty in decision making can also be handled by attempts to manipulate the decision making environment (Young 1975). More specifically, the choices of others are made more predictable by gaining as much influence or leverage over their behaviour as possible (Pruitt 1981). Under complete control, the behaviour of others is predictable so the problem of SI disappears. An agent can gain control of others either through a pre-specified organizational structure or via various manipulation tactics (such as lying) in the information others utilize in their decision making processes (Rosenschein & Zlotkin 1994). The effectiveness of the latter tactics, however, must take into account that others may also be using such tactics in manipulating the agent’s information set. SI can also be overcome through organizational typologies that have formal structures and communication channels. Simon quotes an illustrative example which demonstrates the role of organizations in decision making: *It is not reasonable to allow the production department and the marketing department in the widget company to make independent estimates of next year’s demand for widgets if the production department is to make the widgets that the market department is to sell. In matters like this, and also matters of product design, it may be preferable that all the relevant departments operate on the* same *body of assumptions even if....the uncertainties might justify quite a range of different assumptions. In facing uncertainty, standardization and coordination, achieved through agreed-upon assumptions and specifications, may be more effective than prediction* (Simon 1996).

Therefore uncertainty is absorbed by the organizational structure through coordination. In the work reported here, the protocol of interaction is for bi-lateral negotiation, where there is no organizational structure. Furthermore, the protocol treats each agent symmetrically, meaning that no one agent has direct control over another. Therefore no one agent can control, or has more power over, the other(s), thereby influencing their decision making.

SI can also be resolved by transforming a given relationship qualitatively (Young 1975). That is, some third party can strategically intervene by imposing a settlement of the issues. Judicial and governmental enforcement mechanisms are two examples in the real world where the settlement is through the intervention of a third party who imposes its will on the participants rather than a settlement based on the activities of the individuals themselves. Under these conditions, so far as the individuals are concerned, there is no longer any SI. However, the mechanism is no longer negotiation since negotiation ordinarily refers to the settlement of the situation involving SI through the activities of the original participants themselves. Situations involving interdependent decision making can be *partially* transformed, as above, but without producing a determinate solution for the issues. Arbitration and facilitation are such mechanisms, where negotiation interacts with such transforming procedures (Cross 1969).

Alternatively, an agent engaged in SI may attempt to acquire additional information about the other agent(s). Although not directly solving the SI problem (because the choices of other(s) will still depend on choices of the agent no matter how much effort is directed towards computing probable behaviours), this procedure may help in the formulation of subjective estimates or the selection of specific strategies in the negotiation. In addition to this feed forward (prediction of the future through expectation formation to deal with uncertain future events), an agent can also use feedback to correct for unexpected or incorrectly predicted actions of other(s). Therefore, adaptive decision making can remain stable even through large fluctuations in the environment through a feedback control.

Finally, note that the choice of an uncertainty handling method, implemented by a protocol, also directly influences the solution quality (section 2.2.3) and the efficiency of the protocol. For example, a single move sequential protocol may result in lowering the quality of outcomes (a single move prevents search for “win-win” outcomes), but may be more efficient in terms of speed. Conversely, an iterated sequential protocol may result in better outcomes, but at the expense of lower efficiency. A designer of a negotiation protocol must therefore be aware of these tradeoffs between solution quality, the efficiency of the protocol and the amount of information it assumes agents have about one another in reaching agreements.

2.2.7 Time

As noted in the previous chapter (section 1.4.3) time is a significant factor in decision making.[11] Indeed, time is an important feature of all complex and distributed systems (Bond & Gasser 1988). Classic AI theories are limited in modeling such systems because they emphasized not only single agents, but also *static* and *atemporal* environments, where the only source of change was the agent, operating in a predictable and static environment (Russell & Norvig 1995). However, complex systems are characterized by interacting subcomponents, operating in real time and dynamic environments. Thus, theories are needed that not only

[11] When the United States negotiated with the North Vietnamese toward the close of the Vietnam War, the two sides met in Paris. The first move in the negotiation was taken by the Vietnamese: they leased a house for a *two year* period (Raiffa 1982).

model multi-agents, but also their operation in dynamic and *temporal* environments.

Time affects the process of negotiation in two ways. Firstly, decision processes are affected *quantitatively* by time:

> ....the passage of time has a cost in terms of both dollars and the sacrifice of utility which stems from the postponement of consumption, and it will be precisely this cost which motivates the whole bargaining process. If it did not matter when parties agreed, it would not matter whether they agreed at all (Cross 1969).

Therefore, time manipulates the preferences of the agents through their attitudes to time-dependent costs. Secondly, time also influences the *qualitative* nature of interactions, by constraining and limiting the computational and communicational resources needed for interaction. Since interdependent activities are temporally sequenced (for example the design process of BT), activities of individuals are often subject to soft or hard time limits that directly influence the rationality of an agent. Rationality, or the ability to "do the right thing" (see section 2.2.8), requires computation and communication resources. However, if time limits must be met for joint activities then conflicts must be resolved and agreements reached within these time limits. This must be achieved with limited computational and communication resources; agents do not have infinite time to reach agreements. Thus, the presence of different time limits requires both simple and communicatively less expensive coordination decision mechanisms, *and* more complex mechanisms that take more time and may be more costly in communication. As will be shown in the next chapter, the issue of time has been central to formal game theoretic models of negotiation, that specify optimal behaviour, *instantly* attainable by agents.

2.2.8 Bounded Rationality

Another source of uncertainty in decision making relates to the local complexity of computation. In chess, for example, the size of the state space of the game (moves by both players) is 35^{100} (Marsland & Schaeffer 1990). Hence, there is no time to compute the *exact* sequences of actions. Instead, one has to guess (make uncertain decisions) and act before being certain of which action to take. This trade-off between accuracy and time costs is also reflected in negotiation decisions, where agents are time bounded and mechanisms are needed that respect this constraint. The aim, therefore, is to produce *good*, rather than *optimal* solutions.

The complexity of computation is shown in the ADEPT negotiation scenario, for the *DD* agent, the client of the *survey_customer_site* service, over two issues, ($price$ and $quantity$). Associated with each issue is the reservation value of that issue, representing the constraint for an issue's value. Let these reservations be represented as the pair $[min, max]$—$[1, 20]$ and $[2, 10]$ for $price$ and $quantity$ respectively.[12] Finally, offers over the pair of issues (or contracts) are evaluated in terms of utility to the client of the

[12] Concepts such as reservation values and utility are given a formal semantics in proceeding chapters.

contract. The decision problem of an agent is then to generate a contract that maximizes the utility of the contract. The environment of this decision problem is represented as a utility state-space problem in figure

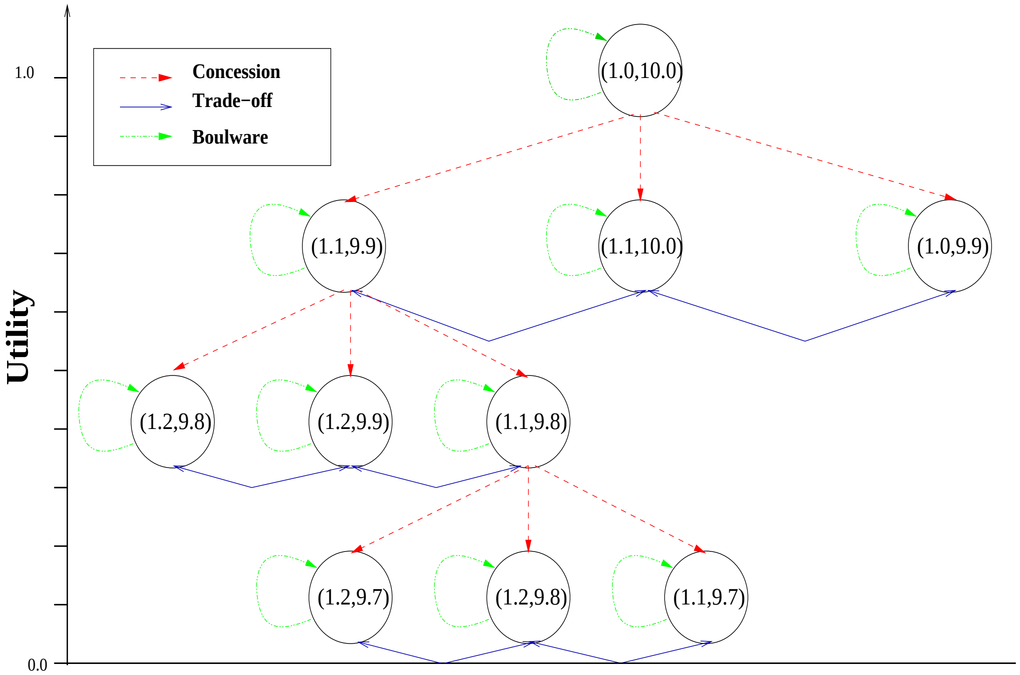

Figure 2.3: Search State Space

2.3. The *initial state may be* the contract offer $(1.0, 10.0)$, corresponding to maximal satisfaction of the agent's preferences, or utility of 1.0. This is one possible starting offer because an agent can offer any contract with different utility values according to its strategy. The final state in figure 2.3 can be any of the states that correspond to where negotiation has terminated successfully or unsuccessfully (not shown in figure 2.3 because the final state is mutually selected by the two agents).

Agents traverse the graph of the state-space using the state-space *operators* (actions). Operators can be: i) concede on utility (shown as dashed arrows in figure 2.3), ii) to demand exactly the same contract corresponding to the same utility state (called boulware and shown as the dash-dot-dot arrows), or alternatively, iii) demand the same utility but of a contract that is different to the previously offered one (shown as solid arrows in figure 2.3). A *path* is then any sequence of actions (concession or demand) leading from one state to another. The path *cost* is the cost of moving from one state to another and the *goal-test* is the evaluation to determine whether the agent is at the goal state or not. The goal state is an agreement that maximizes either the individual or the group utility according to the agent's motivations (see section 2.1.2). Given this problem (defined by the initial state, operators, paths, path-cost and goal-test), search algorithms can then be designed that select a sequence of actions that lead to a desired state.

However, a search algorithm for the above contract negotiation has to operate with two sources of uncertainty. Firstly, the client has missing information about what the server (*SD*) agent will offer. Therefore it cannot formulate a certain sequence of actions in the possible state-space. In fact, the client is unaware whether an agreement is even possible, since the information about the overlap of reservation values between the client and the server is not publicly known. In addition to this, since $price$ and $quantity$ are continuous variables, the range of possible values for each issue is infinite. This uncertainty over the overlap of the reservation values and the continuous valued nature of the issues means that the solution to the negotiation may lie at any depth. Likewise, the breadth of the state-space adds to search complexity. The branching factor (the number of sibling states from a parent state) in general is infinitely large. This combination of i) the infinite depth of the state-space, ii) the branching factor as the number of issues is scaled up from two and iii) the presence of time deadlines in negotiation leads to computational uncertainties about what is the best strategy. Game theory attempts to solve this search problem by assuming agents are rational (thus allowing pruning of segments of the search tree, such as alpha-beta pruning used in parlor games (Knuth & Moore 1975)) and supplementing this assumption with protocols that: i) constrain interactions (for example, a sequential, one round protocol can reduce the depth of the search tree to one level deep), ii) supply the agents with additional knowledge so as to better direct the search, or iii) eliminate the need for search on behalf of the agent altogether by publically supplying all the agents with the information about which strategies are optimal.

Computation, in general, functions to reach decisions that are better than no computation (such as randomness) or that result in successful outcomes. However, different computations have different costs, as well as different likelihoods of resulting in successful outcomes. Thus, in addition to developing search algorithms there is also a need for reasoning about computation (meta-reasoning (Russell & Wefald 1991)). Russell and Wefald call this meta-level rationality (or P_3)—the capacity to optimally select the combination of action and computation as opposed to perfect rationality (or P_1)—the capacity to generate successful behaviour given available information (Russell & Wefald 1991). The evaluation of which search should be implemented can then be delegated to a meta-level reasoner whose decisions can be based on factors such as the opponent's perceived strategy, the on-line cost of communication, the off-line cost of the search algorithm (or its path cost), the structure of the problem or the optimality of the search mechanism in terms of completeness (finding an agreement when one exists), the time and space (measured as memory requirements) complexity of the search mechanism, and the solution optimality of the mechanism when more than one agreement is feasible. The combination of this evaluation function and a description of the permissible mechanism state transitions can then be used by a meta-level reasoner to select amongst the available set of mechanisms.

2.3 Summary

The key issues in the design of a negotiation wrapper architecture were informally identified in this chapter. These issues relate to how the size of a society (section 2.1.1), the motivation (section 2.1.2) and the frequencies of the encounters (section 2.1.4) of the individual agents constrain the choice of models of negotiation. Also discussed was the relationship between the normative rules, the content and the language requirements of an agent communication protocol (section 2.1.3) and the computational considerations of how the choice of this protocol influences the quality of final outcome (section 2.2.3), the levels of uncertainties (section 2.2.6) and the commitments made (section 2.2.5). The nature and the role of the object, or issues, of negotiation were also outlined (section 2.2.1) as were the problems of their identification and modification (section 2.2.2). The decision making of the individual agent was then presented (section 2.2.4) and shown to be a highly uncertain activity, requiring various uncertainty management methodologies, supported by different protocols (section 2.2.6). Decision making was also shown to occur under time restrictions (section 2.2.7) and limited computational capability of the decision maker (section 2.2.8).

The adopted position in this research over these key issues is to develop a decision architecture for the negotiation wrapper that:

- supports one-off bi-lateral negotiations. Many-to-many, many-to-few and one-to-many negotiations have been successfully modeled through market, voting and auction mechanisms. Computational models of bi-lateral negotiation lag behind. As a simplifying assumption, agents are assumed to meet only once.

- supports *both* selfish and benevolent types of attitudes corresponding to maximization of individual and global welfare (or solution quality) respectively.

- supports the requirements of an iterated and sequential integrative negotiation protocol. This protocol is chosen because information is assumed to be private and negotiation over "packages" transforms fully conflicting games into partially conflicting ones, where agents can search for better joint outcomes (increased global solution quality). Furthermore, the wrapper decision architecture must support the permissible modification of the "package" during the course of negotiation.

- supports a wide range of negotiation strategies given that agents are not only under time, information and computational constraints, but they have different motivations. These strategies are introduced as *mechanisms* and function to direct the agents' negotiation decision making. One mechanism, a depth-first strategy (see figure 2.3), is formally presented as *responsive mechanism* (see chapter three), where the depth visited is a function of concession rate, which itself is a function of the resources left in negotiation, the time limits in negotiation and the behaviour of the other agents. Other more complex search strategies (not shown in the figure 2.3) implement a combination of depth-first and breadth-first strategies. This mechanism, called the *trade-off mechanism*, searches for contracts that have the same utility as a given state node, but which may lie at different depths or breadths of the utility state-space. Thus, the trade-off mechanism can explore other nodes' siblings, as opposed to the siblings of the given node alone. Finally, a mechanism, called the *issue-set manipulation mechanism*, is also provided that re-formulates the problem by changing the branching factor through the addition or retraction of issues in the negotiation. As will be shown later, each mechanism also implements a different goal-test function that evaluates whether a goal state has been reached or not.

- supports full and short term committed *contracts*. The contracts are re-negotiable. The contracts may also function as representations for other commitment honoring coordination models during the service execution life cycle (see commitment model in figure 1.1). Thus, the choice of whether to initiate re-negotiation or enact other recovery processes as directed by the commitment model, is left to the domain problem solver (possible models of which choice to make may be based on a decision theoretic cost benefit analysis of re-negotiation versus the execution of some model of commitment). The contract representation also supports both commitment failure recovery during the service execution and service provisioning phases.

Against this background, the aim of this research is to instantiate these selected issues and associated simplification assumptions into a practical negotiation framework that successfully solves the problems of the two target domains. Moreover, this framework should be configurable so that it can be evolved into other domains with a minimal amount of effort. The assumptions, methodology and solutions of the research reported here are compared next in the following chapter with game theoretic bargaining models of negotiation and selected computational models of the issues identified in this chapter.

Chapter 3

Related Work

In the previous chapter a set of important cognitive (informational), affective (choice) and conative (action) issues involved in negotiation were identified and emphasized. The second phase of the wrapper design is the modeling of these issues. To this end, this chapter critically reviews candidate models of these issues, in particular analyzing their application adequacy and assumptions, for the task of modeling the wrapper system. The content of this chapter will be concerned with models of negotiation utilized by the wrapper (coordination module and the associated information models, figure 1.1). This emphasis on the negotiation, rather than the communication, aspects of coordination is because the communication aspect of this research is not novel. Communication protocol such as the Knowledge Query and Manipulation Language (*KQML*) and the Foundation for Intelligent Physical Agents (*FIPA* (FIPA97 1997)) agent communication language (ACL) have been proposed as two solutions to the agent communication problem. *KQML* is a language and a protocol for exchanging information (Neches *et al.* 1991, Finin & Fritzson 1994, Huhns & Stephens 1999) and *FIPA* ACL is also, like *KQML*, a language that allows agents to communicate between themselves using messages (communicative acts). However, whereas the semantics of the *KQML* performatives were described informally by natural language descriptions, the *FIPA* ACL was designed to carry a clearer semantics. The communication protocol of this thesis is simply a set of primitives and associated rules for their usage.

The subject of negotiated coordination has received an in-depth treatment from a number of diverse fields, such as social welfare theory (Arrow 1950), social psychology (Pruitt 1981), economics (see section 3.1 below), marketing (Curry, Menasco, & van Ark 1991), organizational theory (Carley & Gasser 1999), operation research (Shehory & Kraus 1995), and more recently DAI (see section 3.2 below). However, for the reasons presented in the previous chapter, only decentralized models will be reviewed here.

Furthermore, since the concern of this work is negotiation for two agents, as opposed to large scale societies, coordination models such as market mechanisms,[1] voting and auctions are excluded from the

[1] Furthermore, since services in this work are unique (as opposed to being an unrestricted number of commodities) and are not

review process (see (Sandholm 1999) for a comprehensive review of these mechanisms). The class of coordination models of particular interest in this work are bargaining models which are derived from Game Theory. Game theoretic models of bargaining are discussed in section 3.1, followed by DAI extensions of these models for computational systems, in section 3.2. Finally, the overall adequacy of both approaches is discussed in section 3.3.

3.1 Game Theoretic Models of Bargaining

The central focus of economic models is the rational allocation of scarce resources through coordination mechanisms such as markets or bargaining (Binmore & Dasgupta 1989). The class of models which are of direct relevance to this research are the micro economic models of Game Theory (as opposed to macro models which model perfect competition (Gibbons 1992)) which replace the coordination mechanism of the market by individual bargaining in imperfect competition situations such as bilateral monopolies (one seller (monopoly) and one buyer (monopsony)) and oligopolies (few large suppliers (Bannock, Baxter, & Davis 1992)).

The aims (section 3.1.1) and representative key concepts of game theory (sections 3.1.2,3.1.3, 3.1.4, 3.1.5, 3.1.6 and 3.1.8) are discussed in the sections below, before a general discussion of the theory of games is presented. Due to the enormity of the discipline, only the underlying assumptions of the classic models are discussed and evaluated.[2] A concrete, and highly relevant, model is then presented in section 3.1.7 to illustrate some of the specifics of this approach. With the exception of this case study, little attempt is made to cover actual solutions for given problems since the object of the analysis is to determine the adequacy of the underlying assumptions of the models.

3.1.1 Aims of Game Theory

In game theory an agent is viewed as an individual, a firm or some more complex organization. A game is informally defined as the rules of an encounter between players, who have strategies and associated payoffs (see section 3.1.5 for a formal treatment of games). For example, the rules of driving specify drivers of the cars (the players) and a choice of actions open to the agents (to drive on the left or right hand-side). An agent then formulates its *strategy* given its beliefs or knowledge of the other agent's action. The selected strategies result in payoffs. For example, the games where both agents drive on the left or one drives on the left and the other on the right hand side of the road will result in payoffs of no crash and crash respectively. Given these rules, the object of game theory is to analyze what are the players' best choices—either both drive on the left or both on the right hand side. As will be shown formally below, the elements of a game

infinitely indivisible, the general equilibrium of market mechanisms cannot be used (Varian 1992, Kreps 1990).

[2] An explanation of standard game theory terms and concepts can be found in any of the classic text books such as the highly entertaining (Binmore 1992) or (Gibbons 1992), both of which are referenced extensively in this chapter.

are *players, actions, information, strategies, payoffs, outcomes* and *equilibria*. The players, actions and the outcomes are then collectively called the *rules of the game*. A player then selects a strategy with the available information at hand given the rules of the game. The selected strategy then results in a payoff.

The motivation of an agent (or collection of agents) is reductionist in nature. An agent is an optimizer of some function, be it genetic prosperity or maximization of profit (Binmore 1990). The aim of game theory models is to provide a general explanation of data based on a set of assumptions. Concerned by the prediction, explanation and design of economic systems, game theory models are motivated by the necessity to demonstrate that a complex system can be described and predicted without recourse to some hidden variable or *indivisible hand* (Binmore 1990).[3] Its practitioners assert that the models do not claim that this is the way the world *is* or *must* be, but rather the models describe how the world *could* be (Binmore 1990). It is this emphasis on informed design of systems (rather than heuristic approaches to modeling interactions) which has attracted recent interest in designing computational systems based on game theoretic models (Binmore & Vulkan 1997, Zlotkin & Rosenchein 1992, Rosenschein & Zlotkin 1994, Rosenchein & Genesereth 1985, Zlotkin & Rosenchein 1996, Sandholm 1996, Vulkan & Jennings 1998, 2000, Kraus & Lehmann 1995, Kraus, Wilkenfeld, & Zlotkin 1995, Shehory & Kraus 1995, Ephrati & Rosenschein 1994, Ito & Yano 1995).

The methodological stance of classic game theory is essentially testing the internal logic of the economic models through "mind experiments" using factual and counter-factual cases and simply ignoring the realizability or realism of the hypothesis; there is no need to verify or refute a theory's conclusions as long as it is logically consistent (Binmore & Dasgupta 1989).

3.1.2 Game Theory Versus Social Choice Theory

Game theory (strictly speaking, cooperative game theory, see section 3.1.3) is closely related to social choice theory (Arrow 1950), (Guillbaud 1966), (Rosenchein & Genesereth 1985), (Genesereth, Ginsberg, & Rosenchein 1986). However, game theory is concerned with:

- *the benefit of the individual rather than the group*: Social choice theory specifies how the group should behave so that its actions are consistent with some postulate of rationality. In game theory, on the other hand, the rationality principle is imposed on the individual, not the group. Thus, social choice theory seeks to determine the expected group utility function, whereas game theory seeks *first* to determine the individual benefits for each alternative, before determining the group's benefit.

[3] Adam Smith believed that individuals in a society pursued their own goals and the greatest benefit to the society came from people being free to do so. Each individual was *"led by an indivisible hand to promote an end which was no part of his intention"* (Smith 1776).

- *modeling the conflict point*: The conflict point plays a central role in game theory. It occurs where players can either break-off negotiation and receive the conflict benefit or continue to reach a deal whose benefit is relative to this conflict point. Consequently, the notion of threats becomes an important concept that needs to be modeled. A conflict outcome is not needed for a theory that is concerned with how a group should behave as a single unit. Another important consequence of the conflict point is that it (together with the assumption that agents' cardinal utilities really represent ordinal preferences, thus making it possible to transform local utilities—the so-called invariance assumption see (Nash 1950)) eliminates the need to make interpersonal comparison of benefits. Interpersonal comparison of benefits informally means that agents can reason about other's benefits—for example, "for agreement A I will receive a benefit of X and the other agent will receive the benefit Y". In social choice theory, a single group decision requires an exogenous specification of the relative weights of each individual, implying the need for interpersonal comparison among agents (Harsanyi 1967 1968). Therefore, social choice models require more information.

In this thesis the importance of the individual's rationality is, like game theory models, given primary status because agents are assumed to be selfish. However, and again similar to game theory models, decision mechanisms have been developed that also consider the group's welfare, but only when the individual's welfare for a given outcome has been determined.

3.1.3 Cooperative Versus Non-Cooperative Models

Coordination in game theory can be analyzed from two perspectives. One perspective assumes that the players of a game mistrust one another and try to maximize their own benefit irrespective of others (recall the Prisoner's Dilemma game, section 2.1.2). Conversely, the other perspective assumes that the agents make binding agreements to coordinate their strategies. These perspectives are known as non-cooperative and cooperative games respectively. In cooperative games there is a possibility of pre-play negotiations where a joint course of action is agreed on for the ensuing game. As will be shown later, this pre-negotiation communication phase eliminates the problem that occurs when multiple strategies are all the best strategy to use, referred to as multiple equilibria in cooperative games (Gibbons 1992). Nash suggested (in what has become to be referred to as the *Nash program* (Nash 1951)), that the analysis of the game should start by embedding the original pre-negotiation game within a larger game in which the possible negotiation steps appear as formal moves in the expanded game.

The most suitable coordination model on which to have the design of the negotiation wrapper is the non-cooperative model. This is for two main reasons. Firstly, there is no pre-negotiation communication in the problem domains of this research. Secondly, and more importantly, cooperative models concentrate on the outcomes of negotiation. Because of this they are unable to: i) model the negotiation *process* and ii) predict the time of agreements. Instead they concentrate on the desired properties of the outcome alternatives. However, since the agents in this research have to operate under time constraints, they need a model of the *process* of negotiation.

Despite its deficiencies, cooperative game theory is nevertheless beneficial to this research because it has produced a number of outcome criteria that formalize the quality of the outcome. These criteria can be used to evaluate the optimality of the designed search mechanisms. Optimality in these models is described in terms of *equity* (how good an outcome is in its distribution of benefits and losses to the group) and *efficiency* (if there is another group outcome that an individual member would prefer over the current one). Sandholm states that the problem of negotiation can be computationally viewed as two related optimization problems; one is how to optimize local decisions and the other is how to optimize a global criteria (Sandholm 1996). Social welfare, and game theories have both produced a number of solutions to this tradeoff problem (called the *impossibility problem* (Arrow 1950)) which can be used to evaluate the performance of the wrapper (see section 2.2.3). However, for the reasons given in section 3.1.2, welfare theory models are less appropriate than game theoretic models since the goal of this research is the design of a wrapper coordination mechanism for the *individual* agents, rather than the group.

Finally, as will be seen below, computational models of negotiation in MAS are grounded in both cooperative and non-cooperative bargaining models. Therefore, both types of bargaining models will be reviewed first to assist review of the computational models .

3.1.4 The Theory of Cooperative Games

Cooperative models are also known as *axiomatic theories*, where axioms reflect the desirable properties of solutions (Gibbons 1992). A solution in game theory is generally taken to mean agents' strategies are in equilibrium; one agent's strategy is the best response to the other's strategies, and vice versa (see section 3.1.5 for a formal definition). Then, outcomes, rather than the processes, that satisfy these axioms are sought. Non-cooperative theories are also known as *strategic bargaining theories* since in non-cooperative models the bargaining situation is modeled as a game and the outcome is based on an analysis of which of the players' strategies are in equilibrium.

The Nash bargaining solution is the most popular solution concept in cooperative models (Nash 1950). In the problems considered, there are two agents who have to negotiate an outcome $o \in O$, where O is the set of possible outcomes. If they reach an agreement, then they each receive a payoff dictated by their utility function defined as $U_i : O \rightarrow I\!\!R, i \in [1,2]$. A utility function U represents the preference relation $\succeq$ of an agent over the set of outcomes O (Binmore 1992). If they fail to reach a deal, they receive the conflict payoff, $U_i(o_{conflict})$. The set of possible outcomes and the conflict point c (payoffs $(0,0)$) is shown in figure 3.1. The space of *feasible* outcomes (call this B) is bounded by the *Pareto Optimal* line (Debreu 1959). Formally, pareto optimality is defined for a bargaining game (B, c) (the pairs formed by the set of feasible outcomes and the conflict point) as follows. Suppose there are two outcomes b and d such that they both belong to the feasible set, $b \in B, d \in B$. If $U_i(d) > U_i(b)$, for $i = [1, 2]$, then the negotiators never agree on b whenever another available outcome d is better for at least one of the agents. This is formally represented as a function that given the game defined by the pair B and c does not select b—$f(B, c) \neq b$. Note the assumption here, that agents must be able to know and be able to communicate that d is better than b. One implication of pareto optimality is that a deal should always be reached since c is not pareto optimal. Pareto optimality is a useful evaluation criteria of different negotiation outcomes because it takes a global perspective of the *efficiency* of the mechanism in terms of global good (see argument in section 3.1.3). In the remaining part of this section, two measures of *equity* of outcomes will be reviewed.

The outcome region B is bounded because the pareto optimal line represents outcomes that dominate all possible feasible outcomes (i.e. outcomes on the pareto optimal line are the best). However, agents can negotiate on an altered outcome set in a number of ways. Firstly, more solution points in area B can be represented by extending *pure* strategies to *mixed* strategies. Assume agents a and b have choices of actions, s_1, s_2 and t_1, t_2 respectively. A pure strategy is then pairings such as (s_1, t_1),(s_1, t_2),(s_2, t_1),(s_2, t_2)—a pure strategy is the action of one player given the other's action ((Neumann & Morgernstern 1944), (Binmore 1992), p. 175). A mixed strategy, on the other hand, is achieved by a lottery, where strategies are selected from a probability distribution. In the example above this means that agent a, for example, plays strategy s_1 *and* s_2 with a probability of say 0.3 and 0.7 respectively, given that b has played t_1 for example. Given that strategies can be specified with a certain probability, the set of outcomes is now expanded from the original pure strategy case. Another way of changing the set B is to allow agents to change their payoff values before the game starts (i.e. "burn some money" — *free disposal*, section 2.2.5). Alternatively, agents may be permitted to sign types of contracts that specify some transfer of utility from one agent to another after the game ("side payments"—use of pure strategies followed by transfer of 0.5 utility, for example, from agent 1 to agent 2). These three choices can help agents to expand the set of agreements which are not present in the original representation of the problem.

Given the above solution points, payoffs and strategies the key question of cooperative game theories is "what will rational agents choose"—what von Neumann and Morgenstern termed the *feasible bargaining set* (Neumann & Morgernstern 1944). A bargaining set is *individually rational* and *pareto optimal*. An agreement is individually rational if it assigns each agent a utility that is at least as large as an agent can guarantee for itself from the conflict outcome c—if $o >= c$. They argued that the outcome was indeterminate, since any point on the pareto optimal line is as good as another. That is all that can be said.

The aim of other cooperative theories, on the other hand, is to specify axioms that lead to the selection of a *single* point on the pareto optimal line, given the bargaining problem (B, c).[4] Three popular solutions

[4] The process of how to actually reach this point is of no concern to cooperative game theorists.

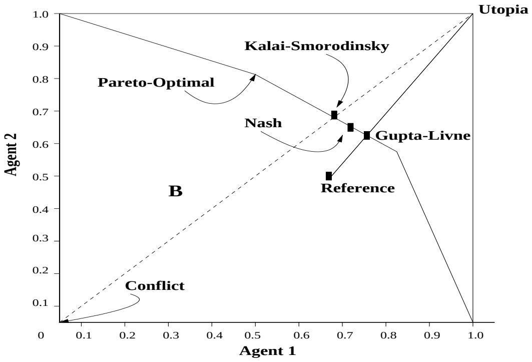

Figure 3.1: Outcome space for a pair of negotiating agents.

are: *Nash Solution* (Nash 1950), *Reference Outcome*, (Raiffa 1982) (Gupta & Livine 1988) and *Kalai-Smorodinsky*, (Kalai & Smorodinsky 1975). The latter solution concept is not expanded on here since the wrapper evaluation is adequately achieved via the first two solution concepts (referred to (Kalai & Smorodinsky 1975) for an exposition). The Nash solution is based on four axioms that must be satisfied (Nash 1950):

- *Invariance under affine transformation*. That is, the particular chosen scale of the utility function ought not change the outcome, only the numbers associated with the outcomes. This axiom is used to prevent the need to make interpersonal comparisons in utility, since negotiators may want or need to transform their utility functions. For example, if one agent has $\pounds 20$ in the bank, and evaluates the deal that gives it $\pounds x$ as having a utility $20 + x$, while another agent evaluates such a deal as having x, it should not influence the Nash solution. That is, a change of origin does not affect the solution.

- *Symmetry*. Also known as the anonymity axiom. This states only the utilities associated with feasible outcomes and the conflict outcome determine the final outcome. No other information is required to select an outcome, and switching the labels of agents does not affect the outcomes.

- *Independence of irrelevant outcomes*. It states that if some outcomes o are removed, but o^* is not, then o^* is still the solution.

- *Pareto efficiency*. As mentioned above, this axiom states the maximum amount of utility that can be reached. Note, this is the maximum attainable amount and not a complete aspiration achievement by both parties (point referred to as *utopia* in figure 3.1 because any gains by one agent above this line result in a loss to another and therefore will not be selected. Indeed, *utopia* can not be the maximum of the gains because of this conflict of interest—one's gain is the other's loss.

The unique solution that satisfies the above axioms is the *Nash solution*, defined as:

$$o^* = \arg\max_o [U_1(o) - U_1(c)][U_2(o) - U_2(c)] \tag{3.1}$$

This corresponds to the points that maximize the *product* of individual utilities for a deal, relative to the conflict payoff c (Nash 1950).[5] When individual utilities calibrate an agent's preferences over certain alternatives, or what is called a value function, as opposed to an agent's preferences over uncertain alternatives (see (Raiffa 1982), (Luce & Raiffa 1957) for an account of risk-less and risky utility functions respectively), the multiplicative form of the Nash solution represents the concern for equity—the product of the value gains is maximized more for more *equal* individual gains. Thus if each agent agrees to the four axioms above, then each is motivated by *proportionate cooperation* (MacCrimmon & Messick 1976). Consequently, both should choose the Nash solution as the outcome. However, if only one agent is not motivated by this proportionate cooperation principle then the the choice of the two agents is not the Nash solution.

The Nash solution is the most popular solution point to the bargaining problem. The other is the *reference* point. This is also observed in experimental bargaining problems where a prominent outcome is used by negotiators to anchor a point in the set of outcomes B (Raiffa 1982). The negotiators can then use this anchorage / reference point as point of improvement to the final point (Raiffa 1982). This point can be used either as a commonly agreed on starting-point, a credible final point, or simply a focal point (Schelling 1960), (Roth 1985). In multi-issue negotiations, the mid point of each issue of both agents' reservation can serve as such a reference point, from which negotiators may attempt to jointly improve (Pruitt 1981), (Raiffa 1982). For example, if the price of a service being discussed between two agents is between $\pounds 0$ (free) and $\pounds 40$ (the buyer preferring values towards 0 and the seller preferring prices closer to 40), then the reference point is $\pounds 20$ for the issue price.

Gupta and Livne's solution formally represents a reference point by replacing the conflict point as an outcome which both parties should attempt to improve jointly (Gupta & Livine 1988). The solution proposed by Gupta and Livne is a point that lies on the pareto optimal line and connects this reference point with the maximum achievement of each party's aspiration levels (utopia, see figure 3.1). This reference outcome has been shown to be appropriate for concession models (log-rolling (Wilson 1969), (Coleman 1973), (Raiffa 1982)) of integrative multi-issue negotiations (Gupta 1989), making it a highly relevant evaluation criteria of the wrapper.

There are other proposed solution points in the space of possible outcomes B which will not be discussed here (see (Corfman & Gupta 1993)). The choice of which solution concept to choose for determining an outcome has itself been problematic, because they are all based on a set of simple and plausible axioms

[5] This is referred to as the *regular* Nash bargaining solution. A *generalized* Nash bargaining solution also exists and this models the "bargaining powers" of both agents. See (Binmore 1992), page 181 for properties of this solution.

(see (Damme 1986) for postulates that provide some solution to this indeterminacy problem). Indeed, problems arise (empirically supported in social psychology findings (Roth 1995)) if each agent is motivated by a different solution concept / social motive. Thus, if designers of different agents are motivated by different social motives, then a difficulty arises over which solution concept to use in axiomatically resolving the conflict. Designers would have to agree *a priori* on a solution concept and the agents would need to be bound to this solution concept independently of their environment. As will be shown below, this is the approach adopted by some computational models of negotiation using principles of mechanism design (see section 3.1.8).

Furthermore, it is interesting to note that the cognitive (motivational) factors of agents are implicitly embedded within the solution concept. Thus a pair of agents who select the Nash solution are motivated by the principle of *proportionate cooperation*. Alternatively, selection of the reference point as a tentative solution to be improved upon indicates the motivation of agents to mutually search for better outcomes. The assumption in the work reported here is that the social motivations of agents should be explicitly represented, and reasoning over which social motive to choose from is a dynamic function of the task-environment of the agent, changing depending on its computational, communicational or task load. The reason for this choice is best illustrated by the following quote:

> ... the distinction between self-interested (competitive) agents that are trying to optimize their own local performance and cooperative (benevolent) agents that are trying to optimize overall system performance is important but not an overriding factor in the design of coordination mechanisms for complex agent societies that operate in open environments. In fact, I feel agents that populate such societies will use performance criteria that combine both local and nonlocal perspectives and that these performance criteria, in terms of the balance between local and nonlocal performance objectives, will change based on emerging conditions. Thus, I see this distinction between self-interested and cooperative agents blurring in the next generation of large and complex multi-agent systems. The basis of this view is that agents that operate in these complex societies and open environments will have to cope with a tremendous amount of uncertainty, due to limited computational and communicational resources, about how to best perform their local activities . . . These factors will lead to self-interested agents behaving in more cooperative ways so that they can acquire useful information from other agents and help other agents in ways which will eventually improve their local performance. In turn, cooperative agents will behave in more self-interested ways given the costs of understanding the more global ramifications of their actions, as a way of optimizing overall performance of the society. (Lesser 1998)

As will be shown later, non-cooperative models are more appropriate for the computational modeling of the negotiation *process*. Nonetheless, the axiomatic models provide a set of useful tools for *analyzing* the performance of the wrapper. Cooperative bargaining models lead to further difficulties because they do not consider the computational difficulties involved in the computation of some of the above solution concepts. These computational difficulties are discussed below in the cases of negotiation over a single and multiple issues. Figure 3.2 a) represents the pareto optimal line and Nash bargaining solution involving only a single issue (distributive bargaining). When only one issue is involved, *all* the possible outcomes lie on the pareto-optimal line—the feasible set. Furthermore, because of the conflicting linear value functions of each agent, the sum of each outcome is 1 (called *zero-sum games* (Gibbons 1992)).[6] The point that maximizes the product of the individual utilities (the Nash bargaining solution) is easily computed as the mid point (and most equitable) of both agents' value function (i.e $(0.5, 0.5)$). The situation is made more complex when

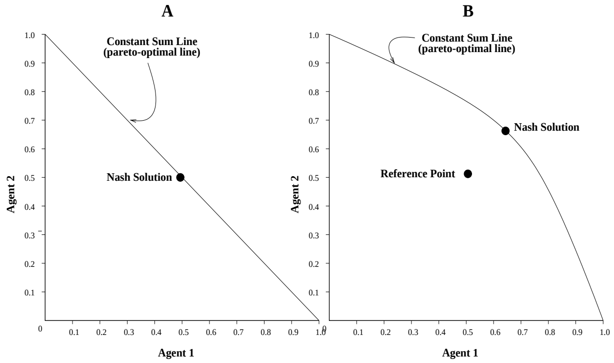

Figure 3.2: Outcome space for a pair of negotiating agents for linear value function and a) single issue and b) multiple issues.

multiple issues are involved. This is important for the types of domains considered in this research where negotiation is over multi-dimensional services. Due to multiple issues, each having a different importance level and linear value function, the outcomes are transformed to a non-constant sum game (where the sum of the individual values for an outcome does not necessarily add up to 1). It is precisely for this reason that agents can look for "win-win" outcomes, improving on the outcome. The pareto-optimal line for integrative bargaining is shown in figure 3.2 b. The only points on this line where the sums of the individual values add to 1 is at the point of connection to the x and y axis. Different points along the pareto-optimal line then do not necessarily add to 1 and do not necessarily have the same addition.[7] More importantly,

[6] The preferences of agents in the work reported here are modeled as a linear additive value function for each negotiation issue. The details of the function and its behaviour are deferred until the next chapter.

[7] Note, the argument is true for a pair of perfectly opposing linear utility functions. The introduction of non-linearity changes the cardinality of values along the pareto-optimal line, meaning that the sum of the individual utilities that lie on the line do not add up to 1.

outcomes of negotiation can now lie below the pareto-optimal line because agents may attach different importance weightings to each of the issues. Thus, an agent who places a lower importance on one issue than another, but possibly more on yet another issue, can result in outcomes that lie below the pareto-optimal line. Compare this to the distributive bargaining case, where the outcome of a negotiation *had* to be on the pareto-optimal line (due to the conflicting linear value functions and the importance weighting of value 1, the sum of individual values has to add to 1). Furthermore, the Nash bargaining solution is no longer at $(0.5, 0.5)$, because the pareto-optimal line has moved from the constant sum line to another point. Indeed $(0.5, 0.5)$ can now be viewed as the focal point.

There are a number of computational implications in integrative bargaining. Specifically, whereas the maximization of the sum of the individual values is computationally straightforward, the same is not true of the computation involving the maximization of the product of the utilities (or the Nash bargaining solution). The Nash bargaining solution is inadequate in cases of multiple issues because its computation becomes intractable in the presence of multiple issue reservation values and weights. The maximization problem then becomes maximization of a quadratic function with restrictions (the reservation values of an issue), where the solution to the quadratic function may violate the restrictions. It is a quadratic problem because the individual utilities of agents are linear:

$$\max\left(\sum_{i=1} w_1^i U_1^i(o)\right)\left(\sum_{i=1} w_2^i U_2^i(o)\right)$$

Numeric methods, such as *active sets*, can handle such problems (Luenberger 1973). However, with this method as the number of issues increases then so does the complexity of the computation involved in solving the quadratic problem. Therefore, active sets become unlikely candidates for computing the Nash solution for bargaining problems involving large number of issues.

To summarize, in this section the theory and assumptions of cooperative games were briefly reviewed. It was shown that although impractical for modeling the processes of negotiation, cooperative game theory has nonetheless produced: i) a formal definition of the possible space of outcomes and how this space can be represented and transformed and ii) a number of global evaluation criteria (such as pareto-optimality, Nash, reference and Gupta-Livne solutions), a number of which will be used in the empirical evaluation phase of this research. Finally, the last section discussed the effect of bargaining problems involving more than one issue on: i) some of the global measures and ii) the computations involved in finding a solution. Implicit in the above arguments was the availability of information in making social decisions. For example, to compute the reference point, or outcomes that actually lie on the pareto-optimal line, agents have to know the utilities the other agent places on all the set of outcomes. The treatment of information in game theory is discussed in the next section.

3.1.5 Complete Information Games

The theory of complete information is not directly relevant to the research reported here. In this research it is assumed that information is private in interactions. Nonetheless, the theory of complete information is reviewed here because it formally represents some important concepts (such as Nash equilibrium) and assumptions of game theory (such as the rationality and common knowledge of agents). Furthermore, the exposition will provide a framework for better understanding a number of computational models of negotiation, reviewed in section 3.2, which are a natural extension of complete information cooperative games.

von Neumann and Morgenstern (Neumann & Morgernstern 1944) classified games into games of *complete* and *incomplete* information.[8] In games of complete information the players are assumed to know all the relevant information—that is, they have knowledge of:

1. *The rules of the game*: The rules, or the protocol of interaction, are a specification of when an agent may act, the actions available at these permissible times and the information concerning the history of the game until the current decision point. A player then formulates a *strategy* for the game, given the rules.

2. *The players of the game*: A player is specified by: a) their preferences: represented as payoffs or a utility function. The utility functions are defined on the set of possible outcomes of the game. b) their beliefs: formally represented by a subjective probability distribution over a set of possible states of the world. It is the combination of the chosen strategies and the states of the world which determine the outcome of the game. States of the world are attributed to chance moves.

More formally, a game is described in normal form as:

Definition 2 *The normal form representation of an n-player game specifies the player's strategy spaces $S_1, ..., S_n$ and their payoff functions $u_1, ...u_n$. The game is then denoted by $G = (S_1, ..., S_n; u_1, ...u_n)$*

Game theory then predicts a unique solution to the game (such as the Nash bargaining solution) as to what each agent will choose. However, in order for this prediction to be true, it is necessary for each agent to be willing to choose the strategy predicted by the theory. Thus, the predicted strategy for each agent must be the agent's best response to the predicted strategies of the other agents. Rationality is then the adherence to this *self-enforcing* property (because no single agent wants to deviate from its predicted strategy), while at the same time maximizing its expected utility.

In a game of complete information, all the above are *common knowledge* (Aumann 1976). The implication is that not only does each agent know it, but also that each agent knows that each agent knows it,

[8] Games of incomplete information are also referred to as "asymmetric information" in the game theory literature (Gibbons 1992).

that each agent knows that each agent knows that each agent knows it, and so on *ad infinitum* (Mertens & Zamir 1985). In addition to this, in a game of complete information the information need not be *perfect*. For example, chess is a game of perfect information, where for each decision node each agent always knows the complete history of the game. Conversely, in a game like poker an agent has *imperfect* information about the history of the game thus far; a player does not know what cards other players hold when at a decision node.

Although the players have common knowledge about the state of the world, their subjective beliefs about what strategy the other player is following are determined by the analysis of the game. The question of which analysis is the appropriate one is itself problematic (Binmore & Dasgupta 1986). In particular, the infinite regress problem means that all strategies appear equally reasonable (Luce & Raiffa 1957). Infinite regression allows reasoning of the kind, *"if I believe, that he believes, that I believe, that he believes, etc."*, which, in turn, makes all possible strategies candidates for selection. To overcome these difficulties, three additional requirements, representing the nature of rationality, are needed:

- c) A rational player quantifies *all* uncertainties using a subjective probability distribution. The player then maximizes its utility given this distribution. Thus the subjective probability distribution is common knowledge to all the other players.

- d) All rational players are computationally equivalent. Thus if one player is given the same information as another, then it can duplicate its reasoning process. This does not mean that an agent knows everything (is omniscient); rather, the agent is infinitely capable of introspecting other agent's reasoning.

- e) Rationality of players is common knowledge. In game theory, rationality requires that an agent maximizes its utility *and* each agent will *necessarily* select an equilibrium strategy when choosing independently and privately.

The implications of assumptions d) and e) are that it is common knowledge that the players are rational (what is referred to as consulting the same game theory book which contains all the commonly held assumptions such as the rationality and beliefs of agents as conventions (Binmore 1992), p. 484). Taken together, it is possible to show that assumptions a) to e) sanction any choice of pair of strategies which are not in equilibrium. In economics, an equilibrium is defined to occur when the actions of an agent are consistent given the actions of others (Gibbons 1992). There are numerous equilibria concepts in game theory, each stricter in sanctioning possible strategies, but the most popular one is the Nash equilibrium.[9] This is formally defined as:

[9] Not to be confused with Nash bargaining solution which was defined in section 3.1.4.

Definition 3 *In the n-player normal-form game $G = (S_1, ..., S_n; u_1, ...u_n)$, the strategies $(s_1^*, ..., s_n^*)$ are a* **Nash equilibrium** *if, for each player i, s_i^* is player $i's$ best response to the strategies specified for the $n-1$ other players, $(s_1^*, ..., s_{i-1}^*, s_{i+1}^*, ..., s_n^*)$:*

$$u_i(s_1^*, \ldots, s_{i-1}^*, s_i^*, s_{i+1}^*, \ldots, s_n^*) \geq u_i(s_1^*, \ldots, s_{i-1}^*, s_i, s_{i+1}^*, \ldots, s_n^*)$$

for each feasible strategy $s_i \in S_i$. That is, s_i^* maximizes:

$$\max_{s_i \in S_i} u_i(s_1^*, \ldots, s_{i-1}^*, s_i, s_{i+1}^*, \ldots, s_n^*)$$

Assumption e) enables plan recognition which, in turn, supports assumption d) and without it an agent is incapable of predicting other agent's behaviour. The assumption states that all agents are rational in that: a) they are utility maximizers and b) they will independently choose an equilibrium strategy. Under assumption d), a rational agent can only model (or predict) the behaviour of another rational agent. However, if assumption e) is violated, in that an agent chooses a non-equilibrium strategy (and hence behaves irrationally by deviating from the Nash equilibrium) then the rational agent can no longer predict the behaviour of the irrational one because of the violation of assumption d). However, the rational agent can derive more utility (by deviating from Nash equilibrium) if it can model this irrationality on the part of the other agent (using another assumption, say d^*). As Luce and Raiffa (Luce & Raiffa 1957), have argued:

> Even if we were tempted at first to call a Nash non-conformist "irrational", we would have to admit that his opponent might be "irrational" in which case it would be "rational" for him to be "irrational".

Therefore, if the rationality assumptions, included to solve the infinite regress problem, are violated, then the outcome of interaction is indeterminate since any non-Nash pair of strategies can be chosen. However, the knowledge that agents are all perfectly rational, or the assumption on the part of the agent that other agents are also rational (consult the same game theory book), does substantially reduce the decision problem of the agent to one of selecting the strategy that is known to be in equilibrium independently of what the other agent does. As will be seen in section 3.2, a similar notion of perfect rationality is also developed in computational models of negotiation where agent designers are provided with negotiation protocols that have known equilibrium strategies. This fact is publicly known and deviation from it is irrational. Therefore, an agent designer can design his/her agent to behave independently of the other's choices.

3.1.6 Games of Incomplete Information

The arguments above concentrated on models of complete information which are suitable for games like chess. However, in real environments agents seldom know as much as the above models assume. What is also required are models of decision making with uncertainty over both the rules of the game as well as the preferences and beliefs of others. Such models are highly relevant to the domain of this research, once again, because of the privacy of information assumption.

Harsanyi developed a model which represents optimal behaviour given the fact that an agent has incomplete information about its world (Harsanyi 1955). Since uncertainties over the rules of a game can be expressed as uncertainties over the payoffs, assumption b) is the most fundamental assumption which needs to be relaxed. If assumption b) is relaxed, then the agents are no longer certain as to the *type* of the other players. To know an agent's type is to have complete knowledge of its preferences and beliefs. Each agent then only knows for certain its own type and its uncertainties of the other agent's type may be expressed as a probability distribution over the set containing all possible types.

Given the above, an agent's uncertainty over the types of others is modeled by introducing a chance move at the first step of the game where nature selects the type of the player of the ensuing game with a probability distribution which is common knowledge to all players. Then, before the game begins, each agent updates its belief about the type of all others, given it has been chosen using Bayes rule. The introduction of the move by nature at the first step converts the game of incomplete information to a game of imperfect information, where at some point in the game the player with the move does not know the complete history of the game thus far.

In essence, uncertainty is dealt with by assuming that the agents have a certain limitation on the form of their utility functions. Thus, there exists a known set of all possible utility functions. Each agent is then assigned a *type* based on which of those utility functions it is currently using. Other agents then update their beliefs about the type of others by acquiring information in the process of interaction. Then the choice problem reduces to a point that is fundamentally the same as a game against nature (for example, probability that it will rain tomorrow, given that it is sunny today) as in a traditional single-agent decision making situation.

3.1.7 Non-Cooperative Games

Non-cooperative models are also known as *strategic bargaining theories*, where the bargaining situation is modeled as a game, and the outcome is based on an analysis of which of the players' strategies are in equilibrium. This type of model was first motivated by Harsanyi (Harsanyi 1956), but is best represented through the *Sequential Alternating Protocol (SAP)* ((Rubinstein 1982), (Rubinstein 1985b), (Osborne & Rubinstein 1990)). The SAP, unlike the cooperative models, models the *process* of negotiation, one of the requirements of the problem domains of this research. The complete information version of the game is described first, followed by the incomplete information one.

There are two players 1 and 2, whose task is to divide \$1, and receive the share they each agree to. If they fail to agree, they get the conflict payoff of \$0. The bargaining process is normatively specified by the sequential alternating protocol where player 1 makes an initial offer of its share for the dollar at stage 0. Player 2 immediately accepts or rejects the offer. If the offer is rejected, then player 2 makes a counter-offer at $T = 1$. This process is repeated until either a successful settlement is reached or else both players receive the conflict payoff. In cases of successful outcomes, the payoff to player 1(player 2) is computed as the share of the dollar agreed at stage t, modified by a *discount factor* δ_t^1 (δ_t^2). The discount factor represents the incentive to reach an agreement early and $\delta_t^1, \delta_t^2 < 1$. Thus in round one the dollar is worth 1, in round two it s worth δ, in round three it is worth δ^2, and so on. A strategy is then a specification of the proposal/reply at each stage of the game as a function of the history to that point.

Since the dollar is an infinitely divisible good, any division of the dollar is a Nash equilibrium. A stronger equilibrium solution was introduced by Rubinstein to solve the indeterminacy problem, called the *subgame perfect* equilibrium (Rubinstein 1982). Subgame perfect equilibrium sanctions commitments to *contingent* courses of action that would result in lower payoff to a player if the contingency did actually arise. For example, a threat by player 1 to walk off from negotiation if it did not receive 90 cents of the dollar is not credible, because if player 2 did offer 10 it would not be in the interest of player 1 to enforce the threat. Thus subgame perfect equilibria effectively prunes the search tree on the assumption that the other agent is rational (see section 2.2.8).

In the above model the subgame-perfect equilibrium is unique and agreements are immediate with player 1 receiving share $(1 - \delta_2)/(1 - \delta_1\delta_2)$, while player 2 receives the share $1 - ((1 - \delta_2)/(1 - \delta_1\delta_2))$. Thus the more impatient an agent (the larger the value of δ), the smaller the final payoff.

For example, consider a *finite* version of the divide the dollar game with $\delta_1 = \delta_2 = 0.9$. Table 3.3 shows the offer's maximal claim that are acceptable to the other agent. Assume that in the last round (T) agent 2 would accept \$0. However, in the next to last round, 2 can keep 0.1, because it knows this is how much 1 will lose if it waits till the next round ($1 - \delta_1 * 1$). This reasoning continues backwards and the process terminates when the time limits of the game has been reached.

Round	*1's share*	*2's share*	*Offerer*
⋮	⋮	⋮	⋮
T-3	0.819	0.181	2
T-2	0.91	0.09	1
T-1	0.9	0.1	2
T	1	0	1

Figure 3.3: Maximal acceptable claims of an offerer for a finite game

Problems occurs when the protocol permits an infinite rounds of bargaining *and* non-discounted offers.

Under such circumstances any splits of the dollar is Nash equilibrium. However, as mentioned above, Rubinstein showed that for an infinite game where offers are discounted then a solution does exist and it is reachable within the first step of the protocol. The proof is as follows. Let the maximum and minimum agent 1 can get in any round be denoted as A_1 and a_1 respectively. Conversely, let B_2 and b_2 denote the maximum and minimum agent 2 can get in any round respectively. The proof consists of showing $A_1 = a_1$ and $B_2 = b_2$. If agent 1 makes the first offer then the maximum it can claim of the dollar has to satisfy the inequality:

$$A_1 \leq 1 - b_2\delta_2 \tag{3.2}$$

That is, the maximum agent 1 can claim on its turn for agent 2 to be indifferent between accepting and refusing is what remains of the dollar once the discounted minimum of agent 2 has been allocated to 2. Conversely, the minimum agent 1 can claim on its turn has to satisfy the inequality:

$$a_1 \geq 1 - B_2\delta_2 \tag{3.3}$$

To see this, suppose 1 offers 2 an offer that violates this inequality, $x < 1 - B_2\delta_2$. Let $x < y < 1 - B_2\delta_2$. Then since $1 - y > B_2\delta_2$, a demand of y by 1 at time 0 will be accepted by 2, because if 2 refuses y then the maximum 2 can get in the next time step is $B_2\delta_2$ which is less than $1 - y$. Thus 2 gets more by accepting $1 - y$ at time 0 than waiting until the next round. It follows that it can not be optimal for 1 to demand an offer x which will be rejected when another demand y exists which will be accepted at time 0. This logic is used to show agreements are reached instantly.

Two further inequalities are then needed to compute the final share each agent receives. These inequalities are derived by exchanging the roles of the agents, giving the requirements of the maximum and minimum demands (B_2 and b_2 respectively) of agent 2 as:

$$B_2 \leq 1 - a_1\delta_1 \tag{3.4}$$

$$b_2 \geq 1 - A_1\delta_1 \tag{3.5}$$

Substituting 3.5 for b_2 in 3.2 gives:

$$A_1 \leq 1 - b_2\delta_2 \leq 1 - \delta_2(1 - A_1\delta_1) = 1 - \delta_2 + A_1\delta_1\delta_2$$

Therefore

$$A_1 \leq \frac{1 - \delta_2}{1 - \delta_1\delta_2} \tag{3.6}$$

Similarly, by substituting 3.4 for B_2 in 3.3 we get:

$$a_1 \geq 1 - B_2\delta_2 \geq 1 - \delta_2(1 - a_1\delta_1) = 1 - \delta_2 + a_1\delta_1\delta_2$$

Therefore

$$a_1 \geq \frac{1-\delta_2}{1-\delta_1\delta_2} \tag{3.7}$$

Therefore, since a_1 and A_1 are the minimum and maximum demands of agent 1, then $a_1 \leq A_1$. Thus 3.6 and 3.7 and the corresponding inequalities for B_2 and b_2 imply that:

$$a_1 = A_1 = \frac{1-\delta_2}{1-\delta_1\delta_2} \qquad\qquad b_2 = B_2 = \frac{1-\delta_2}{1-\delta_1\delta_2}$$

The above model not only addresses some of the key issues identified in chapter two (the protocol of interaction, time, strategies, commitments and costs), but it also has the desirable property that agreements are immediate. However, the SAP's adequacy is weakened for application to the problems of this domain because there are possibilities of inefficient delays and deadlocks when information is incomplete. In the SAP, the problem of incomplete information in a service market would be addressed by specifying a seller and a buyer type (see section 3.1.6), where the seller's type represents the lowest price value for which the seller is willing to sell a service, and the buyer's type represents the highest price the buyer is prepared to pay for the service. Each agent is certain about its type and the uncertainty over the other's type is represented by either a continuous distribution or discrete probabilities (e.g. a buyer with a high or low price valuation). These distributions are common knowledge. Uncertainties can then either be two sided (Fudenberg & Tirole 1983), (Perry 1986) or one-sided (Cramton 1991), (Admati & Perry 1987).

As a consequence of these uncertainties there is no subgame-perfect equilibrium. The analysis is instead made using the stronger equilibrium concept of *sequential equilibria* (Rubinstein 1982), where in addition to specifying a strategy, each uncertain player's belief must be specified given every possible history. Then, a sequential equilibrium is a set of strategies and beliefs such that for every possible history each player's strategy is optimum given the other's strategy and its beliefs about other's valuation. Beliefs are made consistent by using Bayes rules. Since agents are bound to the protocol of communication that permits only the transmission of offers and counter offers, the process of learning other's types through Bayes rule typically requires multiple stages, leading to delays in reaching agreements. However, if the other agent's behaviour is off the equilibrium path, then Bayesian updating is not possible since these off equilibrium paths are assigned zero probability. This may result in incentives for agents to deviate from the equilibrium to increase the number of possible outcomes. Out of equilibrium behaviour cannot be ruled out in games of both sided uncertainty and a sequential alternating protocol (this problem is solvable for one-sided uncertainty and a protocol where the uninformed agent makes all the offers and the informed agent either accepts or rejects offers (Vincent 1989)).

In addition to the above properties, the results from non-cooperative models of the negotiation process are highly sensitive to the particular assumptions made about the bargaining process (Sutton 1986). For example, two-sided versus one-sided uncertainty ((Fudenberg & Tirole 1983) and (Sobel & Takahashi 1983) respectively), finite horizon versus infinite horizon time limits ((Fudenberg & Tirole 1983) and (Rubinstein 1985a) respectively), possibility of strategic delays (Admati & Perry 1987), different bargaining costs (Perry 1986), different offer patterns (alternating versus uninformed player makes all the offers (Rubinstein 1985a) and (Cramton 1991) respectively), all result in a different process of bargaining. For example, the SAP protocol can be altered to allow strategic delays where the players are allowed to make offers at any time after some minimum time between offers has passed. This leads to agents strategically delaying their offers which is interpreted as a signal of the position of the delaying agent (Admati & Perry 1987). Consequently, different outcomes are selected.

In summary the SAP is a more operational protocol for computational purposes than cooperative game theoretic models of negotiation. Not only does it model the protocol of interaction, but it also includes the time of reaching agreements, strategies and commitments in interaction. However, small variations in this protocol, and non-cooperative models in general ((Binmore 1992) page 196) result in the protocol selecting different outcomes. Nonetheless, as will be shown in section 3.2.2, the SAP has been usefully extended by Kraus to solve a number of computational problems.

3.1.8 Mechanism Design

In addition to its explanatory purposes, game theory models are used for the design and implementation of organizations, or of an activity within an organization, where the participants do not share the same goals but there exist opportunities for mutual cooperation as well as real conflict (see (Marschak & Radner 1972) for a theory of the team who share a common goal). Previous sections have concentrated on two different perspectives of how to model interactions between agents. The aim of this section is to discuss how such models can be used to *design* and *implement* interacting systems, an activity highly relevant to computational systems.[10] Indeed, the best example of mechanism design is the various types of auctions that exist on the Internet. Additionally, as will be shown in section 3.2, mechanism design has also been heavily used to design computational negotiation protocols that have certain useful features. Therefore, this section will briefly introduce the key concepts that will assist in later exposition.

The problem of designing and implementing activities is referred to as the “implementation problem” or *mechanism design* where the designer’s preferred negotiated outcome (in terms of some criteria such as social or individual welfare) is derivable from a given specification of the rules of the game (see (Rosenschein & Zlotkin 1994)). It is called a mechanism because what is being designed is not a specific game (concrete utilities), but a *“game form”* (utility types). In general, the aim of mechanism design is to create

[10] Mechanism design can be thought of the problem of design a system that implements a game theory text book, containing the assumptions and implications of the theory. For example, a mechanism is designed such that the players in that mechanism commonly know what the most rational strategy is.

a society of agents who are engaged in a cooperative venture for mutual gains. Rules, laws and regulations (or protocols) are used to define a game which specifies the feasible set of negotiated solutions and eliminate individuals' feasible set of actions. As will be shown in section 3.2.1, mechanism design has been central to computational models of negotiation in MAS, by constructing games whose equilibria have some centrally desired properties(s). However, since the computational models of coordination in MAS come from mechanism design, the principles are described in this section.

The problem of mechanism design is formulated in game-theoretic terms as the *principle agent(s)* problem (Binmore 1990). The most popular application of the principle-agent problem is auctions (see (Sandholm 1999)), where the principle is a seller of some good and the agent(s) can be one or more buyers. The problem then is reformulated as one of devising a selling mechanism that satisfies some features such as efficiency and individual rationality (see section 3.1.4), *given that the seller does not know the reservation values of the buyers*. Because the principle cannot observe the *hidden* reservations, the problem is sometimes called *hidden type*, borrowing from Harsanyi's theory of incomplete information (section 3.1.6). [11] This lack of knowledge is addressed by devising *incentive schemes* that reward the agents that submit bids that are at their true reservation values.

In summary, the principle attempts to induce the agents to behave in a certain manner using a mechanism M. However, the principle does not know the types of agents, but it is common knowledge how chance selects the agents for each buying role. The principle's choice of M then serves as a rule of the game G. The agent's actions in G then determine an outcome o. Given that the agents are rational, then the principle will be offered a choice of outcome o in G that is Nash equilibrium. This o is then said to be *implementable* for the principle—it can get o if it wants it by selecting mechanism M. The decision of whether or not an outcome is implementable is simplified through another principle called the *revelation principle* (Binmore 1992). If a mechanism asks an agent what its type is, then it is a *direct* mechanism. Then based on the declared type the mechanism generates some outcome. If the agents are not asked what their type is, then mechanism is called *indirect*. The revelation principle then states that whatever can be done with an indirect mechanism can also be done with a direct mechanism (called incentive compatible). Thus any social function implemented by an indirect mechanism can also be implemented by a direct one where agents have an incentive to declare their true types.

This simple principle that *"if something can be done, then it can be done by just asking people to reveal their true characteristics"* (Binmore 1992) is useful in designing optimal mechanisms—to decide what outcomes are implementable, it is only necessary to consider outcomes that are implementable by

[11] The Principle-agent problem is studied under the subject of *moral hazards* (Binmore 1992), because the principle is taking a risk if it relies on the morals of agent(s) to carry out what they committed to in a contract. In the literature, moral hazards are also called *hidden action* and *adverse selection* problems (Binmore 1992).

a direct mechanism. As will be shown in section 3.2.1, a number of MAS have used these principles of mechanism design for the design of computational models of negotiation.

3.1.9 An Evaluation of Game Theory

Game theory has proved useful in modeling social phenomena in disciplines such as economics, political theory, evolutionary theory, moral philosophy, social psychology and sociology. The reasons for this success have been its (Castlefranchi & Conte 1997): i) conceptualization of a synthetic, meaningful and formal prototypical context as games which are open to experimental analysis; ii) its ability to predict and explain these games in a manner which does not rely on post-hoc explanation, but rather uses formal and sound notions; and iii) identification and conceptualization of a host of social problems such as free-riding, cheating, reciprocation, coalition formation, reputation and emergence of norms. The first two contributions are highly relevant to the research reported here because the formal elements of game theory permit unambiguous modeling of the decision making involved in negotiation. In addition to providing a "modeling language" the theory provides formal concepts such as Nash solution, pareto-optimality and reference point that can be used to empirically evaluate the developed components of the negotiation wrapper.

In addition to the above, the impact of game theory within DAI has been to (Castlefranchi & Conte 1997): iv) challenge the benevolence assumption as well as notions of common problem, social goal and global utility; v) demonstrate that cooperation can emerge from local utilities; and vi) quantify the costs and benefits associated with actions (e.g communication, exchange and formation of groups as coalition). This emphasis of game theoretic models on local preferences makes them highly appropriate for modeling the type of tasks faced by the wrapper (section 2.2.4). Recall that the task of the negotiation wrapper in this body of work is decision making since no objectively correct answer exists (tasks where an objectively correct answer exists are termed problem solving (Laughlin 1980)). In decision making tasks, the object of coordination is an agent's goals and its preferences over these goals.

However, game theory models have generated considerable debate as to their efficacy and the theory's usefulness in guiding the design of an agent (Castlefranchi & Conte 1997, Fishburn 1981, Simon 1996, Zeng & Sycara 1997, Binmore 1990). An adequate evaluation of game theory, due to the enormity of the discipline, is beyond the scope of this thesis. Therefore, only a few select problems relevant to this research are presented below.

The greatest criticism of game theory from the perspective of the objectives of this thesis is its rationality assumption that i) beliefs are common knowledge, and ii) individuals are optimizers and computationally unbounded.

- The first assumption is appropriate for games such as chess where the choices of the individuals, and their interactions, are written into the rules of the game. Players motivations are also common knowledge—each prefers to win. However, in the real world there is no rule book which describes how individuals actually acquire beliefs. The assumptions are based on an "ideal" world in which beliefs deduced rationally from a common prior can be common knowledge. Yet, the world is not "ideal"—there are imperfections in our knowledge.

- The assumption that individuals are optimizers has also been critically challenged. The question of what is optimal, in game theory models, is independent of actual human behaviour—the question is reformulated from one of how *do* people actually behave to how *should* people behave given that each individual were to maximize his utility. Cognitively inspired modelers and designers state that game theory only models a subset of the cognitive makeup of an agent. In particular, economic rationality is not a model of rationality in *general* but only one of a large subset of human goals (Castlefranchi & Conte 1997). The subjective expected utility model (Neumann & Morgernstern 1944) rules out decisions and behaviours which may be perfectly rational but which are economically irrational. For example, to persevere in an investment which has a lower utility than another investment (sunk costs) may be subjectively rational if the agent *desires* to avoid public admission of failure. Cognitive scientists claim that game theory does not consider the entire set of an agent's goals when formulating the criteria of rational behaviour. This observation is supported by the fact the theory is experimentally unsupported (Roth 1995).

- Related to the above point, is the concern that the theory is one of behaviourism and that it excludes from the models any deliberative intervention. The theory models the actions of an agent given its information set, whereas a satisfactory theory of cooperation requires the modeling of the agent's cognition, especially its goals, motivations and intentions rather than the knowledge only. Furthermore, the theory is silent with regards to the contents of preferences, their legitimacy, their nature and their social desirability (Fishburn 1981).

- In addition, most of the models described above assume perfect computational rationality (assumption d in section 3.1.5). Under this assumption, no computation is required to find mutually acceptable solutions within the feasible range. Furthermore, this space of possible deals is assumed to be fully known by the agents, as are the potential outcome values. Generally, the theory is silent with respect to the actual computational rationality of the agents (Simon 1996). To know a solution *exists* is not to know what the solution *is*. Chess is a classic example of this point. The game has a solution—a strategy for white or black which is either a win or a draw, but the search is computationally complex. Game theory models are of type P_1 (the capacity to generate successful behaviour given available information), whereas a more satisfactory model of rationality may be of type P_3 (the capacity to optimally select the combination of action and computation as opposed to perfect rationality—see section 2.2.8). The perfect rationality of all agents, although useful in designing, predicting and proving properties of a system, is not altogether useful in system design since it:

  - does not exist (physical mechanisms take time to process information and select actions). Hence the behaviour of real agents cannot immediately reflect changes in the environment and will generally be sub-optimal (Simon 1982)
  - does not provide for the analysis of the internal design of an agent; one perfectly rational agent is as good as another. Therefore, what is required are different agent architectures that implement different search mechanisms, capable of heuristically exploring a set of possible outcomes, under both limited information and computation assumptions.

  In particular, as Sandholm notes,

  > future work should focus on developing methods where the cost of search (deliberation) for solutions is explicit, and it is decision-theoretically traded off against the bargaining gains that the search provides. This becomes particularly important as the bargaining techniques are scaled up to combinatorial problems with a multi-dimensional negotiation space as opposed to combinatorially simple ones like splitting the dollar (Sandholm 1999).

- The theory is a closed system. It has failed to generate a general model governing rational choice in interdependent situations (Zeng & Sycara 1997). Instead, the discipline has produced a number of highly specialized models applicable to specific types of inter-dependent decision making (e.g. the von Neumann-Morgenstern solution to two-person). As Binmore notes:

  > ...conclusions (of non-cooperative models) only apply to one specific game. If the details of the rules are changed slightly, the conclusions reached need no longer be valid (Binmore 1992), p. 196.

Classical game theorists claim that the models are prescriptive and consequently cannot invalidate themselves if they were universally adopted by all players (if all agents consulted the same game theory text book—if other agents play according to the theory's prescription then the behaviour prescribed to the agent is already optimal). However, even though the internal logics of the models may be true, the models still remain a poor description of the world.

Other game theorists have addressed some of the above criticisms by replacing prescriptive (or *eductive*) models of rationality, based on omniscient unbounded decision makers, by descriptive (or *evolutive*) models which are based on myopically simple agents (Smith 1982), (Axelrod 1984), (Binmore 1990), (Ito & Yano 1995). The theory has also been criticized for its characterization of individuals as logical and rational agents. Rational theories (or what Binmore calls *eductive* models (Binmore 1990)) are inappropriate for the equilibrium existence and selection problems. The former problem appears in games where the determination of equilibria is problematic and, conversely, the latter problem occurs for types of games that have multiple equilibria (Gibbons 1992). Some game theorists claim that the indeterminacy of deciding which strategies are in equilibria is the result of assuming that the process that brings about equilibrium is a logical and rational process, rather than a "myopic tâtonnenment" (or blind groping) process, similar to evolutionary mechanisms (Binmore 1990). For such theorists, rational behaviour is itself the subject of selection and one that has survived after less successful ones have been eliminated. In humans, the process that brings about equilibrium is very complex, employing coordination mechanisms such as thinking and signaling (Binmore 1990). However, although complex, rational behaviour does exhibit imperfections due to its assuming an infinite capability to reason (perfectly rational). Therefore, it is a mistake to take it for granted that decision makers are perfectly rational [12] and as Binmore notes (Binmore & Dasgupta 1986):

> ...the most important equilibrating mechanisms, as in animal biology, are those which operate through the short-sighted and mechanical adjustment of strategies in the indefinitely repeated play of a game.

There exists a vast literature on the equilibria selection problem which is beyond the scope of the discussion here (see (Gibbons 1992) for an introduction to the problem). It is generally accepted that if the equilibrating mechanism is a rational and conscious process then the choice of which equilibria to select is determined by negotiation among the players of that game (Nash 1951). Conversely, if rational behaviour has been made by unthinking evolutionary forces then the selection problem becomes a meaningless problem since the choice of the actual equilibrium observed is due to random fluctuations in the equilibrating process.

The individual *is* merely a strategy which is subjected to survival criteria in a population of other strategies. The problems associated with the prescriptive models are eliminated by replacing the agents with simple stimulus-response machines—the beliefs, motivations and abilities of the agents are no longer an issue and the equilibrating mechanisms is no longer the reasoning process of the agents but an evolutionary process. Under this methodology rationality itself is a candidate for change. However, the solution is bought at a cost. Descriptive models may address the above problems but they may be too specific by assuming far too much that can be justified, as well as generating dynamic systems that are too complex to analyze.

An additional problem raised is the level of complexity of the agents in the generated descriptive models. For example, even single celled organisms can learn from their experience. Therefore any descriptive model must take into consideration not only the learning aspect of the agent but also the level of complexity of the learning involved (for example, should agents be modeled as learners of other's learning process).

[12] Even professional economists can not be relied on to behave rationally in the simplest of bargaining games (Guth, Schmittberger, & Schwarze 1982).

Learning rules have been suggested as a possible strategy candidate (Smith 1982), (Axelrod 1984) and the criteria of how complex these learning rules are delegated to the principle of bounded rationality, since increasing the complexity of an individual incurs costs (search and management) which, in turn, imposes a constraint on the models of the individual. This bounded rationality will constrain the complexity of the agent.

As was mentioned at the beginning of this section, the approach adopted in this work is to adopt the formal game theoretic constructs such as protocols, outcomes, utilities, and strategies (represented computationally as permissible state-space transitions, terminal states given paths from an initial state, traversal strategies, state utilities and path selection strategy, respectively in search algorithm terminology), as well as solution concepts such as pareto-optimality, Nash bargaining solution and reference point. However, for computational and informational reasons, the assumption that rationality is selection of outcomes that are optimum (lie on the Pareto optimal line) is relaxed. Agents operate in dynamic and uncertain environments, where, at best, even the identity of the other agents is uncertain, let alone the assumption that there is common knowledge of the prior distribution of others' types. The combination of uncertainty and computational boundedness of physical systems, results in a sub-optimal heuristic search that may not be able to select feasible outcomes on the pareto-optimal line. Under such contexts, there is a tradeoff between solution quality and the computational and informational requirements—the optimality of the search outcome will be a function of the certainty levels and the computational efforts.

The computational and domain specificity problems of game theory have also been one of the central concerns of DAI models of negotiation. To this end, a number of key representative computational models from this paradigm are discussed in the following section.

3.2 Computational Models of Negotiation

This section is a description of the class of models which this research is primarily concerned with, namely *computational* agents that use negotiation to further coordination. Sections below describe in more detail models from a mainly MAS perspective (with the exception of the Contract Net Protocol, section 3.2.3, which belongs to the Cooperative Distributed Problem Solving paradigm). The presented work below can be viewed as proposals for the design of negotiation protocols that are progressively less restrictive on the agents and where interactions become more direct.

3.2.1 Domain Theory of Negotiation

The application of mechanism design (see section 3.1.8 above) to different types of computational domains has been central to the work of Rosenchein and Zlotkin, (Rosenschein & Zlotkin 1994). The main idea behind this body of work is that protocols of interaction can be designed that are self-enforcing and that have certain desirable properties for different domains. These properties can then be used by agent designers as a standard of interaction. The assumptions of this body of work are as follows:

1. **Expected Utility Maximizers:** individual decisions are rational only if they maximize the expected utility of an agent.

2. **One-off Negotiation:** Agents' current actions are not dependent on future encounters. This independence of histories on the current encounter is common knowledge.

3. **Inter-agent Comparison of Utility:** Agents, or the designers of agents, have a means of transforming others' utility into a common utility.

4. **Symmetric Abilities:** All agents are capable of performing the same set of actions. Also, the cost associated with each action is independent of the agent carrying out the action. Costs are specified as a part of the agent's utility function.

5. **Binding Commitments:** Designers design their agents to keep all their commitments.

6. **No Explicit Utility Transfer:** Agents cannot explicitly transfer utility between one another—there is no side payment (section 3.1.4). Utility is however transferred *implicitly* as agreements.

Based on these assumptions the authors use the principles of mechanism design to construct protocols of interaction:

> We are interested in social engineering for machines. We want to understand the kinds of negotiation protocols, and punitive and incentive mechanisms, that would cause individual designers to build machines that act in particular ways. Since we assume that the agents' designers are basically interested in their own goals, we want to find interaction techniques that are "stable", that make it worthwhile for the agent designer not to have his machines deviate from the target behaviour (Rosenschein & Zlotkin 1994), p. 4–5.

The function of a protocol is the specification of the set of possible deals agents can make together with the sequences of permissible offers and counter-offers. Properties of protocols are then analyzed so as to guide agent designers' decisions about which protocol to use for different domains. The properties the authors suggest are (note the similarity with the axioms of Nash bargaining solution, section 3.1.4):

1. **efficiency:** agreements should be either Pareto-Optimal or globally optimal. The latter is achieved when the *sum* of the agents' utilities is maximized.

2. **stability:** no agent has an incentive to deviate from the strategy specified by the protocol—"*the strategy that agents adopt can be proposed as part of the interaction environment design*" (Rosenschein & Zlotkin 1994), p. 21.

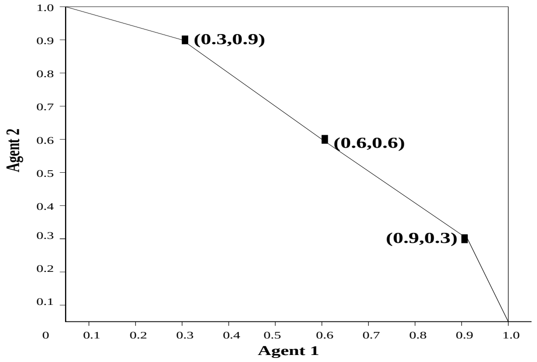

Figure 3.4: Three Outcomes That Maximize the Sum of the Utilities.

3. **simplicity:** related to the two points above is the property that the protocol should make low computational and communication demands on the agent. If a protocol is simple, then fewer system resources are used up by the negotiation. Hence simplicity increases efficiency. Similarly, simplicity is achieved when a protocol is stable, since the agent does not need to spend a significant amount of resource in thinking about the optimal strategy. The optimal behaviour has been publicly provided by the protocol and the best thing the agent can do is to carry out this optimal suggestion.

4. **distributed:** the protocol is not centralized.

5. **symmetric:** the protocol should not favor one agent over another. Symmetry implies that the outcome of the negotiation will not be affected if an agent was replaced by another of exactly the same type.

The efficiency property of a protocol relates to the social welfare function that it implements, here it is the sum of the agents' utilities. Requiring that the sum of the utilities be maximized reduces the number of possible outcomes and rules out many social behaviours. However, Arrow's impossibility theory remains (section 3.1.3)—even though some outcomes are ruled out, there are still multiple outcomes that maximize the social welfare (equity), but each agent prefers a different social outcome (efficiency). This is represented in figure 3.4, where each of the three hypothetical points maximize the sum of the individual utilities. The point shown by the utility vector $(0.3, 0.9)$ is preferred by player 2, since it gives more weight to player 2. Conversely, the point at utility vector $(0.9, 0.3)$ is preferred by player 1, since it gives more weight to player 1. Therefore, each agent prefers a different outcome. Negotiation, then, is defined as *reaching an agreement over the division of the group utility*. The regular Nash bargaining solution (section 3.1.4) is used to solve this fairness problem, resulting in the selection of point $(0.6, 0.6)$. [13] If there is more than one Nash

[13] Note these different solution points on this efficient line (such as $(0.3, 0.9)$ or $(0.9, 0.3)$) can be selected using the *generalized* Nash bargaining solution which models the power, or weight, of the negotiators (Binmore 1992).

solution (recall that the Nash bargaining solution maximizes the product of the deal) then the protocol will select the deal that maximizes the sum and the product of the utilities. Finally, if there is more than one deal that maximizes the product and the sum of the utilities, then the protocol randomly selects one deal. The authors then concern themselves with the class of protocols that satisfy these efficiency criteria. They refer to this type of protocol as *Product Maximizing Mechanisms* or *PMMs*—or the Nash bargaining solution. [14]

The stability property of the protocol is its ability to select and *maintain* equilibrium strategies. This is a highly advantageous property for open societies where malicious agents can enter with their own strategies and attempt to extract the best deal for themselves (what economists call extracting the entire surplus from the interactions (Binmore 1992)). However, if strategies are stable then they are the best responses irrespective of the private strategies of others and the protocol is immune to attack (Smith 1982).

The simplicity property is derived directly from the revelation principle introduced in section 3.1.8. Strategies are simple because PMM protocols are direct, giving agent designers the incentive to declare their utility types (see incentive compatibility in section 3.1.8).

Given this set of properties, an agent designer is then told that for domain D: protocol $Pr1$ is distributed, symmetric, stable, simple but inefficient; and $Pr2$ is distributed, symmetric and stable, but more efficient and complex. The novelty of the approach is this domain theory of negotiation, which can be used for classifying interaction types and assisting designers to choose the appropriate negotiation protocols. The domains they suggest are:

- **Task Oriented Domains (TOD):** Agents in TOD attempt to achieve their tasks, which do not interact with other agents' tasks. However, benefits can be gained by all parties under certain task redistribution patterns. These are inherently cooperative domains, where agents attempt to find mutually beneficial task distributions.

- **State Oriented Domains (SOD):** SOD represents classic AI problem domains, where agents attempt to move the world from an initial state to a goal state. In comparison to TOD, real conflict is possible in SOD because the agents have different goals and there may be no single goal state that mutually satisfies all the agents.

- **Worth Oriented Domains (WOD):** In WOD agents can express a desirability scale, or worth function, to potential outcomes. In both TOD and SOD agents can only *wholly* satisfy their goals (in TOD a goal is completion of tasks, in SOD a goal is a state an agent wishes to reach); they cannot relax their initial goals to reach an agreement. In WOD, however, a continuous worth function (as opposed to the binary functions of TOD and SOD) allows agents to compromise on their goals, and even increase the overall efficiency of the agreements. Negotiation is then cooperative.

[14] A mechanism, in their terms, is both the protocol and the strategy.

Overall, agents can compromise and reach deals over how much work they do (TOD), which final state they reach (SOD), as well as how much worth they extract from the deals. In the types of problems considered in this research, agents do have conflicting goals and conflict resolution is assumed to be a concession over demands. Indeed, some of the most interesting results from integrative bargaining come from the ability of agents to concede and/or make demand on goals.

3.2.1.1 Evaluation of Domain Theory

The work of Rosenchein and Zlotkin has been pioneering in its contribution towards the design of protocols of negotiation for MAS. In addition to being the first to apply cooperative game theoretic models and mechanism design to computational agents, the domain theory of negotiation has been particularly useful in guiding the design of different negotiation protocols for different domains. However, in adopting the Nash solution and principles of mechanism design the approach inherits the criticisms raised in section 3.1.9.

More specifically, a domain theory of negotiation is a step towards developing a general theory of negotiation (one of the criticisms outlined above in section 3.1.9), but, like most game theoretic models, at the cost of making further assumptions that are unrealistic. For example, the fourth assumption above states that agents have the same ability. This allows the modeling of symmetric interactions where negotiation is seen as the optimal sharing or swapping of a set of tasks (in TOD), or the desired final states (in SOD) or worth (in WOD). Worth in WOD is shared implicitly when an agent *"agrees to do more work in a joint plan that achieves both agents' goals, he increases the utility of the second agent"* (Rosenschein & Zlotkin 1994), page 150. However, in the domains of interest of this research, agents do not have symmetric abilities and they cannot trade off worth with tasks. In fact, agents interact and negotiate for services which they themselves cannot perform in the first place. Negotiation then is not about swapping, but rather delegating tasks to other agents to perform. The worth of a goal can no longer be traded off against tasks.

There may also be circumstances when the social function (or the global utility) cannot be maximized due to not only the uncertainty and computational boundedness of agents, but also the structure of the problem domain. One possible way to increase the global utility function (but not maximize it, again due to privacy of information or computational limitations of agents) is to search for *"win-win"* outcomes in *integrative* bargaining, involving more than one negotiation issue, as opposed to distributive bargaining over, for example, tasks, states or worth only. As mentioned earlier, real world problems are seldom described with preferences over a single issue. Furthermore, in the domains targeted by this research, agents cannot exchange tasks. These two points taken together mean that the protocols developed by Rosenchein and Zlotkin are inappropriate for the problem addressed in this thesis—the global maximization of utility by the PMM protocol depends on the exchange of tasks, states or worths. There is a need for other search mechanisms that solve problems that do not just involve exchange *and* that attempt to *increase* the social welfare.

For the above reasons, the generality of the domain theory is restricted to domains that are characterized by the trading of the goals (or tasks, states or worth).

Furthermore, the assumptions that the cost of an action is independent of the agent that carries it out and that each agent has sufficient resources to potentially handle all of the tasks of all agents are unrealistic. These assumptions are clearly violated in real world problems such as scheduling (see section 3.2.4 below for an in-depth discussion) where agents are endowed with different tasks, resources and costs to achieve them. The implications and consequences of asymmetry for a general domain theory are themselves research questions and ones that the authors do not address. The modeling of cost and its assymetric nature has been one of the central contributions of the work of Sandholm (section 3.2.4).

Finally, the authors use principles from mechanism design to transform direct to indirect interactions, in a similar manner to auctions. The declaration of preferences or any information to a principle (either an auctioneer or the protocol) achieves some desirable properties such as efficiency, simplicity and stability, thereby addressing the bounded rationality problem of agents since agents don't need to out-guess others' strategies or engage in costly deliberations for strategy selection. Thus agent designers know what the optimal strategy is for a given domain and they program such behaviours into their agents. In this way, the protocol is restrictive; agents are free to choose any strategy they wish, but the best strategy is public knowledge and deviations from it are irrational. However, mechanism design is ineffective if agents, or their designers, fail to agree to declare their types to a protocol designer. Incentive mechanisms can be constructed to implement a direct mechanism *only after the designers have agreed to reveal their types*. This is in effect a pre-negotiation negotiation among the designers. The theory is not applicable if there are no such agreements between the designers themselves. Interactions therefore need to be direct, and mechanisms are needed that assist agents in the direct interactions with one another when their preferences are private knowledge. The authors do not assess the implications on the behaviour of protocols when the assumption that agents, or their designers, can compare other agent's utilities (assumption three) is violated. Agents may refuse to reveal their utilities.

3.2.2 Non-Cooperative Computational Negotiation

A number of key principles from mechanism design (section 3.1.8) and non-cooperative models (section 3.1.7) for problems that involve time and resource restrictions in worth oriented domains have been central to the work of Kraus; see (Kraus 1997b, 2000) for an overview of this body of work and (Kraus & Wilkenfeld 1995, 1993, Kraus, Wilkenfeld, & Zlotkin 1995, Kraus & Lehmann 1995, Kraus 1997a) for details of the models. In this body of work, strategic models of negotiation have been applied to bilateral and multi-lateral negotiations, single and multiple encounters, complete and incomplete information in negotiations, as well as the impact of time on the utility of deals. The contribution of this body of work is its ability to:

- provide the agent with domain dependent utility functions that take into consideration the passage of time and the costs of negotiation. In the work described in the previous section, “the source of the utility function or the preferences of the agents, …, was rarely discussed. It was assumed that each agent knows its utility function (and has some knowledge of its opponents’ utility function). However, a designer of an automated agent is required to *provide* the agent with a utility function or a preference relation. Without doing so, formal models cannot be used for automated agents” (Kraus 2000).

- model power relationships. In the types of problems considered by Kraus, in the process of negotiation one agent can gain while another loses utility. Therefore, “the stronger agent may be able to “force” the other agent to reach an agreement which is best for it, among the deals that are possible” (Kraus 2000).

- models strictly conflicting preferences, where agents’ preferences are diametrically opposite. For example,“if two agents need the same resource at the same time, and each would like to use it as much as possible, then their preferences are conflicting” (Kraus 2000).

- tackle the computational problems of “developing low complexity techniques for searching for appropriate strategies” (Kraus 1997b), p. 84.

In more detail, Rubinstein’s strategic sequential alternating model (section 3.1.7) has been modified to provide a unified solution to both task and resource allocation problems. These modifications include the modeling of: i) the way time influences the preferences of agents, ii) the discrete, as opposed to continuous, outcomes, and iii) the possibility that both agents can opt out of the negotiation as well as their preferences for doing so. The model is evaluated by the amount of time it takes to reach deals, as well as the efficiency, [15] simplicity and stability of the deals.

Agents’ preferences over the time of the outcome are achieved by building time-dependent preferences into their utility functions. Moreover, Kraus argues that whereas formal theories all acknowledge the importance of a utility function, none of the actually provide any such function. This makes them unoperational; a designer of an automated agent is required to provide the agent with a utility function or a preference relationship. The actual utility function is likely to be domain dependent, but Kraus identifies three categories (Kraus 2000):

1. **Fixed loses/gains per time unit:** $U^i(o,t) = U^i(o,O) + t.C_i$, where o is an outcome, t is the current time in negotiation, O is the set of possible of outcomes, and C_i is the cost/gain to agent i. Each agent

[15] Efficiency in this work is viewed not in terms of pareto-optimality. Rather, in resource allocation problems an efficient outcome is one where the resource is not in use only when no agent in the group needs the resource.

has a utility function that carries a cost gain or loss, due to delays, for each period of negotiation.[16] Costs may be communication load, negotiation costs, resource storage costs or task execution costs, and gains can be the usage of the resource which is the subject of negotiation.

2. **Time constant discount rate:** $U^i(o,t) = \delta_i^t U^i(o,O)$, where $0 < \delta_i < 1$. Similar to the SAP where each agent has a fixed time discount rate that modifies the utility of an outcome.

3. **Finite-horizon models with fixed losses per time unit:** $U^i(o,t) = U^i(o,O).(1 - t/\hat{N}) - t.C$ for $t \leq \hat{N}, C \in R$, where $\hat{N}$ is a finite number of steps in negotiation. Like the previous case, there is a constant gain or loss over time during the negotiation process. However, the utility function also quantifies the gains *after* the end of the negotiation, when the outcome of the negotiation is valid for $\hat{N}$ periods and at each time step after the end of negotiation the agents can gain $U^i(o,O)$.

It is these preferences over time, together with agents having the option to opt out, that motivate them towards reaching deals. However, since time plays no important role in the agent's utility models described in the domain theory of negotiation, presented in the previous section, new strategies are provided. Strategies are, like classic strategies, any function that maps the history of the negotiation to a next move, specifying what the agent has to do next. At each turn of an agent to respond, a strategy specifies i) which offer to make, and ii) whether to accept or reject an offer or alternatively opt out of negotiation. It is this evaluation component of the strategy that is different from the strategies presented in the previous section where time is taken into account.

Given the possible set of outcomes and the agents' utility functions, an agent's strategy is then analyzed using subgame perfect equilibrium (for games of perfect information) and sequential equilibria (for games of incomplete information) as solutions (see section 3.1.7) that any agent will necessarily select if it was rational. Given this property of the non-cooperative model, a mechanism (or the rules of the alternating sequential protocol) is designed that is incentive compatible with selecting the subgame perfect equilibrium strategy for games of perfect information and sequential equilibria strategy for games of incomplete information.

Another major contribution of this body of work is an implementation that addresses the issue of the complexity involved in having to compute strategies, rather than having the equilibrium strategy publicly known (Lemel 1995):

> The drawback of the game theory approach is that finding equilibrium strategies is not mechanical (computational): an agent must somehow make a guess that some strategy combination is in equilibrium before it tests it and there is no general way to make the initial guess (Kraus 1997a), p. 48.

[16] The range of these utility functions are not in the interval $[0, 1]$.

The implementation solution Kraus proposes is to store strategies in libraries represented as **AND/OR** trees where the internal nodes consist of conditions (such as the possibility of opting out of negotiation, the cost of negotiation, the time left in negotiation or the number of negotiators) and the strategies are stored in the leaves of the tree. These strategies, in turn, consist of compiled functions with variables, some of which are already instantiated during the search in the tree, and others of which are instantiated during the execution of the function.

3.2.2.1 Evaluation of the Non-cooperative Computational Negotiation Model

One of the key driving forces of agreements in the work of Kraus is the time and cost consideration in negotiation. The agent's decision problem is formulated as the selection of an offer that maximizes the utility given the time and costs involved. However, although useful, no model of time or cost consideration is provided. Even if such models were provided, they are likely to be domain dependent reflecting the concerns of the domain. Furthermore, "building" into the agents' utility functions additional deliberation factors can result in functions that are over complicated and difficult to design and analyze. This task is not easy for an agent designer who is not an expert in utility theory. Instead, what is required for a flexible and configurable negotiation wrapper are utility functions that are domain independent. To achieve this, simpler utility functions are sought that evaluate the worth of the offer *independently* of the time and cost considerations. These considerations, and indeed any other environmental consideration(s) such the behaviour of the other agent, are delegated to other agent's deliberation mechanisms. These mechanisms then generate offers, each possibly having a different worth to the agent, based on a number of environmental considerations. In this manner a single generic utility function can be provided to the designer inside the negotiation wrapper who can then add additional mechanisms in a modular fashion without affecting the utility function.

Finally, in this thesis only the protocol of the SAP is used to model the process of negotiation, because the assumption that agents "consult the same game theory book" (see section 3.1.5) is not a valid assumption and also because small variations in the parameters of the SAP (brought about by making different set of assumptions) lead to indeterminacy of equilibrium strategies and inefficient delays (see section 3.1.7).

3.2.3 The Contract Net Protocol

The contract net protocol (CNP) is a classic example of a DPS system (cooperative solution synthesis through a decentralized, loosely coupled collection of problem solvers—see section 1.3) used for the task distribution phase of cooperative problem solving (Smith 1980). Therefore, it does not belong to the class of quantitative models of bargaining, although its operation closely resembles a market-like mechanism. The protocol focuses on the traditional problem of how to resolve disparate viewpoints in task allocation problems in a simulated distributed sensor network for acoustic interpretation. Nonetheless, it is included here because: i) it was traditionally the first negotiation protocol in DAI, ii) it models contracts and iii) its extension by Sandholm (section 3.2.4) brings it into the class of quantitative models of negotiation.

The CNP was motivated by the problem that distribution by its very nature requires supplying problem solvers with only a limited local view of the problem. However this conflicts with the desire to achieve global effects (solution to a problem). Therefore, coordinated activity within the system cannot be guaranteed. To overcome this problem, the CNP solution was derived as a mechanism that extends across the network nodes and that can be used as the foundation for cooperation and organization. Cooperation is designed into the system through a communication protocol which facilitates and organizes communication among entities *and* a problem solving protocol which organizes the group of problem solver's activities. The two protocols bring about form and content respectively; how to communicate and what to solve. The discussion below will center mainly around the problem solving protocol because it is the most relevant model for the decision processes involved in the negotiation wrapper.

The CNP consists of a collection of nodes, referred to as *contract net*, where each node in the net may take on the role of a *manager*, responsible for monitoring the execution and processing the result of a task, or a *contractor*, responsible for the actual execution of the task. Roles can be adopted dynamically by all nodes at runtime, therefore nodes are not *a priori* tied to any particular control hierarchy. The negotiation process is then initiated by the generation of a new task by a node. That node announces the newly generated task using a *task announcement* message and becomes the manager of that task. Other nodes in the network evaluate their level of interest in the announced task with respect to their specialized resources (e.g. hardware). If the task is of sufficient interest, a node then submits a bid which indicates the execution capabilities of the bidder. The manager may receive several bids for a single task and it then selects one or more of the bids (based on the information regarding the execution capabilities disclosed in the bid). The selected nodes then assume responsibility for the execution of the task and each is called a contractor for that task. The contractor may need to subdivide the task into sub-tasks and become the manager for these tasks. The manager may also terminate contracts and the contractor can inform its manager of either the partial or completed state of its task(s).

Sandholm compares the CNP to a directed government contracting scheme, where *"each party is allowed to make one bid for each announcement it receives, and the bids of the other parties are not revealed to it. The negotiations are directed in the sense that an announcement is not sent to all other agents—only to likely contractees"* (Sandholm 1996).

The description above, although simplistic, has a number of important contributions. Firstly, commitments are explicitly represented as contracts—a contract is an explicit agreement between nodes. Furthermore, compared to the game theoretical models of section 3.1 the *process* of negotiation is also explicitly represented in the protocol:

> ...establishing a contract is a process of mutual selection. Available contractors evaluate task

> announcements until they find one of interest; the managers then evaluate the bids received from potential contractors and select the ones they determine the most appropriate. Both parties to the agreement have evaluated the information supplied by the other and a mutual selection has been made (Smith 1980).

In summary, the CNP framework provides a mechanism for coordinated behaviour that is symmetric (that is both the caller, or manager, and the respondent, the contractor, have a selection to make) [17] through: i) the concept of negotiation as a mechanism for interaction, ii) a common language shared by all nodes and iii) the announcement-bid-award sequence of messages which offers some support for cooperation since due to incomplete knowledge, the messages give a node an understanding of who else has the relevant information.

3.2.3.1 Evaluation of the CNP

The CNP provides a coordination architecture which is distributed and addresses a number of factors described in chapter two. It has been applied to job dispatching among machines within a manufacturing plant (Parunak 1987), allocation of computational jobs among processors in a network (Malone *et al.* 1988) (where the choice of processor is based on expected completion time), and to distributed meeting scheduling (Sen 1994). However, the protocol has a number of limitations which are borne out of the fact that it belongs to CPS system. In particular, cooperation is an integral part of the protocol. There cannot be any conflict between the agents to start the CNP. Furthermore, in non-cooperative domains the search for acceptable solutions may be more elaborate than two messages—negotiation, especially in uncertain and open environments, is an iterative process of search for possible agreements. In addition to this, the CNP is a theory of the system architecture and is silent with respect to the agent architecture. This latter problem was addressed by the work of Sandholm, described next.

3.2.4 The Contracting and Coalition Model of Negotiation

A decision theoretic agent architecture for the CNP that solves some of the limitations of the CNP was proposed by Sandholm. Additionally, he developed a game theoretic negotiation mechanism that normatively and quantitatively solves the computational difficulties of game theory (the problem of bounded rationality of selfish agents). Sandholm notes that:

> ... the traditional CNP is not an off-the-shelf mature technology that can be applied to different domains as is. The protocol really includes an enormous numbers of design alternatives. ... For example, previous work on the CNP has not addressed the risk attitude of an agent toward being committed to activities it may not be able to honor, or the honoring of which may turn out to be unbeneficial. Additionally, in previous CNP implementations, tasks have been negotiated

[17] Symmetric autonomy of both parties (or bi-directional selection of caller and respondent) was first modeled in *PUP6* (Lenat 1975) which viewed selection as a *discussion* between the caller and potential respondents.

> one at a time. This is insufficient, if the effort of carrying out a task depends on the carrying out of other tasks. The framework is extended to handle task interactions, among other methods, by clustering tasks into sets to be negotiated over as atomic bargaining items. ... . Finally, the question of local deliberation scheduling in the negotiations has not been discussed earlier, ... The hypothesis is that distributed contracting can be developed into an efficient—in terms of results and computational complexity—interaction mechanisms for self-interested agents whose rationality is bounded by limited computational resources (Sandholm 1996), p. 67–68.

From this, it can be seen that commitments to, and the efficiency of contracts given the computational boundedness of agents are the main concerns of the work. The type of problem considered for negotiation is the distribution of agents' tasks. However, tasks can be achieved by other agents and each agent has assymetric costs (compare to the work of Rosenchein and Zlotkin, section 3.2.1, where an agent's task set could also be performed by other agents. However, costs are not assumed to be symmetric). Given that agents have differing costs and are capable of performing others' tasks, a task reallocation mechanism [18] can be prescribed that is beneficial to all agents through cost savings.

Concrete domains that influenced the design of, Sandholm's negotiation mechanism were the distributed vehicle routing problem, the production planning and scheduling problem in manufacturing, and meeting scheduling (Sandholm 1996). The second scenario is expanded below to better illustrate not only the contributions and drawbacks of this quantitative line of work, but also the limitations of the symmetric cost assumption made by Rosenchein and Zlotkin's domain theory (section 3.2.1).

In the manufacturing production planning and scheduling problem, an agent has a set of tasks (such as manufacturing operations and setup operations) and a set of resources (such as machines, people and storage area). The problem then is the scheduling (planning the assignment of tasks to resources for given time windows) for the execution of the tasks on the resources. The problem structure has many cost functions (e.g. minimization of lateness of jobs or completion time). These cost functions, also referred to as objective functions by Sandholm, are subject to constraints such as the order in which tasks can feasibly be executed or the resource capacity. The combination of the objective functions and the constraints define a constrained optimization problem. Furthermore, different manufacturing enterprises can handle the same operations. Therefore, there are potential savings that can be achieved by negotiation. Another feature of the considered domains is that different enterprises may behave cooperatively or selfishly.

In summary, the features of the problems considered are:

- problems are combinatorially difficult. The solution costs and feasibility of the task distribution problems limit the rationality of agents, since they cannot locally compute the costs and benefits

[18] In this subsection, mechanism is interchangeably referred to as protocol and (reallocation) algorithm.

associated with delegating or accepting tasks to other agents *exactly*.

- the asymmetric costs among agents for handling others' tasks often makes it beneficial (individually rational) to reallocate tasks among agents.

- individual members (companies in the case of manufacturing or centers in the case of vehicle distribution routing) can form *virtual enterprises* by joining together and cooperatively, although the intention of each individual is selfish, taking care of production or delivery tasks more economically than if performed individually.

- agents can be selfish or cooperative in task allocation. Cooperative agents attempt to maximize social welfare, measured as the sum of the agent utilities. They are willing to accept task distribution allocations that lower their individual utility but increase the utility of the group. Selfish agents, on the other hand, want to maximize their own profit without regards for other distribution centers or manufacturing companies involved in the virtual enterprise.

The second feature is where Sandholm's work diverges from that of standard game theory. This is because his notion of individual rationality is different from the game theory concept of individual rationality as maximization of payoff. For Sandholm, an agent may reject an individually rational contract if it believes it will be better off waiting for a more beneficial contract that cannot be accepted if the former contract is accepted. Likewise, an agent may accept a non individually rational contract in *"anticipation of a synergic later contract that will make the combination beneficial"* (Sandholm 1999), p. 237.

Given these features, Sandholm presents a negotiation model that addresses three areas of negotiation: *contracting*, *coalition formation* and *contract execution*. In contracting negotiations (referred to as the contracting protocol), agents iteratively reallocate tasks amongst themselves to reach a globally more desirable solution. Whereas in contracting all the agents work in one large coalition, in coalition formation game theoretic normative models are used to analyze the stability of coalitions of agents (the so called *"virtual enterprises"*) where task allocation and problem solving are *"pooled to occur centrally within each coalition"*. Finally, in contract execution an exchange mechanism is developed that solves the problems that occur in honoring task execution in environments where agents may *"vanish easily, and the connection between the agent and the real world party it represents is often hard to detect"*. In this thesis negotiation is in the main between two agents, thus coalitions of large numbers of agents are not possible. Therefore, the coalition protocol is not relevant to the research reported here. Likewise, in this research the problem of contract execution is not addressed. All that is said is that there are execution monitoring protocols (or a commitment model (*commitment model* in figure 1.1, section 1.2)) that can be added to the service execution phase of the service life cycle, that assists in the execution phase. Coordination mechanisms are sought for only the service provisioning phase. Although it is acknowledged that negotiation can be successfully applied to service execution, the object of this thesis is focused on the provisioning phase. For these reasons, only the contracting contribution of the work is detailed below.

Contracting negotiation, developed as the Transport Cooperation Net (TRACONET (Sandholm 1996)), addresses the CNP problem mentioned earlier; namely how to formally model announcing, bidding and awarding decisions involved in the contracting of tasks (Sandholm 1996, 1993). These decisions are based entirely on marginal costs for performing a task. Marginal costs are formally presented below, but informally they represent the difference between the total cost of having to perform another agent's task as well as agent's own task and the set of agent's own tasks. Agents pay one another to perform tasks. Because decisions are based purely on the marginal costs (defined next) analysis (as opposed to the CNP where agents *freely* perform the tasks of others) this pricing mechanism generalizes the CNP to work for both cooperative and selfish agents.

Sandholm defines the task allocation problem as follows (Sandholm 1999), p. 234. The task allocation problem is defined by a set of Tasks T, a set of agents A, a cost function $c_i : 2^T \rightarrow I\!R \cup \{\infty\}$ (or the cost agent i incurs by handling a subset of tasks) and the initial allocation of tasks among the agents $\langle T_i^{init}, \ldots, T_{|A|}^{init} \rangle$, where $\cup_{i \in A} T_i^{init} = T$, and $T_i^{init} \cap T_j^{init} = \emptyset$ for all $i \neq j$. Given this definition, the decision schemes for computing offers are as follows. When an agent makes an announcement for a task, it tries to buy some other agent's capability to perform a task. In announcing, an agent specifies the maximum price it is willing to pay for its task(s) to be carried out. Call this $\rho^{announce}$. When agents make a bid for an announced task, the agents try and sell their services at a bid price, uttered in a bid. Call this ρ^{bid}. Given an announcement and a bid, a reward is then a contract between two agent, details of which are described below. Then, an agent i will make an announcement if:

$$\rho^{announce} = c_i^{remove}(T^{announce}|T_i)$$

where $c_i^{remove}(T^{announce}|T_i)$ is agent $i's$ marginal cost for removing the task set $T^{announce}$ from all of its tasks T_i:

$$c_i^{remove}(T^{announce}|T_i) = c_i(T_i) - c_i(T_i \cap T^{announce})$$

where $c_i(T_i)$ is the cost of optimally achieving all the tasks T for agent i and $T_i = T_i^{init}$. Sandholm suggests the use of approximation schemes for computing $c_i^{remove}(T^{announce}|T_i)$, since it is intractable for most types of problems.

When an announcement has been received by an agent, an agent sends out a bid, ρ^{bid}, if the maximum announced price $\rho^{announce}$ is higher than the price that the task will incur on the agent to perform it. Bidder j bids according to:

$$\rho^{bid} = c_i^{add}(T^{announce}|T_j)$$

where $c_j^{add}(T^{announce}|T_j)$ is agent $j's$ marginal cost for adding the task set $T^{announce}$ to all of its current tasks T_j:

$$c_j^{add}(T^{announce}|T_j) = c_j(T_j \cup T^{announce}) - c_j(T_j)$$

Again, marginal costs are computed using an approximation method. Finally, the awarding price, ρ^{award} is computed using a new task set of the announcer, $T_i^{'}$. This is because the task set of i may have changed within the window of announcing the tasks and waiting to receive all bids. ρ^{award} is computed as:

$$\rho^{award} = c_i^{remove}(T^{announce}|T_i^{'})$$

If ρ^{award} is greater than the lowest bid, the task is awarded to the least expensive bid and *"by convention"* the contract takes place by the awarder paying the bidder the price $(\rho^{bid} + \rho^{award})/2$.

The protocol (or, as Sandholm refers to it, the algorithm) is as follows. Initially each agent computes a solution to the tasks in its own task set (referred to as the local optimization problem). Then, each agent can potentially negotiate with other agents to take on some of its task or, alternatively, take on some of their tasks for a price. Note, that agents in Sandholm's work are allowed to make side payments for the task allocation problem through payments, whereas for agents in Rosenchein and Zlotkin's work no side-payment is allowed. Negotiation is then the exchange of task sets that are profitable (i.e. at a lower cost—referred to as individually rational). The task redistribution protocol is then an iterative exchange mechanism that increases the global utility of agents by traversing a sequence of task allocation configurations among agents. At every step of the iteration, an agent computes a feasible solution for the tasks it has been allocated (a feasible solution consists of an agent assigning resources for the tasks allocated). The task re-allocation procedure is a real-time, anytime hill-climbing algorithm. It is real-time because at each iteration a price equilibrium has to be reached in the task set exchanges before the next iteration—after each contract is made the exchange of tasks and payments are made immediately. It is anytime because the algorithm can be terminated at any point in time and a solution is available that is both individually rational to all the agents and is globally better than the initial solution if each agent carried out its tasks individually. It is hill-climbing because at each iteration a global solution closer to the optimum is reached (in a distributed manner). In comparison, the PMM protocols of Rubinstein and Zlotkin are not anytime. Agents first reveal their costs for all possible task distributions. Then the PMM selects the allocation that maximizes the sum of the utilities and assigns payoffs according to the Nash bargaining solution. This is not anytime because all task allocations have to be evaluated *before* any agreement is reached.

Contracts in a contracting protocol are given search operator semantics by Sandholm. That is, if the task reallocation (or contracting) protocol is interpreted as a global hill-climbing algorithm, then contracts can be interpreted as its search operators. The search for a global optimum is also made more efficient by supplying the contracting protocol with different contract types. Rather than negotiating over single tasks, one at a time, Sandholm shows that a hill-climbing algorithm can reach an optimal task allocation, in a finite number of steps, when agents combine *clustering*, *swap* and *multi-agent* contracts into a *single* contract called an *OCSM-contract*. *O* contracts are over a single task (as in the original CNP) and they are shown to lead the reallocation algorithm into local minima — where contracts are individually rational (agents are better off with the contract), but are not globally optimum. In cluster contracts, *C* contracts, a set of tasks is contracted from one agent to another, whereas in *S* contracts a pair of agents swap tasks. Finally, in multi-agent *M* contracts, tasks are exchanged between multiple agents. It is also shown that when used individually, or in pairs or threes, these contract types are insufficient for the maximization of global utility. However, when each individual contract type is applied simultaneously (called *OCSM-contract*) they:

- allow the algorithm to hill-climb from a task-allocation to any other task allocation with a single contract
- bring about the existence of a sequence path from an individually rational OCSM-contract to the optimal one.
- allow the algorithm to reach the optimal allocation in a finite number of contracts, for *any sequence of contracts*. This result means that i) no central processor is required to select the contract sequence and ii) agents can accept any OCSM-contract that is individually rational, and need not wait for more profitable contracts.
- the algorithm need not backtrack, since there are no local minima.

These properties are achieved because with OCSM-contracts there are no local minima, since the global optimum can be reached with a single contract.

The above contracting protocol has been extended to handle partial commitment contracts (Sandholm 1996, 2000). Informally, partial commitments represent tentative, as opposed to absolute, agreements to perform the agreed task(s) (see section 2.2.5). The contracting protocol described above consists of, like the CNP, a single round announcement, bid and award because all offers are fully binding. An iterative contracting protocol, called the leveled commitment protocol, is also presented. Under this protocol, commitments are not fixed and are themselves made a negotiation item. This new protocol allows unilateral decommitment at *any point in time*, as opposed to conditioning the contract on possible future events, as is done in contingency contracts (see section 2.2.5). Agents negotiate over decommitment penalties, one for each agent, and if an agent wants to decommit then it does so through the payment of the decommitment penalty specified in the agreed contract. It is also shown that selfish rational agents are reluctant to decommit because there are no incentives to do so. Therefore, it is to the best interest of even selfish agents to honour their commitments.

The computational boundedness of agents is given a treatment in the analysis and empirical evaluation of methods for decreasing the local computational costs. Sandholm identifies three categories of tradeoffs which in some contexts are guaranteed to reduce the cost of computation. Firstly, agents can tradeoff the complexity of marginal cost computations (discussed above) against the monetary risk. That is, agents can use different cost approximation schemes to make bids and awards while their previous bids are still pending. It is shown that some approximation schemes lower computational costs whereas others do not. Alternatively, agents can tradeoff obtaining more precise marginal cost estimations (and save computation) against being able to participate in multiple negotiations simultaneously. However, it is shown that this tradeoff only works in some contexts. Finally, agents can reduce their computational costs by trading off sending messages early on against waiting for more incoming offers.

3.2.4.1 Evaluation of the Contracting and Coalition Model of Negotiation

The contracting protocol presented by Sandholm computationally models the *process* of negotiation, rather than analyzing the optimal outcomes. This computational model thus supports the design and implementation of autonomous negotiating agents. The negotiation model differs along several dimensions from the one proposed by Rosenchein and Zlotkin in that their protocol resulted in negotiation reaching an outcome in a single round and assumed: i) agents were able to optimally compute their decision problems without any costs, ii) there were no side-payments, iii) negotiation was bi-lateral involving only two agents, and iv) the costs and capabilities of agents were symmetric. Sandholm's contracting protocol, on the other hand, is iterative and because of the complexities of the problems he assumes agents cannot compute their local optimization problem exactly. Furthermore, side-payments are allowed (through payment for task re-allocation). Negotiation is also extended from bi-lateral to multi-lateral interactions, in a market-like context, where agents buy and sell tasks from one another. Finally, different agents carry different costs and capabilities, therefore the symmetric assumption has been dropped.

However, the developed contracting protocol can only operate given an existing configuration of task allocations. Indeed, hill-climbing is the process of ascending some objective function *once a configuration of tasks has already been reached*. Thus, the contracting protocol of Sandholm can be used to *re-allocate* already existing task configurations but not to *allocate*, or configure, tasks in the first instance.

The CNP is further extended in the leveled commitment protocol by allowing iteration in interactions. In the problem domains of this thesis, it is this the iterative exchange of offers and counter-offers, due to informational uncertainty and the nature of preferences, that clearly mark interactions. Since iterations are both communicatively and computationally costly, then not only do agents need mechanisms to reason about the cost and benefit of continued negotiation, but the design of an interaction protocol must take this added complexity into consideration.

In addition to an interaction protocol, Sandholm provides a formal model of the decisions involved in agents announcing, bidding and awarding tasks. This extends the original CNP with a formal agent architecture. However, as was shown in some of the target problem domains of this thesis (section 1.4), negotiation decisions are richer than just considerations of costs alone. A richer agent architecture is required that formally accounts for more decision factors such as the time limits of negotiation (similar to the work of Kraus section 3.2.2) or the behaviour of the other agents (especially in environments where there are uncertainties in the what an agent knows about the other(s)).

The combination of contract types into OCSM-contracts (where agents can allocate tasks via combining the allocation of a single task, a set of tasks, swap tasks or share tasks with other agents), helps agents to escape local minima and reach the global optimum re-allocation in a number of steps. However, although tractable for small numbers of tasks and agents, the hill-climbing algorithm may take a large and impractical number of contract iterations for large number of tasks and agents. [19] Furthermore, although representing OCSM-contracts is tractable as the scale of the problem increases, the same is not true for searching a contract that increases the social welfare (Sandholm 1999). As the scale of the task set increases:

> ... the evaluation of just one contract requires each contract party to compute the cost of handling its current task and the tasks allocated to it via the contract. With such large problem instances, one cannot expect to reach the global optimum in practice. Instead, the contracting should occur as long as there is time, and then have a solution ready: the anytime character of this contracting scheme becomes more important (Sandholm 1999), p., 237.

The inability to escape local minima in negotiation is acknowledged in this thesis (detected as deadlocks in a contract's utility dynamics). However, in this thesis, agents negotiate over atomic services, or *O-contracts*. This is because of the agentification process that assigns services to agents (section 1.1). There may only be a single service provider for the types of problems considered in this research (e.g. $cost\_and\_design$ service provided by agent DD in the ADEPT scenario, section 1.4.1), excluding the possibility of *M-contracts*. Likewise, an agent may not be capable of performing another agents' tasks (e.g. a user agent, *IPCA* agent, cannot perform the tasks/services of a telecommunication service provider agent SPA, section 1.4.2). This excludes the possibility of swaps in contracts (*S-contracts*). Finally, since *each* service is usually performed by a unique agent, different tasks cannot be clustered and assigned to a single agent (excluding the possibility of *C-contracts*). For example, the service $Provide\_Customer\_Quote$ is performed by a single autonomous agent who is the only agent that has necessary domain expertise and resources to solve the problem(s). For these reasons, a decision mechanism is required that helps escape local minima in the task allocation algorithm. No analysis is provided as to the computational implications of the contracting protocol when the problem domain is scaled up, not in terms of the number of tasks, but in terms of the number

[19] Sandholm found that the TRACONET algorithm took *"multiple hours of negotiation on five Unix machines"* for a large-scale real world distributed vehicle routing problem (Sandholm 1993).

of issues involved in integrative bargaining (when agents negotiate not just over the price of a task/service but also its quality and delivery time). Multiple issue negotiation is an important feature of the types of problem domains of this thesis.

3.2.5 The Persuader System

The PERSUADER system was developed to model adversarial conflict resolution in the domain of labour relations which can be multi-agent, multi issue, and single or repeated negotiation encounters (Sycara 1987). The system uses both case-based reasoning (CBR) and multi-attribute utility theory (MAUT) for conflict resolution problems (Sycara 1987, 1989). PERSUADER is different to the CNP in that negotiation is modeled as an incremental modification of solution parts (rather than composition of partial solutions) through proposals and counter-proposals. The model, with its iterative nature, is used to narrow the difference between the parties involved, takes into consideration changing environments, and models social reasoning (by modeling other parties' beliefs) as well as belief modification of parties.

The system represents and reasons about three types of agents: a company, the union and the mediator. The latter agent's task is to engage in parallel negotiations with the parties when conflicts arise. Specifically, the mediator generates an initial compromise which both the union and the company evaluate from their own perspective. If the initial solution is acceptable to both parties then the process is terminated. Otherwise the mediator's task is transformed into considering whether to change the proposal or whether to attempt to change the belief of the disagreeing parties using persuasive argumentation (as defined in section 1.3.3).

In this context, negotiation is viewed as an iterative process since the parties entering negotiation have disparate goals. This "distance" in their goals is iteratively reduced to zero. To do this, agents must have the capacity to predict and evaluate whether new proposals do actually narrow the difference. Furthermore, agent communication is directed towards those parts of the proposal which are acceptable or unacceptable which implies that agents must be able to evaluate their plans and possibly modify or construct new ones based on this feedback. In addition to this deliberative component of negotiation, agents must also be reactive since the world changes constantly. That is, the expected goals and behaviours of other agents may change (through irrationality for example—note, the mechanisms are designed to handle irrational behaviour, unlike game theoretic models). Finally, since negotiation is viewed as a narrowing of the differences between goals *and* since agents are unwilling to give up their own goals, then they must be convinced to do so. Therefore negotiation requires persuasive argumentation.

3.2.5.1 Evaluation of the PERSUADER System

The PERSUADER system models *both* the iterative process of negotiation and the multi-issue nature of interactions. Therefore, these two features of the system capture some of the problem requirements of this thesis. However, mediation is unsuitable for the problem domains of this research since negotiation is a *mutual* selection of outcomes. Furthermore, in the problem domains of this research, it is not necessary for the agents to have similar beliefs at the end of negotiation. For example, inter-organizational agents may have diametrically opposed beliefs at the end of negotiation over the price of a service; the motivation of the *Vet_Customer* agents is to maximize price while the *Customer_Service_Division* agent seeks to minimize price and although they may settle on an agreed price, their goals have not changed. Therefore persuasion (operating over beliefs) is not a necessary condition for coordination in this problem domain.

In this and sections above, coordination models were presented that successively modeled the nature of interactions in open systems, where protocols of interactions are less normative and more descriptive and informal. The next three sections reviews other DAI models of negotiation that, although more descriptive in nature, have nonetheless design features that are relevant to the problem and the approach of this thesis.

3.2.6 Constraint Directed Negotiation

Constraint Directed Negotiation (CDN (Sathi & Fox 1989)) is an algorithm that belongs to the class of negotiation models that represent the decision making in negotiation as a constraint satisfaction process. It was developed by Sathi and Fox for the problem of resource re-allocation and is the precursor of the model presented in the next sub-section by Barbuceanu and Lo. Resource reallocation, or the adjustment of initial resources, is performed through the buying and selling of resources between agents. The authors have applied CDN to the real world problem of workstation requirements within an engineering organization (Sathi & Fox 1989). There, resources are workstations used by each group within the organization and as projects change so do the requirements of the groups. Therefore, the initial allocations of the resources have to be adjusted to reflect the new requirements.

The central concern of CDN is not so much with the communication protocol, but rather with the decisions, or the resolution mechanisms, involved that provide the content of communication. That is, “what is communicated about an agent’s bargaining position and how their positions are to be changed over time”. The mechanism the authors suggest is the constraint directed negotiation where the constraints represents agents’ objectives together with their utilities. Constraints are used for both offer *generation* and offer *evaluation*. At the conflict point the agents then negotiate either by modifying the current solutions or the constraints until a compromise is reached. “*Thus joint solutions are generated through a process of negotiation, which configures or reconfigures individual offerings*” (Sathi & Fox 1989), p. 166. The authors argue that because in the problem of resource reallocation there are many dependencies amongst the constraints of many agents (closely resembling the distributed planning problem), then a theory that only models how constraints affect individual offers, such as game theory, is inadequate (e.g. under market mechanisms an agent a sells resource S_1 to agent b for $\pounds 12$ and agent c sells to agent a resource S_3 for $\pounds 20$). What is required is a theory that can model multiple constraints that are conditional upon multiple offers (e.g. agent a offers resource S_1 to agent b *if* agent c allows access to a over resource S_3). The authors claim that in the latter case there is a need for more cooperative mediator-driven negotiation. They propose a distributed constraint mechanism to solve this type of cooperative problems.

In CDN the negotiation process is seen as a directed search in the problem space. The *problem state* is defined first, followed by an evaluation of states and finally generation of new solution states. Agents can make either simple transactions involving the selling and buying of a resource from one group to another. Alternatively, agents can make cascades involving open or closed chains of buying and selling between two or more groups. The *problem state* is then defined as a set of transactions and cascades that are formed by pairing buy and sell bids for resources.

These problem states are then *evaluated* using constraints. The authors elaborate on the contents of constraints, their classification hierarchy and a methodology for evaluating them. The *content* of constraints are represented as attributes of a resource, where each resource is described as a set of attribute-value pairs. The requirement(s) of agents then place restrictions on the attribute value. These restrictions are then classified into three constraint types (see (Sathi & Fox 1989), p. 169). The *evaluation* of offers involves firstly giving each constraint, or restriction on values, an importance and a utility function that represents preferences agents have about the offered transaction over the given constraint. Furthermore, the utilities are thresholded to represent minimal acceptability condition. Offered attribute values below this threshold are considered a violation of the constraint. Furthermore, in the resource reallocation problem most of the constraints are qualitative in nature. For example, an agent may own a Unix box and may require a Mac instead for a project. The agent may specify the buy and sell bids as conditional, one of their three classifications of constraint types, meaning that the Unix box is sold by the agent unless the agent receives a bid for the Mac. Therefore, the utility functions represent the ordinal preferences of the agents. Finally, an offer is evaluated over the total set of constraints by combining the individual utilities of all the sub-constraints. The combination policy they use are the elimination by aspects and lexicographic semi-order (Tversky 1969, Payne 1976, Svenson 1979, Johnson & Payne 1985). Agents then use these strategies to identify their favorite alternatives. The elimination by aspects combination strategy works by comparing the utility of each constraint with the corresponding utilities on other constraints. Offers with the lowest utilities on any constraint are eliminated from the consideration. This process continues until only one offer remains. This strategy is particularly well suited for qualitative constraints. Lexicographic semi-order is similar to elimination by aspects. However the method of elimination is different. It works by examining each constraint of an offer and eliminating those offers that have a lower value than a dominant alternative. The strategy is applied by using the elimination process operations on first the most important constraint, followed by less important constraints.

Given the overall utility of the offer, derived from using the elimination by aspects and lexicographic semi-order strategies, an agent evaluates the offer as: i) acceptable (the offer is above the threshold on all constraints but is not better (or what they call dominate) every other offer, ii) dominant (the offer is above the threshold on all constraints but *is* better (or what they call dominate) than every other offer or iii) unacceptable (the offer is below a threshold of at least one constrain).

Constraints are also used to generate *solution states*. Offers are generated via satisfaction and relaxation of constraints and are based on a set of qualitative operators which are motivated by human negotiation problem solving (Pruitt 1981). The operators, or search strategies, are:

- composition (bridging)—composition occurs when a new option is developed by combining together two existing alternatives which satisfy both parties' most important constraint. Sometimes in such cases both parties receive all they were seeking due to discovery of a good composition.

- reconfiguration (unlinking)—when good composition agreements are not available, one or both of the agents must make selective changes in their offer. As the authors state *"reconfiguration is the process of regrouping the bundle of negotiated goods"*. For example, consider an agent m who requires a Unix box running LaTeX Version 3.14159 (Web2C 7.3.1). Assume that agent n is offering a Unix box but with FrameMaker v.5.01 as the only word processing tool. The Unix box is therefore reconfigured by n to satisfy $m's$ requirements at a cost to n.

- relaxation (log-rolling)—is defined as when an agent ignores a specific constraint on an unacceptable alternative. For example, if negotiation involves five issues $\{i_1, i_2, i_3, i_4, i_5\}$ between two agents a and b, and if a values $\{i_4, i_5\}$ more than b who in turn values $\{i_1, i_2, i_3\}$ more, then a protocol can be agreed that a concedes on, and possibly violates the constraints of, $\{i_1, i_2, i_3\}$ and b concedes on $\{i_4, i_5\}$ where considerable benefits can be gained by both parties. As the authors state "relaxation provides an approximate technique for selecting transactions or cascades that perform the best on the most important constraints for each individual".

Typically, a good solution is found that maximizes the number of bids satisfied by composing and reconfiguring bids iteratively and not on simple pair-wise exchanges.

3.2.6.1 Evaluation of the Constraint Directed Negotiation

The CDN is novel in the manner it integrates informal models of negotiation, inspired by human negotiation problem solving, with AI techniques. The work presented in this thesis closely resembles the CDN in many respects. The conflict resolution mechanism of CDN is relevant to the problem domains of this thesis. The algorithm emphasize the importance of preferences of agents over multiple constraints, explicit representation of strategies as search operators, time deadlines and privacy of information in negotiation. For this reason the CDN shares many features with the developed coordination framework. The decision mechanisms of both systems are presented as evaluatory and offer generation processes.

However, both the evaluatory and the generation mechanisms of CDN do not model some of the requirements of the domains of this thesis. In CDN the constraints are qualitative in nature, whereas in the domains of this thesis constraints, represented as limitations on issue-value pairs that are exchanged between agents, can be both quantitative and qualitative. Therefore evaluatory utility functions are required that model preferences of agents for both types of constraints. The possibility of offers that contain both qualitative and quantitative issues means that an evaluation utility function is required that can represent the combined preferences of an agent over the constraints and is likely to include, because of the quantitative issues, arithmetic operations to consolidate the result of each individual utility, rather than elimination by aspect or lexicographic semi-order strategies. Furthermore, the accuracy of the two presented consolidation strategies is highly dependent on the distribution of the importances agents place on constraints; the further apart the importances of two agents on a constraint, then the combination of the two strategies is sufficient to identify the agreement set (Johnson & Payne 1985). Furthermore, sometimes it may be useful to model the preferences of agents as a whole for a set of offers. Complications with the two chosen strategies arise if such policies need to consolidate the preferences *across* agents (Johnson & Payne 1985). Simple quantitative additive models are better suited for such tasks (Corfman & Gupta 1993).

Furthermore, the CDN reconfiguration and relaxation search operators suit the problems of this domain. Reconfiguration, the process of regrouping the bundle of negotiated goods, is applicable when issues are added and removed during negotiation. Likewise, violation of constraints in order to search for other types of agreements is reflected in the need for agents to make trade-offs, lowering the acceptability constraint of one issue and simultaneously increasing the acceptance level of another issue. Composition, the search for alternatives by combining together two existing alternatives which satisfy both parties' most important constraint, is not a feature of the problem domains of this research because agents do not know, and are assumed to be unwilling to provide, constraint importance information to other agents necessary for composition. However, although relevant the authors do not provide any formal specification of the algorithms that implement these search operators. Therefore, not only are they inspired by informal theories, but they can not be operationalized due to a lack of any formal models. One of the contributions of this thesis is the formal modeling and empirical analysis of three algorithms that implement constrained search.

3.2.7 The Constraint Optimization and Conversational Exchange Negotiation Engine

Optimization methods, multi-attribute utility theory (MAUT (Keeney & Raiffa 1976, Luce & Raiffa 1957)) and conversational models are integrated into a single "Negotiation Engine" (NE) ((Barbuceanu & Lo 2000) [20]). The NE models more adequately, than the CDN, the multi-issue nature of negotiation, agents' goals

[20] This work complements both the CDN algorithm (section 3.2.6) and the PERSUADER (section 3.2.5) and was initiated from criticisms raised against the sole focus of auction technology on price of the commodity (Guttman & Maes 1998, Doorenbos, Etzioni,

and preferences, goal modification as well as the communicative elements of negotiation. In addition, the engine shares a common design philosophy as the coordination framework design of this thesis. It works by describing the local decision problems of agents as a multi-attribute decision problem and formulated as constrained optimization. Then, this constrained optimization problem solver is used by each local domain problem solver to find best solutions from their local perspectives. The best solution found is then communicated using a conversation interaction technology. If the received offer is not acceptable, then a constrained relaxation protocol is used to generate the next available best solution. Thus, the aim of the NE is to integrate both the local reasoner and the interaction system. The latter is part of the conversation technology that includes (Barbuceanu & Fox 1997): i) *conversation plans*, ii) *conversation rules*, iii) *actual conversations* and iv) *situation rules*. Conversation plans describe both how an agent *acts locally*, and, *interacts* with other agents by means of communication actions. *Conversation rules*, in turn, specify the permissible states (including the initial and final states) of the conversation plans. The execution state of the conversation is maintained in *actual conversations*. Finally, *situation rules* assist an agent with decisions about which conversations to instantiate. The *conversation plans*, *conversation rules* and the *actual conversations* conversation components of the NE can be used for normative communication models of the ACL component of the coordination framework (figure 1.1) of this research. *Situation rules* are similar to the *service description language* (SDL) developed in the ADEPT project for specifying the local service execution plans of each agent (Jennings *et al.* 2000a). Further similarities lie with the design philosophy of the conversation technology. It can be used for not only representing and executing a structured patterns of agent interaction, but also as a “scripting language”. The NE provides API-s, using its conversational and reasoning language (described below), for the local reasoner to construct both models of the situation and goals and reason about interactions with other agents. These API-s can be seen as interfaces between the wrapper and the local problem solver and the agent and the ACL in figure 1.1. For example, the local problem solver can interact with the wrapper using the service description language and the wrapper interacts with other agents via the ACL interface.

The MAUT and constraint optimization elements of the reasoning component of the NE are discussed next. In NE an agent behaves to achieve its goals. Goals can be: either composed (containing other (sub) goals) or atomic (immediately executable); either *controllable* (goal is under the control of the agent) or *non-controllable* (part of the agent’s plan, but agent has to obtain the commitment of the agent controlling these goals for their achievement); either “on” (is achieved) or “off” (is not achieved). Agents then attach preferences, or utilities, towards the achievement or non-achievement of these goals. Thus agents are utility maximizers. Utilities not only model the preference of an agent, but also, as the authors claim, quantify the influences between agents where the utility of non-controllable goal describes in “some way” the power

& Weld 1997).

that the other agent has on the agent needing the goal. The final element of the language of the decision model is the agent *roles*. Roles describe the goals an agent controls and the goals it needs and function to form a strategic coalition formation, choosing who to involve when needing to achieve a certain task.

Given this language of decision making (goals, utilities and roles), the decision problem of an agent, P, is formulated as a constraint optimization problem:

$$P = \langle G, C, U, \mathit{criterion} \rangle$$

where $G = \{g_i, \ldots, g_n\}$ is a network of n goals, $C = \{c_i, \ldots, c_n\}$ is a set of constraints of the form $on(g_j)$,$off(g_k)$ or an implication on both sides of which there are conjunctions of on-off constraints, $U = \{(g_1, U_{on}(g_1), U_{off}(g_1), \ldots, (g_l, U_{on}(g_l), U_{off}(g_l)\}$ is a utility list consisting of a set of goals with either associated on/off utilities and *criterion* $\in \{max, min\}$ which is either a maximization or minimization optimization criterion. The overall utility of the labeled goal network G, $Util(L, G)$, is computed as the sum of the "on" labeled goals, plus sum of the "off" labeled goals (called the additive scoring model (Keeney & Raiffa 1976)), where $L : G \rightarrow \{on, \mathit{off}\}$ is a function that maps each goal in the goal network with either an "on" or "off" label . Thus, solution P is a labeling L such that $Util(L, G)$ is either maximized or minimized, according to the criterion.

The authors then show that the above labeling problem P that maximized (minimizes) utility is equivalent to the satisfiability (MAXSAT) problem in optimization (Barbuceanu & Lo 2000), p.241. Two optimization algorithms are provided within the NE that operate over the same representation of the goal network to solve this optimization problem of an agent. One is a stochastic search based algorithm that is incomplete and not guaranteed to find a solution, but performs well on large scale problems both in terms of time and its ability to actually find a solution (Selman, Levesque, & Mitchell 1992, Jiang, Kautz, & Selman 1995) and another the branch and bound search algorithm that is complete and guaranteed to find the optimal solution (Mitten 1970). The latter algorithm operates by maintaining the utility of the current best solution. If the utility of another explored partial solution does not exceed the utility of the current best solution then that partial solution is dropped and another partial solution is explored. The decision mechanism supplied in the NE also allows for the integration of the two algorithms, using, for example, a random search first for a number of runs and then using the best solution from the random search as a bound constraining the branch and bound algorithm to find a better solution.

The reasoning procedures are extended by a multi-attribute utility theoretic language within the NE that support optimization of search for utilities over multiple issues. Specifically, agents share a set of negotiation issues, or what the authors call the *attributes* of negotiation, defined as $A = \{a_1, \ldots, a_n\}$. The domain of an attribute a_i, D_{a_i}, is an interval $[l, h]$, where l and h are integers or reals, describing the range of values that the attribute can take. Agents then interact by exchanging multi-attribute specifications. The assumption made is that the agents share *both* the attribute list *and* the domain of each attribute. Furthermore,

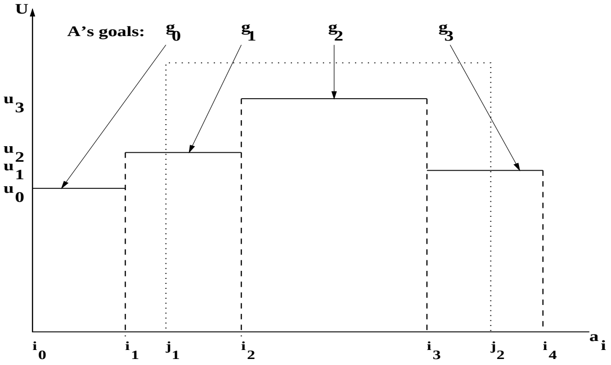

Figure 3.5: Exemplar Utility of an Attribute a_i.

for each attribute a_i there is a utility function $U_{a_i} : D_{a_i} \rightarrow [0, 1]$ and agents have opposing interests over each issue, expressed as different directions over the maximization U_{a_i} for each agent. Another important assumption the authors make is that utility function of an agent has the form shown in figure 3.5, where the domain of the attributes can be decomposed into a set of disjoint sub-intervals that cover the entire domain, such that on each sub-interval the utility is constant. Figure 3.5 shows an example of an attribute whose domain values between $i0$ and $i1$, for example, have the same utility to the agent (represented as the horizontal utility line). It follows that fewer sub-intervals, at the extreme where there is only one sub-interval corresponding to equal utility across all domain values of the attribute (the agent values all solutions of the issue equally), then the easier the resolution of that issue. Then for each sub-interval $[i_l, i_{l+1}]$ an atomic goal $g^l_{a_i}$ is created which is *on* iff the value of a_i is in the interval $[i_1, i_{l+1})$. Furthermore, the authors assume that although agents have different valuations over different ranges of an attribute's domain, they nonetheless have further acceptability constraints about what attribute values are acceptable (thresholded utilities of CDN perform the same function 3.2.6). For example, in figure 3.5 only values between $[i_3, i_4]$ may be acceptable to an agent. Given the above a MAUT problem of the NE is then the assignment of *on-off* labels to the goals of the problem that satisfy the limits of all of the attributes' domains as well as their acceptability constraints. This solution the authors call the *deal*. Also, an optimal solution is one that has maximum utility for the agent. A deal acceptable to *both* agents is one where for each attribute a_i the acceptable set of values for the two agents have a non-empty intersection. An example of such a deal is shown in figure 3.5. Assume there are two agents A and B. Further assume that figure 3.5 shows the utility for values of attribute a_i for agent A. Now assume that $[i_3, i_4]$ is the set of acceptable values for A (this utility is the result of A having goal g^3—$on(g^3)$) and $[j_1, j_2]$ is the set of acceptable values for B. Then $[i_3, j_2]$ is the non-empty intersection for attribute a_i. This intersection solution represents a possible agreement between the agents, because each solution contains ranges of values acceptable to each agent.

The sequences of local decision making and communication of the offers are as follows. At the first time step each agent represents its problem as a MAUT problem, defining attributes, goals, constraints and utilities. Then each agent specifies its acceptable solution which defines the interval of acceptable values for each of the issues. After defining the problem the first best solution is computed by solving the initial problem using the branch and bound algorithm. The branch and bound algorithm can support a concession protocol by searching for lower utility solutions. Lower utility solutions are generated by over-constraining the problem, achieved by negating the previous best solution and then adding this new constraint to the goal network. The best solution is communicated to the other agent at the end of each iteration of the algorithm. If the proposed solution is acceptable to the other agent then the process terminates successfully. Alternatively, the other agent may communicate the fact that it can not find any more new solutions to the part. When both of the agents can not search for any new solutions, then negotiation terminates unsuccessfully. Otherwise, the agent that can generate new solutions continues to generate and propose them. Finally, the offered deal by the other agent is checked for intersection with the agent's own last offer. Negotiation terminates successfully if such an intersection exists, otherwise the agent searches for other solutions to propose and the process continues. The process ends when a mutually acceptable deal has been found or else no more solutions exists.

3.2.7.1 Evaluation of the Constraint Optimization and Conversational Exchange Negotiation Engine

The NE is closely related to the work reported here and models many of the features and requirements of the problem domains of this thesis. It models both the communication aspects of interaction (through the conversation technology) and complex local decision mechanisms, and a formal goal network representation language, that account for some of the requirements of this thesis such as: i) multiple issues ii) constraints of agents over these issues iii) conflicting preferences of agents and iv) a concession protocol that is guaranteed to find a solution if one exists. Furthermore, this protocol is interleaved with a stochastic search algorithm that is scalable to large problems and assists the concession protocol with new search locations. These two search protocols, as well as their combination, represents two strategies agents can use to reach agreements.

However, the concession protocol is guaranteed to find a solution because of the assumption that the agents share the same domain specification of the attribute (or issue interval). Given that the interval value of agents are exactly the same, and it is only the acceptability constraints that differ, then it naturally follows that a solution *must* exist. Although this assumption is useful for system analysis, an approach also adopted in the evaluation phase of this thesis, it is nonetheless a strong assumption that is not applicable to the type of problem domains of this thesis. Agents do not necessarily share the same intervals over each and all of the issues in negotiation. Indeed, negotiation *can* fail when there exists no such intersection.

Furthermore, no formal model of how utility theory is used to model power of agents or how roles can be used to form strategic coalitions. In addition to this, and more importantly, it is not clear, and the authors do not make any reference to the fact, that the developed negotiation, like the CDN above, protocol models interactions amongst cooperative agents only. This can be seen in the conversational protocol described above where agents truthfully reveal that they can not generate any more solutions. The assumption of truthful revelation is strong especially among open system agents that may be selfish and have incentives to lie about their negotiation positions in order to maximize their own welfare.

3.2.8 Multi-dimensional Service Negotiation as an English Auction

Vulcan and Jennings have applied the principles of mechanism design to model (as an English auction, see (Binmore 1992)), the one-to-many service negotiation between the CSD and the VC agents for the $Vet\_Customer$ service in the ADEPT scenario (Vulkan & Jennings 1998). The English auction has been modified to handle service negotiation over multi-dimensional private value objects.[21] Services are described by the tuple $(p, \overline{s})$, where p is the price of the service and $\overline{s}$ are the additional issues of a service. A service buyer's preferences are then defined by a linear utility function $u_b(\overline{s}) - p$, that increases with increasing quality of the service. p is the price of the service and is restricted to a maximum value. A service seller's preferences, on the other hand, are defined by the cost function $c_s(\overline{s}) + p$. The preferences of the buyers and sellers of services are also conflicting, meaning that the preferences of both agents over each issue, move in the opposite directions.

In addition to a service client (or what they call a service seeker) initiating the auction, the authors propose a pre-auction protocol (as well as incentive conditions and the required auction knowledge for an agent to initiate an auction) where the service providers can hold an auction amongst themselves. The winner of the auction then approaches the service-seeker with a "take-it-or-leave-it" offer. Analysis is provided, in terms of dominant strategies [22] that result in outcomes that are efficient (increase the sum of the individual utilities and are fast). Agents then need store, as knowledge, only these dominant strategies (hence, individually rational) of the resulting protocol.

However, modeling a part of the ADEPT business process as an English auction has a number of limitations. Firstly, an English auction models one-to-many interactions, where a single auctioneer (or a service buyer here) interacts with a number of buyers (or a service seller).[23] Because it is an open-cry auction, all the valuations of agents are publicly "heard". This may be regarded as undesirable by, for instance, a Vet Customer organization who does not, for competitative reasons, want to reveal its valuation

[21] A private value object is an object, or a service in this case, whose worth depends solely on an agent's own preferences. See (Binmore 1992) for an explanation of other value type auctions.

[22] A dominant strategy is a strategy that yields an expected payoff which is higher than other strategies *whatever* the behaviour of other agents and the state of the world. Note that using dominant strategies eliminates the need for agents to condition their strategies on beliefs.

[23] Note that the principles and results of mechanism design still apply in-spite of the reversal of labels.

to other Vet Customer service providers. Instead it may prefer to enter a more “private” dialogue in the form of one-to-one negotiations. The public revelation of valuations in an English auction also leads to possibility of collusions between auction buyers, resulting in lower revenue for the auctioneer (Rasmusen 1989). The example below from (Sandholm 1996) and (Rasmusen 1989) illustrates these collusion possibilities. Let buyer agent i have a valuation of 20 and all the rest of buyers have a valuation of 18 for the service on offer. Further assume that the bidders collude by agreeing that i will bid 6 and all the rest bid 5. If one of the other buyers exceeds 5 then i can observe this and will go as high as 20 and the cheater will not gain anything from breaking the coalition. Therefore, collusions are self enforcing in an English auction. Although collusions in an open environment are technically difficult to electronically implement, since agents will have to identify one another and agree to form a coalition, they are nonetheless possible and hard to detect electronically. This is especially true in virtual worlds where it is relatively inexpensive to create virtual identities. Furthermore, the auctioneer itself can profit from collusions, by placing agents representing it in the auction, who then stimulate the market by unfairly raising the bids.

In spite of these technical difficulties electronic auction houses have provided the technological foundations of the recent rise in electronic commerce for business-to-business, business-to-customers and customer-to-customer applications (eBay, AuctionBot, i2, Amazon, FishMarket). However, auctions, although popular, are also qualitatively problematic. Technically an auction is profitable for the auctioneer in the *short term* because of the *winner’s curse* (Binmore 1992) which is where the winning bid for a good occurs *above* the good’s market price. Therefore, in the *long term* a buyer is likely to be dissatisfied with paying for a good above its market valuation. This is more likely to be true for business-to-customer or business-to-business types of electronic commerce applications (Guttman & Maes 1998). Furthermore, some auctions (such as the English auction) may require a critical number of bidders before they can commence. Coupled with the communication latencies involved, bidders, or agents representing them, may have to make bids over several days. This problem is exacerbated when a buyer’s bid is not the winning bid, requiring the bidder to restart the whole process of bidding once again. Apart from technical limitations, auctions “pit” the buyer against the seller and they tend to focus solely on the price of a good. Auctions are generally viewed as hostile exchange environment, where the buyers are pitted against the sellers, where neither party considers the long term relationships and the benefits that may actually increase profit for both. This type of relationship is more likely to occur between businesses and their customers or other businesses. Paying exclusive attention to price also hides from the consumer important information about the added value of a good by a seller, resulting in an undifferentiated and homogeneous representation of sellers (Guttman & Maes 1998, Doorenbos, Etzioni, & Weld 1997).

3.2.9 Kasbah Electronic Agent Marketplace

For some of the reasons above, negotiation technology has been proposed as an alternative solution to auctions as the next generation of e-commerce products (Guttman & Maes 1998). Below, one representative e-commerce negotiation solution, called Kasbah (Chavez & Maes 1996), is briefly reviewed. Kasbah depart from normative game theoretic approaches to negotiation, hence is less formal, sometimes heuristic, ad hoc and are user, as opposed to protocol, centered.

Kasbah is a multi agent system *application* for electronic commerce (Chavez & Maes 1996). It is an electronic agent marketplace where agents negotiate to buy and sell goods and services on behalf of the user. The motivation behind Kasbah is to assist users in electronic shopping:

> by providing agents which can autonomously negotiate and make the "best possible deal" on the user's behalf (Chavez & Maes 1996).

The system itself is a hosted web site where users visit to buy and sell goods. Users create buying or selling agents which interact in a marketplace. The marketplace itself provides a common language for the agents as well as a yellow pages service. The agents are simple, in that *"they do not use any AI or machine learning techniques"*, share no common goal, have diametrically opposite aims and are autonomous (Chavez & Maes 1996). However, motivated by acceptance by the user, the system is designed to allow the user to have a certain degree of control over the agents. The selling user, for example, can define the goal of the agent by specifying: i) the desired date to sell the item by, ii) the desired price, and iii) the lowest acceptable price. The reverse is true for the buyer. These parameters define an agent's goal and the achievement of this goal is modeled heuristically as the strategy to begin offering the item at the desired price and if it is not accepted then the selling agent lowers the price. The price is iteratively reduced with the constraint that the price is at the lowest acceptable price when the desired date is reached [24]. How the agent decreases, or increases in the case of a buyer, its offer is modeled as one of linear, quadratic or cubic decay functions.

Kasbah addresses some of the issues mentioned in chapter two and is an attempt to actually engineer a real world application. The system models time, actions and strategies involved in negotiation. However, the system fails to properly address the issues of commitments and uncertainty mentioned in the previous chapter. The bounded nature of agents is omitted from the model by developing very simple agents, which incur minimal computational costs. The majority of the computationally demanding tasks are not delegated to the agent, but rather remain at the user level. Therefore the agents are only semi-autonomous, since Kasbah only models a subset of the decision making which is involved in negotiation—the user makes the other decisions. Furthermore, the decisions that are delegated to the agents (called strategies in Kasbah) are severely limited to only three and even their selection is not autonomous, but again, is made by the user.

[24] The reverse is true for the buyer agent

Also no formal account or analysis is given of what exactly is the “best possible deal” or the likelihood of outcomes given strategies of agents.

The problem of introducing multiple issues into a negotiation is also not addressed in Kasbah. Scaling the problem to multi-dimensional scales influences not only the computational complexities of the search for solutions, but also raises the problem of the representation of preferences. Negotiation search algorithms are needed whose domain is constrained by the specification of user’s preferences over a multi-dimensional space. These constraints can be restrictions over the content or the process of negotiation. Content constraints specify preferences over the types of outcomes preferred by a user. These constraints can either be hard constraints, such as *“I am willing to pay between £*20 *and £*40 *for a service”* or soft constraints, such as *“quality of a service is more important than its price”*. Therefore, in multi-issue negotiation a more sophisticated methodology is required to capture and represent user’s preferences, which are ultimately delegated to the agents who interact with one another on behalf of the user. Constraints on the process of negotiation, on the other hand, specify the preference of a user about the style of negotiation such as the concession rate. Kasbah agents can only concede on offers. With multi-issue negotiation agents can also spend time searching for win-win outcomes. Therefore the agent, or the user, has more choices of behaviours when multi-issues are considered. Furthermore, in Kasbah the user makes the choice of concession rate. This contrasts with the prescriptive game theoretic models of negotiation where the decision making of the agents are normatively bounded to rational choices that are known to be optimal decisions. Kasbah belongs more to the descriptive models of choice whose aim is to describe how individual actually do, rather than should, behave. These models range from behavioural negotiation heuristics (Pruitt 1981, Fischer & Ury 1981, Kraus & Lehmann 1995) that provide guidelines for negotiation decision making, to models that describe decisions as evolving in response to the negotiation environment (Binmore 1990, Matos, Sierra, & Jennings 1998, Oliver 1994).

3.3 Assessment of Related Work

Features of the *computational* models covered in this chapter are summarized in figure 3.6 along some of the important dimensions identified in the previous chapter.

It can be seen from the table that the problem of bi-lateral negotation has recieved little attention from the computational community. Furthermore, little or no work has addressed the problem of repeated protocols, reasoning about uncertainties or commitments during negotiation.

The protocol of this thesis needs to be designed for highly structured interactions between two agents only. Therefore, game theoretic models are an appropriate candidate for the problem of coordination. These models are not only analytically useful, but they also have several desirable properties. However, there are a number of criticisms of these models with regards to the requirements of the target domains of this research

	Domain Theory	*Kraus*	*CNP*	*Sandholm*	*Persuader*	*CDN*	*NE*	*Multi-issue Auction*	*Kasbah*	*Wrapper*
Number of Agents	N	N	N	N	N	N	N	N	2	2
Symmetric capabilities	Yes	No	No	Yes	No	No	No	No	No	No
Cooperative(C) / Selfish(S)	C & S	S	C	S & C	S & C	C	C	S	S	C & S
Protocol	one-shot	one-shot	one-shot	iterative	iterative	iterative	iterative	one-off & iterative	iterative	iterative
Encounters	one-off	one-off	one-off	one-off	one-off	one-off	one-off	one-off	one-off	one-off
Number of Issues	1	1	1	1	N	N	N	N	1	N
Commitment	No	No	No	Yes	No	No	No	No	No	No
Uncertainty	No	Yes	No	Yes	No	No	No	No	No	Yes
Time Limits	No	Yes	No	No	Yes	Yes	No	No	Yes	Yes
Agents Bounded	No	No	No	Yes	No	No	No	No	No	Yes

Figure 3.6: Comparison Matrix of Computational Models of Negotiation

(section 3.1.9). In addition to these criticisms, the operational mapping of game theory models into DAI environments introduces further difficulties. As Kraus notes, in order to apply these models a designer must (Kraus 1997b):

- choose a strategic bargaining model
- map the application problem to the chosen model's nomenclature
- identify equilibrium strategies
- develop simple search techniques for appropriate strategies
- provide utility functions

Although choosing a strategic bargaining model and mapping it to an application may not be too difficult, game theory requires that all the agreements be known in advance before equilibrium strategies can be proven. The theory's basic assumptions also mean that most game theoretic models do not consider the computational and communication complexities which are important in practical applications. Furthermore, multiple issues are not adequately represented in game theoretic models.

Informal models such as CDN and Kasbah, on the other hand, are beneficial in that there is no need to build models of interactions from scratch—there already exists a large body of research which has developed over a number of years in other disciplines such as behavioral and social sciences. However, informal models have a different set of limitations. Again, as Kraus notes, applying informal models to DAI problems can be done in two ways (Kraus 1997b)

- develop heuristics for cooperation based on informal models (e.g. (Kraus & Lehmann 1995)) or
- apply informal models to DAI problems after formalizing the models (for example through logics (Kraus, Nirkhe, & Sycara 1998))

However, there is a need for evaluation techniques such as simulations or empirical analysis in both cases above since informal models do not formally analyze the behaviour of the system (unlike game theoretic models).

The aim of this chapter has been to show that the general requirements of the target domains together with the need for developing a flexible decision mechanism have meant that the negotiation wrapper cannot be adequately modeled using normative game theoretic models. Instead, these requirements have meant adopting a more descriptive approach that provide decision heuristics. However, an agent is cast (having preferences) and described (a utility maximizer) and analyzed (in terms of Nash, pareto-optimality and reference solutions) in the nomenoculture of game theory, but their decision making are based on informal and descriptive models. Therefore, because of the limitations of informal models mentioned above, the developed model is empirically evaluated to discover properties of the wrapper (see chapter 5).

When viewed operationally the developed coordination *framework* (the protocol, services and the reasoning models, see figure 1.1) is normative in that the agent is required to adopt the protocol of interaction specified by the communication language, but is free to adopt any decision strategy (or any implementation of the wrapper) to execute within the protocol. This means that a game theoretic agent can interact with a heuristic rule based agent using the framework. They differ in what decision schemes they use to implement the negotiation wrapper. However, for evaluation purposes a descriptive approach is adopted, where the interaction protocol and a set of strategies is imposed on the agent.

Chapter 4

A Service-Oriented Negotiation Model

A formal account of the developed coordination framework is the subject of this chapter. This formalization specifies two protocols of interaction (section 4.1) and three negotiation decision making mechanisms (section 4.2). This formalization is intended to model the important issues identified in chapter two and addresses the criticisms of the related approaches described in chapter three. The context in which the service-oriented negotiations take place has already been described in the first chapter (section 1.4).

4.1 Interaction Protocols

A protocol of interaction is required because sub-problems interact during domain problem solving and agents therefore have to communicate and interact (section 1.3). A protocol of interaction can also reduce the uncertainties involved in strategic interactions (section 2.2.6.3). Thus protocols of interaction assist agents in their problem solving. The computations involved in such problem solving can usefully be categorized into on-line and off-line. Off-line computations are the processes involved in the local deliberation phase of what to offer and are presented in section 4.3. On-line computations, on the other hand, are the processes involved in the communication of the deliberated offer itself. The on-line computations, as well as the knowledge required for computation, are discussed in this section. There are two protocols: one for negotiation proper and the other for issue manipulation. Two protocols are needed because the language and rules of interactions differ when agents are exchanging contracts during negotiation to when they communicate about which issues should be included or retracted from the current set of issues in negotiation. The negotiation protocol is described first.

4.1.1 The Negotiation Protocol

The design of the protocol of interaction has been motivated by the normative models of coordination such as game theory (see chapter 3).[1] Agents' interactions are constrained by the rules of a normative structure which specifies their interactions independently of their roles. The interaction is modeled as an alternating

[1] Note that a *norm* refers to prescriptive rules of the game (in the game theoretic sense).

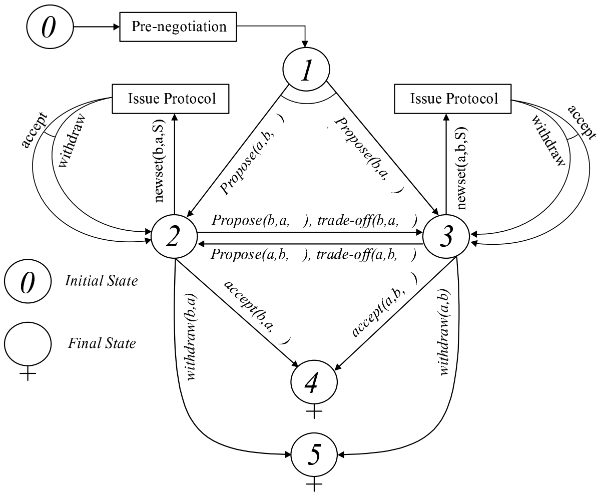

Figure 4.1: The Negotiation Protocol.

sequence of offers and counter-offers (sections 3.1.7, 3.2.3) which terminates with either a commitment by both parties to a mutually agreed solution or else terminates unsuccessfully. The protocol (figure 4.1) starts with a dialogue to establish the conditions for the negotiation. These conditions are mutually satisfied in this pre-negotiation phase and must specify the set of initial issues (see section 2.2.2) as well as a shared meaning of the not only these issues but also the meaning of the conversation terms of the ensuing interaction (the primitives shown in figures 4.1 and 4.2). Additionally, during this pre-negotiation phase agents may agree on which role will begin the negotiation, acting as the initiator of the protocol, while the other role becomes the responder to the role who initiated the negotiation.[2]

Then, the agent who was selected to make the first offer proceeds to make the first offer for contract ϕ (transition from state 1 to state 2 or 3—the or transition, representing who starts the negotiation, is shown in figure 4.1 as the arc joining the two possible proposals given state 1). Note the protocol is for integrative negotiation, where agents negotiate over packages, rather than individual issues. After that, the responding agent can make a counter-offer (see section 4.4) or a trade-off (see section 4.5.2) (moving to state 2 or 3 depending on who was the initiator). The initiating agent can in turn make a new counter-offer or a new trade-off (going back to state 2 or 3). Since the information models of the other agent(s) (such as the reservation values, the weights and preferences of each issue) are not publicly known, offers may be outside the mutual zone of agreement. Additionally, even though offers may be within a zone of agreement, they may nonetheless fail to meet the current aspirational demand of the other agent. For example, if an (seller)

[2] In cases of no agreement the conflict may be resolved through randomly selecting which which role will be the initiator and which role will be the responder.

agent is demanding a price over $\pounds 20$ for a service, with price reservation values of $[10, 30]$, and the other (buyer) agent has offered $\pounds 15$, then although the offer is within the reservation values of the seller, the buyer's offer fails to meet the *current* aspirational needs of the seller. Therefore, agents may iterate between states 2 and 3, taking turns to offer new contracts. In either of these two states, one of the agents may accept the last offer made by the opponent (moving to state 4) or withdraw from the negotiation (moving to state 5). Agents always withdraw from the negotiation process when the negotiation deadline has been reached.

While in state 2 or 3, agents may start an ellucidatory dialogue to establish a new set of issues to negotiate over (see section 5.2.3 for more details). This transition to the issue manipulation sub-protocol is represented in figure 4.1 by the primitive *newset* to the issue protocol. The execution of the negotiation protocol is resumed when either the agents have agreed to a new set of issues (represented in figure 4.1 by the accept primitive from the issue sub-protocol back to the negotiation protocol where negotiation resumes with a new set of issue) or else when then agents have failed to come to an agreement over a new set of issues (represented in figure 4.1 by the withdraw primitive from the issue sub-protocol back to the negotiation protocol where negotiation resumes with the same issue set as before).

This negotiation protocol is a natural extension of the contract net protocol (section 3.2.3) permitting iterated offer and counter-offer generation and permitting the modification of the set of issues under negotiation. The presence of time deadlines guarantees termination of the protocol.

4.1.2 Issue Protocol

As mentioned in section 2.2.2, agents may not share the same goal set at the outset of negotiation. Alternatively, agents may identify an issue that they both agree on in the course of negotiation. Conversely, there may be a need to introduce a new issue(s). Therefore, there is a need for a protocol that normatively specifies how the set of issues in negotiation can be amended.

The protocol for establishing a new set of negotiating issues (figure 4.2) is isomorphic to the negotiation protocol described in figure 4.1, with the exception that the meaning of the primitives and the content of this protocol (a new set of issues) are different to the negotiation protocol of figure 4.1. Additionally, the choice of the initiator of this sub-protocol is strategically determined by the agent who wishes to initiate this sub-protocol while executing the negotiation protocol. The pre-negotiation phase is omitted (since the current set of issues has already been agreed). The object of negotiation, contract ϕ, is replaced by a new set of issues S, and primitives propose and trade-off are replaced by *newset*—a request for a new set of issues to be included in to the negotiation. Each negotiating agent can start a dialogue over a new set of issues S where the numbers reflect the same state as the main negotiation protocol. Thus, if agent a starts the issue manipulation dialogue with the utterance $newset(a, b, S)$ while in state 3 in the negotiation protocol, in figure 4.1, then this results in the transition from state 3 to state 2 in the sub issue-manipulation protocol in figure 4.2). Each agent can then either propose a new set (transition from state 3 to 2, or 2 to 3, depending on who started the dialogue), accept the other's proposed set (state 4) or withdraw and continue with the original set (state 5). An agent's strategical choice of the protocol usage is captured in the wrapper deliberation architecture (section 4.3). However, before the deliberation architecture is formally specified, first the meaning and rules of communication are informally presented in the next section, followed by the basic building blocks of the formalization (section 4.2).

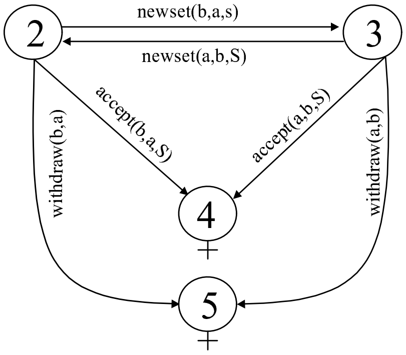

Figure 4.2: The Issue Manipulation Protocol.

4.1.2.1 Normative Rules of the Protocol

Communication among the agents using the protocol follows a set of normative rules represented as simple if-then rules. The content of the messages used in the agent communication language (ACL, shown in figure 1.1, section 1.2) is shown in figure 4.3 and consists of: one of a limited number of primitive message types, the identity of the sender and the recipient (both agent identifiers), and the service concerned through the set of negotiation issues that describe the terms and conditions of service production and consumption. Additional information, and not shown in figure 4.3, may be included (such as the message number) that facilitates conversation management. However, figure 4.3 depicts the main requirements of the communication protocol.

The first three primitives (*can-do, not-capable* and *capable*) are used in the pre-negotiation state of the protocol. They provide "connection" capabilities, functioning to initiate negotiation for a service that is actually provided by a seller and is required by a buyer. Note that in this research the performative *can-do* means *capable of* as opposed to *it is permitted to*.

The agents then enter negotiation proper and use the remaining communication primitives to provision services. The next four primitives are messages that agents utter when using the negotiation protocol de-

Action	*Content*	*Semantics*	*Context*
can-do (a,b,s)	Empty	Sender a asks if the recipient b is, in principle, able to provide the service s.	Message can be sent by any agent at pre-negotiation phase.
not-capable (a,b,s)	Empty.	a informs b that it is not capable of s.	Used by a only in response to a can-do action at pre-negotiation phase.
capable (a,b,s)	Empty.	a informs b that, in principle, it is capable of s.	Used by a only in response to a can-do action at pre-negotiation phase.
propose (a,b,ϕ)	A single contract information object.	a proposes to b that b performs service under the conditions specified in contract by ϕ and communicates that the contract under the terms of ϕ has a lower utility to a than the previous offered contract under previous contract terms.	Used only in response to either an action of type propose or trade-off or start of initial state
trade-off (a,b,ϕ)	A single contract information object.	a proposes to b that b performs service under the conditions ϕ described in the contract that is on the table and communicates that the contract under the terms ϕ has equal utility to a than the previous offered contract with the terms.	Used only in response to either an action of type propose or trade-off.
accept (a,b,ϕ)	Empty.	a accepts to performing the service under the contract ϕ that is on the table.	Used only in response to either an action of type propose or trade-off.
withdraw (a,b)	Empty.	a wishes to terminate the negotiation.	Used only in response to either an action of type propose or trade-off.
newset (a,b,$\mathcal{S}$)	A single contract information object.	a proposes to b that b performs service under a new set of conditions $\mathcal{S}$.	Used only in response to either an action of type propose, trade-off or newset
accept (a,b,$\mathcal{S}$)	Empty.	a accepts negotiation dialogue over the service under a new contract conditions $\mathcal{S}$.	Used only in response to an action of type newset.
withdraw (a,b)	Empty.	a rejects a newset of service conditions and resumes negotiation dialogue over the service under the old contract conditions.	Used only in response to an action of type newset.

Figure 4.3: The Communicative Rules

scribed in figure 4.1, and the last three are messages belonging to the issue manipulation protocol described in figure 4.2. The meaning of each of these primitives is described in the column entitled semantics. Rules, in turn, represented as contexts in figure 4.3, specify the usage of the above primitives which all agents must adhere to during negotiation.

The building blocks of the formalization are introduced next.

4.2 A Bilateral Negotiation Model

This section presents the developed model for representing agents' knowledge about services. This model includes: i) the set of negotiation issues, their reservation values and importances as well as the domain problem solver's preferences over each issue (section 4.2.1), ii) the roles agents can adopt in service-oriented negotiation (section 4.2.2), and iii) the thread of offers and counter-offers exchanged in negotiation (section 4.2.3). The role of this model is to support the decision making functionalities of the wrapper during multi-attribute bilateral negotiation.

4.2.1 Issues, Reservations, Weights and Scores

The aim of this section is to formally represent issues. This representation will serve as a data structure during the negotiation process. An informal example of multi-issue negotiation is presented first, followed by a formal treatment.

The object about which agents negotiate is referred to as a *contract* (ϕ in figure 4.1). Contracts represent the bid (or offer) on the table during negotiation and the final contract at the end of a successful negotiation. The contract structure is derived almost exactly from the types of legal contract which are often used to regulate current business transactions (Jennings *et al.* 2000a).

Figure 4.4 is a sample contract from the BT business process management domain (section 1.4.1). The contract contains both an identification and a negotiation part. The identification part is shown in figure 4.4 by the slots *service_name*, *contract_id*, *Server_agent* and *client_agent*. These features function to uniquely identify the contract under negotiation. The negotiation part is represented by the remaining slots and describe the actual issues agents negotiate over. Note that the any ambiguity over both the meaning and the value of both the identification and the negotiation issues is assumed to have been resolved at pre-negotiation phase of interaction. For example, it is assumed that both agents know the meaning of the contract attribute *price* and also have a common value (dollars for example). The attributes of this sample contract are described next.

The *service_name* is the service to which the agreement refers and *contract_id* is the contract's unique identifier (covering the case where there are multiple agreements for the same service). *Server_agent* and *client_agent* represent the agents who are party to the agreement. *Delivery_type* identifies the way in which the service is to be provisioned—services can be provisioned in two different modes depending on the client agent's intended pattern of usage and the server agent's scheduling capabilities: (i) one-off: the service is provisioned each and every time it is needed and the agreement covers precisely one invocation; (ii) on-demand: the service can be invoked by the client on an as-needed basis within a given time frame (subject to some maximum volume measurement). The contract's scheduling information is used for service execution and management—*duration* represents the maximum time the server can take to finish the service, and *start_time* and *end_time* represent the time during which the agreement is valid. In this case, the agreement specifies that agent CSD can invoke agent DD to cost and design a customer network whenever it is required between 09:00 and 18:00 and each service execution should take no more than 320 minutes. The agreement also contains constraints such as the volume of invocations permissible between the start and end times, the price paid per invocation, and the penalty the server incurs for every violation. The penalty mechanism, in a similar manner to the leveled commitment protocol of Sandholm (section 3.2.4), models commitments (see section 2.2.5). *client_info* specifies the information the client must provide to the server at service invocation (in this case CSD must provide the customer profile) and *reporting_policy* specifies the information the server returns upon completion.

These issues are formally specified next. Let i ($i \in \{a, b\}$) represent the negotiating agents and j ($j \in \{1, ..., n\}$) the issues under negotiation. The set of issues in real world negotiations is assumed to be finite. Let $D^i_j = [min^i_j, max^i_j]$ be the intervals of values for *quantitative* issue j acceptable by agent i. Values for *qualitative* issues, in turn, are defined over a fully ordered domain — $D^i_j = \langle q_1, \ldots, q_n \rangle$. However, because qualitative issues do not have interval values they can not be handled in a similar fashion to quantitative issues. The solution to this problem is to redefine min^a_j or max^a_j of a qualitative issue as the

Contract Name	*Instantiated Values*
service_name:	cost_&_design_customer_network
contract_id:	a1001
server_agent:	DD
client_agent:	CSD
contract_delivery_type:	on-demand
duration: (minutes)	320
start_time (GMT):	9:00
end_time (GMT):	18:00
volume (per invocation):	35
price: (per costing)	35
penalty (per lateness):	30
client_info:	customer_profile
reporting_policy:	customer_quote

Figure 4.4: Sample Contract

maximum and minimum score of the issue. The notion of a score is introduced below, but a score informally means the utility of the issue's value. The exposition of the model only concentrates on quantitative issues. The extension of the current model that formally models qualitative issues can be found in (Matos, Sierra, & Jennings 1998).

Here the formalism is restricted to considering issues for which negotiation amounts to determining a value between an agent's defined delimited range. Each agent has a scoring function $V^i_j : D^i_j \rightarrow [0, 1]$ that gives the score agent i assigns to a value of issue j in the range of its acceptable values. For convenience, scores are kept in the interval $[0, 1]$.

The next element of the model of an issue is the relative importance that an agent assigns to each issue under negotiation. w^i_j is the importance of issue j for agent i. The weights of agents are normalized, i.e. $\sum_{1 \leq j \leq n} w^i_j = 1$, for all i in $\{a, b\}$. With these elements in place, it is now possible to define an agent's scoring function[3] for a *contract*—that is, for a value $x = (x_1, ..., x_n)$ in the multi-dimensional space defined by the issues' value ranges:

$$V^i(x) = \sum_{1 \leq j \leq n} w^i_j V^i_j(x_j) \tag{4.1}$$

The additive scoring system is, for simplicity, a function V^a_j that either increases or decreases mono-

[3] Non-linear approaches to modeling utility could be used if necessary, without affecting the basic ideas of the model.

tonically. The additive scoring function is a model of how an agent can consolidate individual preferences over each issue into a single preference. The advantages of this model, in comparison to elimination by aspects and lexicographic semi-order models, were discussed in section 3.2.6. In addition to these, if both negotiators use such an additive scoring function, Raiffa showed it is possible to compute the optimum value of x (see (Raiffa 1982), p.164). Furthermore, the individual utility functions that are consolidated by the additive scoring system need to be reversible (denoted as V_j^{-1}) because, as will be shown in section 4.5.2, the trade-off mechanism requires a mapping back from a score of an issue to its value.

As an illustration of the above model consider the following example. Let the set of negotiation issues for a server agent a consist of $\{price, volume\}$—the price required to provide the service and the number of service instances attainable by a. In addition to this, let a have the following values $[min^a_{price}, max^a_{price}] = [10, 20]$ and $[min^a_{volume}, max^a_{volume}] = [1, 5]$. Also assume a views the price as more important than the volume by assigning a higher weight to price, where $(w^a_{price}, w^a_{volume}) = (0.8, 0.2)$. Finally, let the value of an offer x, for an issue j, $V^a_j(x_j)$, be modeled as a linear function:

$$V^a_{price}(x_{price}) = \frac{x_{price} - min^a_{price}}{max^a_{price} - min^a_{price}}$$

$$V^a_{volume}(x_{volume}) = 1 - \frac{x_{volume} - min^a_{volume}}{max^a_{volume} - min^a_{volume}}$$

Now consider two contracts, $(11, 5)$ and $(15, 2)$, offered by a client b to the server a. Given the above parameters for a, the value for the first offered price by b is $(11 - 10/20 - 10) = 0.1$, while the value for the first requested volume is $(1 - (5 - 1/5 - 1) = 0$. The *total* value for the offered contract is the sum of the weighted values for each individual issue (namely, 0.8*0.1+0.2*0=0.08). By the same reasoning, the value of the second contract from b is 0.55. Since the rational action is to maximize utility, a prefers the second contract offered by b.

4.2.2 Agents and Roles

In service-oriented negotiations, agents can undertake two possible roles that are, in principle, in conflict. Hence, for notational convenience two subsets of agents are distinguished [4], $Agents = Clients \cup Servers$. Roman letters are used to represent agents; $c, c_1, c_2, \dots$ will stand for clients, $s, s_1, s_2, \dots$ for servers and $a, a_1, b, d, e, \dots$ for non-specific agents.

In general, clients and servers have opposing interests, e.g. a client wants a low price for a service, whereas the potential servers attempt to obtain the highest price. High quality is desired by clients, but not by servers, and so on. Note that roles carry information. Thus, whereas an agent may not know the exact type of the other agent (its preferences), it can reasonably assume the *direction* of change of the preferences of the other, according to its role. For example, increasing offers for the value of price are

[4] The subsets are not disjoint; an agent can participate as a client in one negotiation and as a service provider in another.

valued less by a buyer and more by a seller. Therefore, in the space of negotiation values, negotiators represent opposing forces in each one of the dimensions. In consequence, the scoring functions verify that given a client c and a server s negotiating values for issue j, then if $x_j \geq y_j$ then $(V_j^c(x_j) \geq V_j^c(y_j)$ iff $V_j^s(x_j) \leq V_j^s(y_j))$. However, in a small number of cases the clients and service providers may have a mutual interest for a negotiation issue. For example, Raiffa cites a case (Raiffa 1982, pg. 133–147) in which the Police Officers Union and the City Hall realize, in the course of their negotiations, that they both want the police commissioner fired. Having recognized this mutual interest, they quickly agree that this course of action should be selected. Thus, in general, where there is a mutual interest, the variable will be assigned one of its extreme values. Hence, these variables can be removed from the negotiation set. For instance, the act of firing the police commissioner can be removed from the set of issues under negotiation and assigned the extreme value “done”.

4.2.3 Iteration of Offers: Threads

Once the agents have determined the set of variables over which they will negotiate (possibly using the issue-manipulation protocol, section 4.1.2), the negotiation process between two agents $(a, b \in Agents)$ consists of an alternate succession of offers and counter offers of values for these variables (figure 4.1). This continues until an offer or counter offer is accepted by the other side or one of the partners terminates negotiation (e.g. because the time deadline is reached without an agreement being in place). Negotiation can be initiated by clients or servers.

The vector of values proposed by agent a to agent b at time t is represented as $x^t_{a\to b}$ and the value for issue j proposed from a to b at time t by $x^t_{a\to b}[j]$. For convenience, the model assumes that there exists a common global time (the calendar time) represented by a linearly ordered set of instants, namely $Time$, and a reliable communication medium introducing no delays in message transmission (so transmission and reception times are identical). The common time assumption is not too strong in application domains where offer and counter offers frequencies are not high.

Definition 4 *A* **Negotiation Thread** *between agents $a, b \in Agents$, at time $t_n \in Time$, noted $X^{t_n}_{a\leftrightarrow b}$, is any finite sequence of length n of the form $(x^{t_1}_{a\to b}, x^{t_2}_{b\to a}, x^{t_3}_{a\to b}, \dots)$ with $t_1, t_2 \dots \leq t_n$, where:*

1. *$t_{i+1} > t_i$, the sequence is ordered over time,*

2. *For each issue j, $x^i_{a\to b}[j] \in D^a_j$, where $D^a_j = [min^a_j, max^a_j]$ for quantitative issues, $x^{i+1}_{b\to a}[j] \in D^b_j$ with $i = 1, 3, 5, \dots$, and optionally the last element of the sequence is one of the particles $\{accept, withdraw\}$.*

A negotiation thread is **active** *at time t_n if $last(X^{t_n}_{a\leftrightarrow b}) \notin \{accept, withdraw\}$, where $last()$ is a function returning the last element in a sequence.*

An offer is assumed to be valid (that is, the agent that uttered it is committed) until a counter offer is received. If the response time is relevant, it can be included in the set of issues under negotiation. For notational simplicity, it is assumed that t_1 corresponds to the initial time value, that is $t_1 = 0$. In other words, there is a local time for each negotiation thread, that starts with the utterance of the first offer.

4.3 Responsive and Deliberative Mechanisms

The negotiation and issue protocols, described in section 4.1, do not prescribe an agent's behaviour; an agent is free to instantiate any valid traversal path according to its strategy. In the next section the wrapper decision architecture is presented, which once instantiated by a negotiating agent designer, assists an agent in performing off-line computations about the decisions involved in negotiation.

As mentioned in section 2.2.4, agents need to address the following evaluatory and offer generation decision problems: what initial offers should be sent out?, what is the range of acceptable agreements?, what counter offers should be generated?, when should negotiation be abandoned? and when is an agreement reached? These decision problems are formally addressed in this chapter by developing a generic model of negotiation for the wrapper.

The offer generation components (or what is referred to as the mechanisms) of the architecture are distinguished from one another by the following properties:

1. the computational and informational cost the mechanism incurs on the agent

2. the social benefit of the mechanism for the community of *agents* that are negotiating

The first property is a feature which distinguishes this work from many of the game theory models. The provisioning of a service is a real time process. Thus services are required within tight scheduling windows and a negotiation mechanism must respect the agent's time limits. Furthermore, negotiation is only a single element of the agent's deliberations. Other agent modules need deliberation resources. Therefore, the negotiation wrapper must not consume too much of the agent's resources. Agents are also informationally, as well as, computationally bounded.

The second property relates to the concern for the design of a mechanism that achieves some measure of social (or global) welfare from local processing. Using these properties, different mechanisms can be distinguished that are concerned with the individual utility of the outcomes without concern for the social welfare, and ones that produce outcomes that are both individually and jointly preferred by the agents. For example, if a deal is required very soon then negotiation between the $IPCA$ and SPA agents is driven by concern for a deal that is perhaps not socially optimal but one that is agreeable by both agents. On the other hand, for reasons of global goodness (or social welfare) of the system, if there is time to negotiate then the same negotiation between the $IPCA$ and a SPA may involve both agents searching for deals that are not only individually rational, but may also be beneficial to the other agent. Additionally, in comparison to the former search, the latter search is likely to be more computational and informationally costly.

Given these properties, three mechanisms have been developed, namely responsive, trade-off and issue manipulation mechanisms, which differentially implement these properties. Figure 4.5 describes the execution model of the agent's reasoning during negotiation. Given the negotiation deadline (t^a_{max}), the

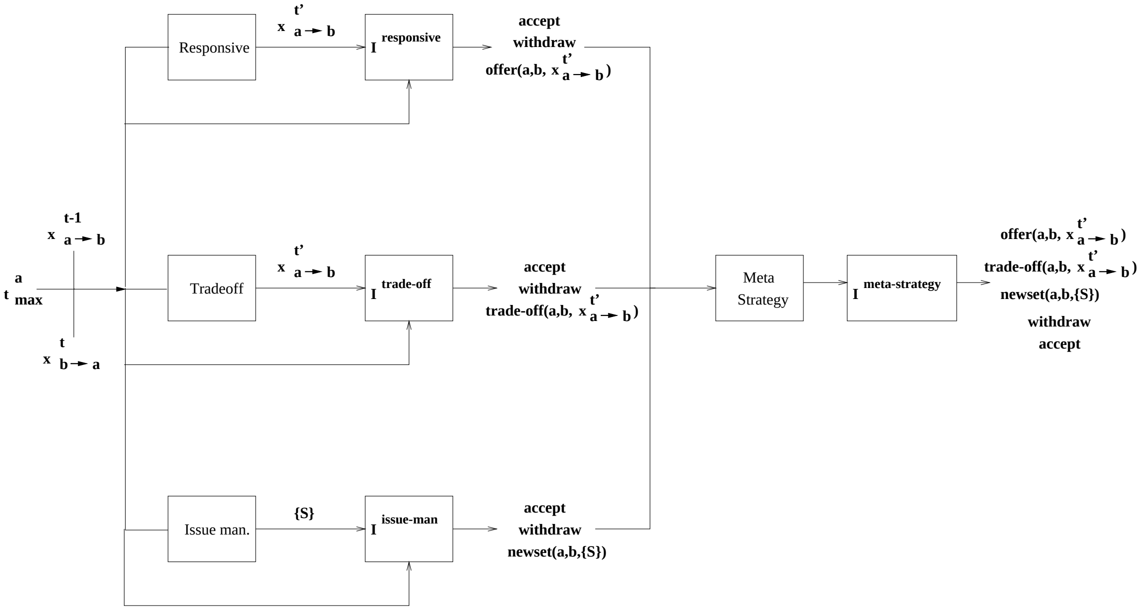

Figure 4.5: Functional View of the Agent Architecture.

opponent's last offer ($x^t_{b\rightarrow a}$) and the agent's last offer ($x^{t-1}_{a\rightarrow b}$) the responsive and trade-off mechanisms simultaneously compute a new offer ($x^{t'}_{a\rightarrow b}$) while the issue manipulation mechanism may generate a new set of negotiation issues. The mechanism's evaluatory component ($I^{responsive}$, $I^{trade-off}$, $I^{issue-man}$ in figure 4.5) then makes the decision to either accept ($accept$) or reject ($withdraw$) the opponent's last offer $x^t_{b\rightarrow a}$, or offer the opponent a new contract ($x^{t'}_{a\rightarrow b}$) in the case of responsive and trade-off mechanisms or a new set of issues ($\{S\}$) in the case of issue-manipulation. The final choice of which mechanism's suggestion to offer is handled by the meta-strategy module (section 4.7). The processes involved in each mechanism are described next.

4.4 The Responsive Mechanism

Responsive mechanisms generate offers and counter offers through linear combinations of simple functions, called *tactics*. Tactics generate an offer, or counter offer, for a single component of the negotiation object (or issue) using a single criterion (time, resources or the behaviour of other agents). These criteria are motivated by an agent's computational and informational boundedness. For example, the time limits and the resources used in negotiation so far, directly constrain the granularity of the search for an outcome. With increasing time limits or on-line costs, an agent may prefer deals of lower score than ones that are higher in score but which may be unattainable given the time and resource constrains. Likewise, uncertainty of others can in the simplest way be handled by reproducing other's behaviour (Axelrod 1984). A more sophisticated uncertainty handling methodology is presented later, but the reproduction of others' behaviour has proven to be a highly successful, and computationally simple, interaction strategy (Axelrod 1984). Different weights in the linear combination allow the varying importance of the criteria to be modeled. For example, when determining the values of an issue, it may initially be more important to take into account the other agent's behaviour than the remaining time. In which case, the tactics that emphasize the behaviour of other agents will be given greater precedence than the tactics which base their value on the amount of time remaining.

However, agents need to monitor and be responsive to their changing environment. Therefore, to achieve flexibility in negotiation, the agents may wish to change their ratings of the importance of the different criteria over time. For example, remaining time may become correspondingly more important than the imitation of the other's behaviour as the time by which an agreement must be in place approaches. This modifying behaviour is referred to as a *strategy* and it denotes the way in which an agent changes the weights of the different tactics over time. Thus, strategies combine tactics depending on the history of negotiations and the internal reasoning model of the agents, and negotiation threads influence one another by means of strategies (see section 4.4.3).

4.4.1 Evaluation Decisions

When agent a receives an offer from agent b at time t, $x^t_{b\to a}$ (represented as y in figure 4.5), it has to rate the offer using its scoring function. If the value of $V^a(x^t_{b\to a})$ is greater than the value of the counter offer agent a is ready to send at the time t' when the evaluation is performed, that is $x^{t'}_{a\to b}$ with $t' > t$ ($x' > y$ in figure 4.5), then agent a accepts. Otherwise, the counter offer is submitted by the mechanism to the meta-strategy component. Expressing this concept more formally:

Definition 5 *Given an agent a and its associated scoring function V^a, $a's$* **interpretation** *(I) at time t' of an offer $x^t_{b\to a}$ sent at time $t < t'$, is defined as:*

$$I_a^{responsive}(t', x^t_{b\to a}) = \begin{cases} withdraw(a,b) & \mathit{If}\, t' > t^a_{max} \\ accept(a,b,x^t_{b\to a}) & \mathit{If}\, V^a(x^t_{b\to a}) \geq V^a(x^{t'}_{a\to b}) \\ \mathit{offer}(a,b,x^{t'}_{a\to b}) & otherwise \end{cases}$$

where $x^{t'}_{a\to b}$ is the contract that agent a would offer to b at the time of the interpretation and t^a_{max} is a constant that represents the time by which a must have completed the negotiation.

The result of $I_a^{responsive}(t', x_{b\rightarrow a}^t)$ is one of the primitives specified in the negotiation protocol (figure 4.1 section 4.1.1). The primitive *offer* is used to extend the current negotiation thread between the agents with a new offer $x_{a\rightarrow b}^{t'}$ (ϕ in figure 4.1). The primitives $accept$ and $withdraw$ terminate the negotiation. The evaluation function can also be viewed as representing the goal-test function of section 2.2.8 that evaluates whether a goal state has been reached or not (an agreement in the form of cross-over in offers). This interpretation formulation also allows modeling of the fact that a contract unacceptable today can be accepted tomorrow merely by the fact that time has passed.

4.4.2 Offer Generation Decisions—Tactics

In order to prepare a counter-offer, $x_{a\rightarrow b}^{t'}$, agent a uses a set of *simple* functions called *tactics*, that generate new values for each variable in the negotiation set. The following families of tactics have been developed:

1. **Time dependent**. If an agent has a time deadline by which an agreement must be in place, these tactics model the fact that the agent is likely to concede more rapidly as the deadline approaches. The shape of the curve of concession, a function depending on time, is what differentiates tactics in this set.

2. **Resource dependent**. These tactics model the pressure in reaching an agreement that the limited resources—e.g. remaining bandwidth to be allocated, money, or any other—and the environment—e.g number of clients, number of servers or economic parameters—impose upon the agent's behaviour. The functions in this set are similar to the time dependent functions except that the domain of the function is the quantity of resources available instead of the remaining time.

3. **Behaviour dependent** or **Imitative**. In situations in which the agent is not under a great deal of pressure to reach an agreement, it may choose to use imitative tactics to protect itself from being exploited by other agents. In this case, the counter offer depends on the behaviour of the negotiation opponent. Another function of this tactic family is to provide default behaviours when there is uncertainty about what action to take (see section 2.2.6). The imitation of others' behaviour can thus serve as a default action when an agent is uncertain about what to do next. The tactics in this family differ in which aspect of their opponent's behaviour they imitate and to what degree the opponent's behaviour is imitated.

This set of tactics is motivated by the domain characteristics of many types of problems mentioned in section 1.4.3, where the time and resources of an agent and the behaviour of other agents are key features. Unlike the models of chapter three, these tactics explicitly motivate rationales for concessions or demands, based on a number of environmental and behavioural characteristics. They determine how to compute the value of an issue (price, volume, duration, quality, ...), by considering a *single* criterion (time, resources,

...). The set of values for the negotiation issue are then the range of the function and the single criterion is its domain.

Given that agents may want to consider more than one criterion to compute the value for a single issue, the generation of counter proposals is modeled as a weighted combination of different tactics covering the set of criteria. The values so computed for the different issues will be the elements of the counter proposal. [5] For instance, if an agent wants to counter-propose taking into account two criteria: the remaining time and the previous behaviour of the opponent, it can select two tactics: one from the time dependent family and one from the imitative family. Both of these tactics will suggest a value to counter propose for the issue under negotiation. The actual value which is counter proposed will be the weighted combination of the two independently generated values.

To illustrate these points consider the following example. Given an issue j, for which a value is under negotiation, an agent a's initial offer corresponds to a value in the issue's acceptable region, (i.e in $[min_j^a, max_j^a])$. For instance, if $a's$ range is $[£0, £20]$ for the price p to pay for a good, then it may start the negotiation process by offering the server $£10$ —what initial offer should be chosen is something the agent can learn by experience. The server, agent b, with range $[£17, £35]$ may then make an initial counter-offer of $£25$. With these two initial values, the strategy of agent a may consist of using a (single criterion) time dependent tactic which might make a reasonably large concession and suggest $£15$ since it does not have much time in which to reach an agreement. Agent b, on the other hand, may chose to use two criteria to compute its counterproposal—e.g a time dependent tactic (which might suggest a small concession to $£24$ since it has a long time until the deadline) and an imitative tactic (which might suggest a value of $£20$ to mirror the $£5$ shift of the opponent). If agent b rates the time dependent behaviour three times as important as the imitative behaviour, then the value of the counter-offer will be $(0.75 * 24) + (0.25 * 20) = £23$. This process continues until the agents converge on a mutually acceptable solution. The origin, and subsequent evolution of these relative weightings may be the result of expert domain knowledge, experience derived from previous negotiation cases, or conditional on other factors.

It should be noted that not all tactics can be applied at all instants. For instance, a tactic that imitates the behaviour of an opponent is only applicable when the opponent has shown its behaviour sufficiently. For this reason, the following description of the tactics pays particular attention to their applicability conditions.

4.4.2.1 Time Dependent Tactics

In these tactics, the predominant factor used to decide which value to offer next is time, t. Thus these tactics consist of varying the acceptance value for the issue depending on the remaining negotiation time (an important requirement in the target problem domains—section 1.4.3), modeled as the above defined constant t_{max}^a. The initial offer is modeled as being a point in the interval of values of the issue under

[5] Values for different issues may be computed by different weighted combinations of tactics.

negotiation. Hence, agents define a constant κ_j^a that when multiplied by the size of the interval, determines the value of issue j to be offered in the first proposal by agent a.

The value to be uttered by agent a to agent b for issue j is modeled as the offer at time t, with $0 \leq t \leq t_{max}^a$, by a function α_j^a depending on time as the following expression shows:

$$x_{a \to b}^t[j] = \begin{cases} min_j^a + \alpha_j^a(t)(max_j^a - min_j^a) & \text{If } V_j^a \text{ is decreasing} \\ min_j^a + (1 - \alpha_j^a(t))(max_j^a - min_j^a) & \text{If } V_j^a \text{ is increasing} \end{cases}$$

A wide range of time dependent functions can be defined simply by varying the way in which $\alpha_j^a(t)$ is computed. However, functions must ensure that $0 \leq \alpha_j^a(t) \leq 1$, $\alpha_j^a(0) = \kappa_j^a$ and $\alpha_j^a(t_{max}^a) = 1$. That is, the offer will always be between the value range, at the beginning it will give κ_j^a as a result and when the time deadline is reached the tactic will suggest to offer the reservation value[6]. Two families of functions with this intended behaviour are distinguished: polynomial and exponential (naturally, others could also be defined). Both families are parameterized by a value $\beta \in I\!R^+$ that determines the convexity degree (see Figure 4.6) of the curve. These two families of functions were chosen because of the very different way they model concession. For the same large value of β, the polynomial function concedes faster at the beginning than the exponential one, then they behave similarly. For a small value of β, the exponential function waits longer than the polynomial one before it starts conceding:

- **Polynomial:** $\alpha_j^a(t) = \kappa_j^a + (1 - \kappa_j^a)(\frac{min(t, t_{max}^a)}{t_{max}^a})^{\frac{1}{\beta}}$
- **Exponential:** $\alpha_j^a(t) = e^{(1 - \frac{min(t, t_{max}^a)}{t_{max}^a})^{\beta} \ln \kappa_j^a}$

In comparison to Kasbah (section 3.2.9) that only models three offer generation functions, these families of functions represent an infinite number of possible tactics, one for each value of β. However, to better understand their behaviour they are classified, depending on the value of β, into two extreme sets showing clearly different patterns of behaviour. Other sets in between these two could also be defined:

1. **Boulware [7] tactics [(Raiffa 1982), pg. 48]**. Either exponential or polynomial functions with $\beta < 1$. This tactic maintains the offered value until the time is almost exhausted, whereupon it concedes up to the reservation value[8]. The behaviour of this family of tactics with respect to β is easily

[6] The reservation value for issue j of agent a represents the value that gives the smallest score for function V_j^a. The function V_j^a depends on the reservation value for agent a and issue j—the range $[min_j^a, max_j^a]$. If V_j^a is monotonically increasing, then the reservation value is min_j^a; if it is decreasing the reservation value is max_j^a.

[7] Lemuel Boulware was a vice-president of the General Electric Company, who rarely made concessions in wage negotiations. His strategy was to start with what he deemed to be a fair opening bid and held firm throughout the negotiations.

[8] Besides the pattern of concession that these functions model, Boulware negotiation tactics presume that the interval of values for negotiation is narrow. Hence, when the deadline is reached and $\alpha(t_{max}^a) = 1$, the offer generated is not substantially different from the initial one.

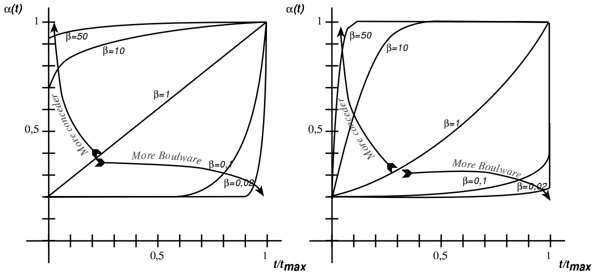

Figure 4.6: Polynomial (left) and Exponential (right) Functions for the Computation of $\alpha(t)$. Time is Presented as Relative to t^a_{max}.

understood taking into account that $\lim_{\beta \to 0^+} e^{(1-\frac{min(t,t^a_{max})}{t^a_{max}})^\beta \ln \kappa^a_j} = \kappa^a_j$ and $\lim_{\beta \to 0^+} \kappa^a_j + (1 - \kappa^a_j)(\frac{min(t,t^a_{max})}{t^a_{max}})^{\frac{1}{\beta}} = \kappa^a_j$.

The Boulware tactics can be selected as a technique to handle uncertainty (see section 2.2.6.2); when others' preferences are unknown, then one possible strategy is to remain firm and demand the same throughout the negotiation.

2. **Conceder [(Pruitt 1981), pg. 20]**. Either exponential or polynomial functions with $\beta > 1$. The agent quickly goes to its reservation value. For similar reasons as before, we have $\lim_{\beta \to +\infty} e^{(1-\frac{min(t,t^a_{max})}{t^a_{max}})^\beta \ln \kappa^a_j} = 1$ and $\lim_{\beta \to +\infty} \kappa^a_j + (1 - \kappa^a_j)(\frac{min(t,t^a_{max})}{t^a_{max}})^{\frac{1}{\beta}} = 1$.

Resource-dependent tactics are similar to the time dependent ones. Indeed, time dependent tactics can be seen as a type of resource dependent tactic in which the sole resource considered is time. Whereas time vanishes constantly up to its end, other resources may have different patterns of usage. Time and resource dependent tactics are also similar in that they are both an attempt to model bounded rationality (see section 2.2.8), in that they attempt to generate successful outcomes given the available information and computational resources. Resource dependent tactics are modeled in the same way as time dependent ones; that is, by using the same functions, but by either: i) making the value of t^a_{max} dynamic or ii) making the function α depend on an estimation of the amount of a particular resource.

4.4.2.2 Dynamic Deadline Tactics

The dynamic value of t^a_{max} represents a heuristic about the quantity of resources that are in the environment. The scarcer the resource, the more urgent the need for an agreement. In the target application domains, the most important resource to model is the number of agents negotiating with a given agent and how impatient they are to reach agreements. On one hand, the greater the number of agents who are negotiating with agent a for a particular service s, the lower the pressure on a to reach an agreement with any specific individual. While on the other hand, the longer the negotiation thread, the greater the pressure on a to come to an agreement. Hence, representing the set of agents negotiating with agent a at time t as: $N^a(t) = \{i|X^t_{i\leftrightarrow a}\text{is active}\}$, the dynamic time deadline for agent a is defined as:

$$t^a_{max} = \mu^a \frac{|N^a(t)|^2}{\sum_i |X^t_{i\leftrightarrow a}|}$$

where μ^a represents the time agent a considers reasonable to negotiate with a single agent and $|X^t_{i\leftrightarrow a}|$ represents the length of the current thread between i and a. Notice that the number of agents is in the numerator, so quantity of time is directly proportional to it, and averaged length of negotiation thread is in the denominator, so quantity of time is inversely proportional to it.

4.4.2.3 Resource Estimation Tactics

The resource estimation tactics generate counter-offers depending on how a particular resource is being consumed. Resources could be money being transferred among agents, the number of agents interested in a particular negotiation, and also, in a similar way as before, time. The required behaviour is for the agent to become progressively more conciliatory as the quantity of resource diminishes. The limit when the quantity of the resource approaches nil is to concede up to the reservation value for the issue(s) under negotiation. When there is plenty of resource, a more Boulware behaviour is to be expected. Formally, this can be modeled by having a different computation for the function α:

$$\alpha^a_j(t) = \kappa^a_j + (1 - \kappa^a_j)e^{-resource^a(t)}$$

where the function $resource^a(t)$ measures the quantity of the resource at time t for agent a. Examples of functions are:

- $resource^a(t) = |N^a(t)|$
- $resource^a(t) = \mu^a \frac{|N^a(t)|^2}{\sum_i |X^t_{i\leftrightarrow a}|}$
- $resource^a(t) = \min(0, t - t^a_{max})$

In the first example, the number of negotiating agents is the resource. That is, the more agents negotiating the less pressure to make concessions. The second example models time as a resource in a similar way as in the previous section. The more agents, the less pressure, and the longer the negotiations the more pressure. Finally, the last case also models time as a resource, but in this case the quantity of resource decreases in a linear fashion with respect to time.

4.4.2.4 Behaviour Dependent Tactics

This family of tactics compute the next offer based on the previous attitude of the negotiation opponent. These tactics have proved important in co-operative problem-solving negotiation settings (Axelrod 1984), and so are useful in a subset of the problem contexts (see Section 1.4.3). Like Boulware tactics, these tactics can also be selected for as a technique for handling uncertainty. However, whereas Boulware tactics handle the uncertainty of strategic interaction by ignoring the behaviour of the opponent, these tactics condition their actions on the observed behaviour of the other(s).

The main difference between the tactics in this family is in the type of imitation they perform. One family imitates proportionally, another in absolute terms, and the last one computes the average of the proportions in a number of previous offers. Hence, given a negotiation thread

$$\{\ldots, x_{b\to a}^{t_{n-2\delta}}, x_{a\to b}^{t_{n-2\delta+1}}, x_{b\to a}^{t_{n-2\delta+2}}, \ldots, x_{b\to a}^{t_{n-2}}, x_{a\to b}^{t_{n-1}}, x_{b\to a}^{t_n}\}$$

with $\delta \geq 1$, the following families of tactics are distinguished:

1. **Relative Tit-For-Tat:** The agent reproduces, in percentage terms, the behaviour that its opponent performed $\delta \geq 1$ steps ago. The condition of applicability of this tactic is $n > 2\delta$.

$$x_{a\to b}^{t_{n+1}}[j] = min(max(\frac{x_{b\to a}^{t_{n-2\delta}}[j]}{x_{b\to a}^{t_{n-2\delta+2}}[j]} x_{a\to b}^{t_{n-1}}[j], min_j^a), max_j^a)$$

2. **Random Absolute Tit-For-Tat:** The same as before but in absolute terms. This means that if the other agent decreases its offer by $\pounds 2$, then the next response should be increased by the same $\pounds 2$. Moreover, a component is added which modifies that behaviour by increasing or decreasing (depending on the value of parameter s) the value of the answer by a random amount. This random element is introduced to enable the agents to escape from a loop of non-improving contract offers, or a local minima in the social welfare function (meaning that the contracts being exchanged have the same utility to both agents). M is the maximum amount by which an agent can change its imitative behaviour. The condition of applicability is again $n > 2\delta$.

$$x_{a\to b}^{t_{n+1}}[j] = min(max(x_{a\to b}^{t_{n-1}}[j] + (x_{b\to a}^{t_{n-2\delta}}[j] - x_{b\to a}^{t_{n-2\delta+2}}[j]) + (-1)^s R(M), min_j^a), max_j^a)$$

where

$$s = \begin{cases} 0 & \text{If } V_j^a \text{ is decreasing} \\ 1 & \text{If } V_j^a \text{ is increasing} \end{cases}$$

and $R(M)$ is a function that generates a random value in the interval $[0, M]$.

3. **Averaged Tit-For-Tat:** The agent computes the average of percentages of changes in a window of size $\gamma \geq 1$ of its opponents history when determining its new offer. When $\gamma = 1$ the behaviour is similar to the relative Tit-For-Tat tactic with $\delta = 1$. The condition of applicability for this tactic is $n > 2\gamma$.

$$x_{a \to b}^{t_{n+1}}[j] = min(max(\frac{x_{b \to a}^{t_{n-2\gamma}}[j]}{x_{b \to a}^{t_n}[j]} x_{a \to b}^{t_{n-1}}[j], min_j^a), max_j^a)$$

Different tit-for-tat tactics were designed to empirically evaluate, similar to the tournament games of Axelrod (Axelrod 1984), the relative success of different manners in reproducing behaviour of others.

4.4.3 Strategic Reasoning—Strategies

The aim of agent a's negotiation strategy is to determine the best course of action (see section 2.2.4) which will result in an agreement on a contract x while keeping V^a as high as possible. However, maximization of the scoring function (a task of the wrapper) must consider changes in the agent's environment. This task-environment coupling is needed because an agent's behaviour should change as the environment changes (hence the name responsive for the mechanism). In practical terms, this equates to how to prepare a new counter offer, taking into consideration a number of ever changing factors.

In the model, an agent has a representation of its mental state containing information about its beliefs, its knowledge of the environment (for example, time or resources), and any other attitudes (desires, goals, obligations or intentions) the agent designer considers appropriate[9]. The mental state of agent a at time t is noted as MS_a^t. The set of all possible mental states for agent a is denoted as MS_a.

When agent a receives an offer from agent b, it becomes the last element in the current negotiation thread between the agents. If the offer is unsatisfactory, agent a generates a counter offer. As discussed earlier, different combinations of tactics can be used to generate counter offers for particular issues. An agent's strategy determines which combination of tactics should be used at any one instant (this concept is similar to the concept of mixed strategies in game theoretic models (Gibbons 1992)).

Definition 6 *Given a negotiation thread between agents a and b at time t_n, $X_{a \leftrightarrow b}^{t_n}$, over domain $D = D_1 \times \cdots \times D_p$, with $last(X_{a \leftrightarrow b}^{t_n}) = x_{b \to a}^{t_n}$, and a finite set of m tactics*[10] *$T^a = \{\tau_i | \tau_i : MS_a \to D\}_{i \in [1,m]}$, a* **weighted counter proposal**, *$x_{a \to b}^{t_{n+1}}$, is a linear combination of the tactics given by a matrix of weights $\Gamma_{a \to b}^{t_{n+1}}$*

[9] There is no prescription of a particular mental state, but rather this work aims towards an architecturally neutral description to ensure maximum generality for the model.

[10] This definition uses the natural extension of tactics to the multi-dimensional space of issues' values.

$$\Gamma_{a\to b}^{t_{n+1}} = \begin{pmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1m} \\ \gamma_{21} & \gamma_{22} & \cdots & \gamma_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ \gamma_{p1} & \gamma_{p2} & \cdots & \gamma_{pm} \end{pmatrix}$$

defined in the following way:

$$x_{a\to b}^{t_{n+1}}[j] = (\Gamma_{a\to b}^{t_{n+1}} * T^a(MS_a^{t_{n+1}}))[j,j]$$

where $(T^a(MS_a^{t_{n+1}}))[i,j] = (\tau_i(MS_a^{t_{n+1}}))[j]$, $\gamma_{ji} \in [0,1]$ and for all issues j, $\sum_{i=1}^{m} \gamma_{ji} = 1$.

The weighted counter proposal extends the current negotiation thread as follows ($\bullet$ is the sequence concatenation operation):

$$X_{a\leftrightarrow b}^{t_{n+1}} = X_{a\leftrightarrow b}^{t_n} \bullet x_{a\to b}^{t_{n+1}}$$

Many-party negotiations are modeled by means of a set of interacting negotiation threads. The way this is done is by making a negotiation thread influence the selection of which matrix Γ is to be used in other negotiation threads. Thus,

Definition 7 *Given $a, b \in Agents$, a* **Negotiation Strategy** *for agent a is any function f such that, given a's mental state at time t_n, $MS_a^{t_n}$, and a matrix of weights at time t_n, $\Gamma_{a\to b}^{t_n}$, generates a new matrix of weights for time t_{n+1}, i.e.*

$$\Gamma_{a\to b}^{t_{n+1}} = f(\Gamma_{a\to b}^{t_n}, MS_a^{t_n}) \tag{4.2}$$

A simplistic example of the application of the model would be to have a matrix Γ built up of 0s and 1s and having $\Gamma_{a\to b}^{t+1} = \Gamma_{a\to b}^{t}$ for all t. This would correspond to using a fixed single tactic for each issue at every instant in the negotiation. Consider another example of when a weighted combination, as opposed to binary and static weighting, could be useful. The example involves negotiation between the VC (Vet Customer agent) and the CSD (Customer Service Department agent) for the $Vet\_Customer$ service, taken from the ADEPT application (section 1.4.1). For simplicity assume that there is only a single issue, the price of the service. Further assume that both agents are currently under no time pressure to reach an agreement. Given these conditions then both agents may begin negotiation by assigning a value of 1 to the Boulware tactic and 0 to all others. However, after the exchange of a number of offers and an increase in time pressure to reach a deal, one (or both) of the agent(s) may begin to reduce the weighting of the Boulware tactic and begin to place higher weighting on the Conceder tactic (believing that concession may result in an agreement being reached sooner rather than later in the negotiation). This example informally shows the usefulness of strategies in modeling a *smooth* transition from a behaviour based on a single tactic (e.g. Boulware, because the agent has plenty of time to reach an agreement) to another one (e.g. Conceder, because time is running out). Smoothness is obtained by changing the weights affecting the tactics progressively (e.g. from 1 to 0 and from 0 to 1 in the example). The current model has been extended to include the evolution of strategies (Matos, Sierra, & Jennings 1998).

4.4.4 Functional Architecture of the Responsive Mechanism

The above model is a *generic* description of the components of the responsive mechanism. It is generic because there can be an infinite number of tactics (and their corresponding strategies)—the model does not commit to any particular agent architecture by specifying that an agent's decision mechanism should be described through N tactics and their corresponding strategies. However, for practical purposes agent architectures are needed that commit to a concrete instantiation, and follow from, this generic model. A responsive agent architecture has been developed to empirically evaluate the behaviour of different tactics and strategies (described in the next chapter), and which can be used as the responsive mechanism component of the negotiation wrapper shown in figure 1.1.

The overall architecture of this responsive mechanism is shown in figure 4.7. The boxes labeled *Expo/Poly, resource* and *tit-4-tat* represent the time, resource and behaviour dependent tactics respectively. The unfilled ovals represent the input parameters into both the tactics and, possibly, the strategy. The latter inputs are the possible set of inputs because in the formal model nothing is said about the actual mental state of the agent. The output of each tactic (the offer suggested by each tactic, represented as $x'_{td}, x'_{rd}, x'_{bd}$ for the contract offer suggested by the time, resource and behaviour dependent tactics respectively) is represented as filled ovals. The agent's strategy then modifies the weights attached to each tactic (represented by boxed ovals, labeled w_{td}, w_{rd} and w_{bd}, for weights of the time, resource and behaviour dependent tactics respectively). The final offer, filled oval labeled x', is then computed as the summation of individual offers from the tactics, after being modified by their strategy selected weights, represented as the $*+$ operation. The value of this final offer, represented as filled oval labeled $V(x')$, is computed as the linear sum of all the issue's weighted values, represented by the box $+w_i * V(x'_i)$. The responsive mechanism was developed as a set of simple functions that solves the decision making problems of an agent given its limited information and computational capabilities. The decision mechanism of the wrapper was then extended by two more complex (deliberative) mechanisms, namely an issue trade-off mechanism (section 4.5.2) and an issue manipulation mechanism (section 4.6). These deliberative mechanisms are discussed next.

4.5 The Trade-off Mechanism

The responsive mechanism implements an iterated search for a contract with a value that is acceptable to both parties. The mechanism can be used to model iterative concession over the score of the contract by

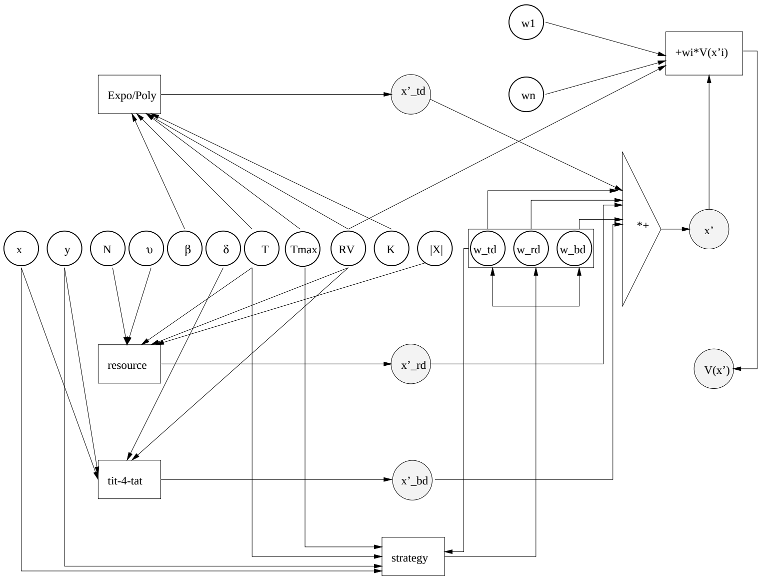

Figure 4.7: Functional View of the Responsive Mechanisms. Ovals depicts data structures, boxes processes, and arrows, flow of information

an agent (based on a number of environmental factors, such as the deadline or the amount of computational resources used), until a point of intersection (or what will be referred to as a cross over of offers) occurs between the value of the offered contract and what the agent is about to offer. Although this mechanism proved useful in a number of real-world applications (FIPA97 1997, Jennings *et al.* 2000a), cross over evaluation is inefficient in that it fails to find joint gains, reaching outcomes that lie closer to the pareto-optimal line (Gibbons 1992). In particular, the mechanism cannot discriminate between contracts that have different scores for the issues, but which have the same overall score (Corfman & Gupta 1993). Therefore, possible joint gains are missed. To improve the *efficiency* of the outcome, while respecting the information and computational constraints, a trade-off mechanism has been designed that searches for potential joint gains. The interpretation component of this mechanism is described first in section 4.5.1 followed by the offer generation mechanism in section 4.5.2.

4.5.1 Trade-off Mechanism Evaluation

The evaluation of a contract from the trade-off mechanism perspective involves:

$$I_a^{\text{trade-off}}(t, x^t_{b\to a}) = \begin{cases} withdraw(a,b) & \text{If } t > t^a_{max} \\ accept(a,b,x^t_{b\to a}) & \text{If } V^a(x^t_{b\to a}) \geq V^a(x^{t-1}_{a\to b}) \\ \text{trade-off}(a,b,x^{t'}_{a\to b}) & otherwise \end{cases}$$

where the content of the primitive *trade-off* ($x^{t'}_{a \to b}$ or ϕ in figure 4.1) is computed by the function given in equation 4.4. Note the similarity between the trade-off and responsive mechanism (section 4.4.1) evaluation function. In both interpretations, negotiation terminates unsuccessfully for the same reason; when the end time of the negotiation has been reached. However, the interpretation functions do differ. Negotiation terminates successfully in the responsive mechanism when the value of the offered contract is higher than the one the agent is *about* to send out ($x^{t'}$). Negotiation terminates successfully in the trade-off mechanism when the value of the offered contract is higher than the *previous* offer of the agent (x^{t-1}). This is because, as will be shown, the trade-off mechanism can only hill-climb (in utility landscape) in the direction of higher utility for the agent performing the trade-off. Therefore, the offered contract, from the other agent, has to have a lower utility to the agent performing the trade-off. Likewise, any mechanism must respect the time deadlines of negotiation. As will be shown in this section, the real difference between the two interpretations are the mechanisms involved in generating the primitives *offer* and *trade-off.*

In spite of the similarities between responsive and trade-off interpretations (and as will be shown below in section 4.6.1) the evaluation components of each mechanism are functionally separated from one another (see figure 4.5). This separation of concerns between the interpretation component of each mechanism and its respective offer generation component allows differential and modular reasoning interpretation policies to be adopted for each mechanism according to the requirements of the agent designer.

4.5.2 Trade-off Mechanism Offer Generation

In the responsive mechanism, agents propose a series of contracts that have diminishing score to themselves. However, in choosing to make a trade-off negotiation action an agent is seeking to find a contract that has the same score as its previous proposal, but which is more acceptable to (has higher score for) its negotiation opponent. Therefore, when an agent implements a trade-off mechanism it behaves as though it is motivated to search for types of outcomes that increase *joint* gains. The next section presents the developed solution to the problem of how to reason about “more acceptable” contracts given the uncertainty of the opponent’s preferences.

4.5.2.1 Fuzzy Similarity

The computation involved in making a trade-off over issues in negotiation is likely to be more costly than the simple responsive mechanisms described above. However, an agent may be cooperatively motivated to increase the joint gains over an outcome given the costs involved. For example, two agents can engage in a more elaborate search of the space of possible outcomes if one or both are under no time pressures to reach an agreement soon. Furthermore, the trade-off mechanism must select a contract that increases the likely score of the opponent, *given that the agent does not know its preferences*. This means that the agent (call this a) in negotiation with another agent (call this b) must be provided with a mechanism to:

1. select a subset of contracts all of which have the same utility as $a's$ previous offer x
2. select from this subset a contract (x') that agent a *believes* (represented by the predicate B^a) is most preferable by b over x

That is, $B^a(V_b(x') > V_b(x))$ and $V_a(x) = V_a(x')$. It therefore follows from the combination of this belief and the fact that agent a believes the proposition $B^a(V_a(x') + V_b(x') > V_a(x) + V_b(x))$ (x' increases the joint utility). The problem being addressed in this section is how to model the agent's uncertain belief (predicate B^a) in the second step of the mechanism's operation. A number of alternatives were considered (section 2.2.6) and the solutions from game theory (section 3.1.6) enumerates the various possible choices in modeling uncertainties. Computing conditional probabilities and formulating subjective expected utility appears a reasonable methodology for handling the uncertainties involved. However, as noted in section 2.2.6, the approach is problematic. Firstly, assigning prior probabilities is practically impossible for the types of problems addressed here (where there can be an infinitely large set of outcomes and the outcome set itself can change dynamically in the course of negotiation through the inclusion and retraction of issues). Even if assigning prior probabilities was practically achievable for interactions that are repeated (hence permitting the use of probability update mechanisms such as Bayes rule (Russell & Norvig 1995)), the same is not true for encounters in an open system—the prior probabilities may simply be wrong, exacerbated by the one-off nature of encounters, preventing the update of prior distribution. Secondly, as mentioned previously, the formulation of decisions based on subjective expected utility introduces the silent out-guessing problem—the agent designer's choice of probabilities is based on guesses about the probable choices of others, whose choice in turn is dependent on the guesses about the probable choices of the first.

Therefore a solution is sought that is simple and applicable to types of problems present in both closed and open systems. The heuristic employed in this thesis is not to directly model the likely choice of the other, but rather, to select the contract that is most "similar" or "close to" to the opponent's last proposal (since this may be more acceptable to the opponent). That is, the heuristic models the *domain* and not the other agent. The agent can then use this domain model to induce the *possible* default preferences of the other. For example, if the seller has demanded a payment of $\pounds 20$ for a service then a client of the service can heuristically assume that the seller will prefer an offer of $\pounds 18$ to $\pounds 10$ because the former is closer, or more similar, than the latter to the initial demand by the seller.

The concept of fuzzy similarity can be used to compute similarity (Zadeh 1971). This shift in emphasis from the probable choices of others to the closeness of two contracts means that any theory that makes the same ontological commitments as logic (such as probability theory, where facts are either true or not and probabilities represent the degree of *belief*) is inappropriate. However, when modeling concepts such as closeness, tallness or heaviness a different logic is required that models the degree of *truth*—a sentence is "sort of" true. Most people would hesitate to say whether the sentence "Carles is tall" is true or not, but would more likely say “sort of”. Note, this is not an uncertainty about the external world (we are sure how tall Carles is), rather it is a statement about the vagueness or uncertainty over the linguistic term “tallness” or similarity/membership of a class prototype. However, an important point to note is that the use of fuzzy similarity and probability are not exclusive. Indeed, the agent can use the heuristic of fuzzy similarity to derive the prior probabilities of the other’s choices from the domain and then update these prior probabilities in the course of interactions using Bayes rule. Thus, fuzzy similarity can be used to “bootstrap” decision mechanisms that operate on the basis of choice distributions.

The next section describes in more detail the notion of similarity and the developed algorithm for performing such trade-offs.

4.5.2.2 Trade-offs: A Formal Model

An agent will decide to make a trade-off action when it does not wish to decrease its aspirational level (denoted θ) for a given service-oriented negotiation. Thus, the agent first needs to generate some/all of the potential contracts for which it receives the score of θ. Technically, it needs to generate contracts that lie on the iso-value (or indifference) curve for θ (Raiffa 1982). An iso-value corresponds to fixing one of the x or y values in the pair (x, y) in figure 3.1 and then selecting an iso-value amounts to considering only contracts on that line. Because all these potential contracts have the same value for the agent, it is indifferent amongst them. Given this fact, the aim of the trade-off mechanism is to find the contract on this line that is most preferable (and hence acceptable) to the negotiation opponent (since this maximizes the joint gain). More formally, an iso-curve is defined as:

Definition 8 *Given an aspirational scoring value θ, the iso-curve set at level θ for agent a is defined as:*

$$iso_a(\theta) = \{x \mid V^a(x) = \theta\} \tag{4.3}$$

From this set, the agent needs to select the contract that maximizes the joint gain. A trade-off is then defined as:

Definition 9 *Given an offer, x, from agent a to b, and a subsequent counter offer, y, from agent b to a, with $\theta = V^a(x)$, a trade-off for agent a with respect to y is defined as:*

$$\textit{trade-off}_a(x, y) = \arg \max_{z \in iso_a(\theta)} \{Sim(z, y)\} \tag{4.4}$$

where the similarity, Sim, between two contracts is defined as a weighted combination of the similarity of the issues:

Definition 10 *The similarity between two contracts x and y over the set of issues J is defined as:*

$$Sim(x, y) = \sum_{j \in J} w_j^a Sim_j(x_j, y_j) \tag{4.5}$$

with $\sum_{j \in J} w_j^a = 1$ and Sim_j being the similarity function for issue j. These weights may represent the level of importance the agent believes the opponent places on issues. For example, an oil company negotiator, in negotiation with an ecologist, may safely assume that the pollution risks are weighted more importantly by an ecologist than the oil production costs when reasoning about what deal to offer.

Following the results from (Valverde 1985), a similarity function that satisfies the axioms of reflexivity, symmetry, and t-norm transitivity can always be defined as a conjunction (modeled, for instance, as the minimum) of appropriate fuzzy equivalence relations induced by a set of criteria functions h_i. In fuzzy set theory, t-norm, or triangular norms, play a central role by providing generic models for intersection and union operations on fuzzy sets (Pedrycz & Comide 1998). A criteria function is a function that maps values from a given domain into values in $[0, 1]$. Correspondingly, the similarity between two values for issue j, $Sim_j(x_j, y_j)$ is defined as:

Definition 11 *Given a domain of values D_j, the similarity between two values $x_j, y_j \in D_j$ is:*

$$Sim_j(x_j, y_j) = \bigwedge_{1 \leq i \leq m} (h_i(x_j) \leftrightarrow h_i(y_j)) \tag{4.6}$$

where $\{h_1, \ldots, h_m\}$ is a set of comparison criteria with $h_i : D_j^i \rightarrow [0, 1]$ and $\leftrightarrow$ is an equivalence operator. Concrete criteria functions are given in section 5.4.1.3 and $1- \mid h(x_j) - h(y_j) \mid$ is used as the equivalence operator (since this is a straightforward measure of the absolute distance between two points).

Consider the example of colours in order to illustrate the modeling of similarity in a given domain. $D_{colours} = \{yellow, violet, magenta, green, cyan, red, \ldots\}$. In order to model how ‘similar’ two given colours are, different perceptive criteria can be considered. For instance, there are ‘warm’ colours and ‘cold’ colours. With respect to this criterion, *yellow* and *orange* are more similar that *yellow* and *violet*. Related to the ‘warmness’ of colours, Newton (Newton 1972) established in 1666 the proportionality factors between colours that determine which should be the size of painted surfaces in order to be in perceptual equilibrium. For instance, yellow has luminosity 9 and violet luminosity 3. This means that if we paint two squares, one in yellow and one in violet, their surfaces have to be in relation 1 to 3 in order for the result to be in ‘equilibrium’, that is, the yellow square must be one third of the violet square in size. Another relevant perceptual criterion of colours is their visibility. There are various physiological characteristics of the human visual field, distribution of cones and rods, that ensure some colours are better perceived when moving away than others (Marr 1982). Green is the colour with the worst visibility and yellow and cyan are those with the best visibility. Other criteria like *memory* or *dynamicity* have also been studied. These criteria can then be used to model the colour example as (functions are presented extensively as sets of pairs (input, output)):

$$h_t = \{(yellow, 0.9), (violet, 0.1), (magenta, 0.1), green, 0.3), (cyan, 0.2), (red, 0.7), \ldots\}$$

$$h_l = \{(yellow, 0.9), (violet, 0.3), (magenta, 0.6), green, 0.6), (cyan, 0.4), (red, 0.8), \ldots\}$$

$$h_v = \{(yellow, 1), (violet, 0.5), (magenta, 0.4), green, 0.1), (cyan, 1), (red, 0.2), \ldots\}$$

where h_t, h_l and h_v are the comparison functions corresponding to temperature (warm is 1, cold is 0), luminosity (maximum is 1, minimum 0) and visibility (again maximum is 1 and minimum 0) respectively. With these functions and using min as conjunction, the following can be obtained through simple arithmetic:

$$\begin{aligned}&Sim_{colour}(yellow, green) = \\ &\quad min(1- \mid h_t(yellow) - h_t(green) \mid, 1- \mid h_l(yellow) - h_l(green) \mid, 1- \mid h_v(yellow) - h_v(green) \mid) \\ &\quad = min(0.4, 0.7, 0.1) = 0.1\end{aligned}$$

or,

$$Sim_{colour}(cyan, violet) = min(0.9, 0.9, 0.5) = 0.5$$

4.5.2.3 The Trade-off Algorithm

The trade-off algorithm performs an iterated hill-climbing search in a landscape of subset of the possible contracts. The search proceeds by successively generating contracts that lie closer to the iso-curve (representing the agent's aspiration level), followed by the selection of the contract that maximizes the similarity to the opponent's last offering. The algorithm terminates when the last selected contract lies on the iso-curve.

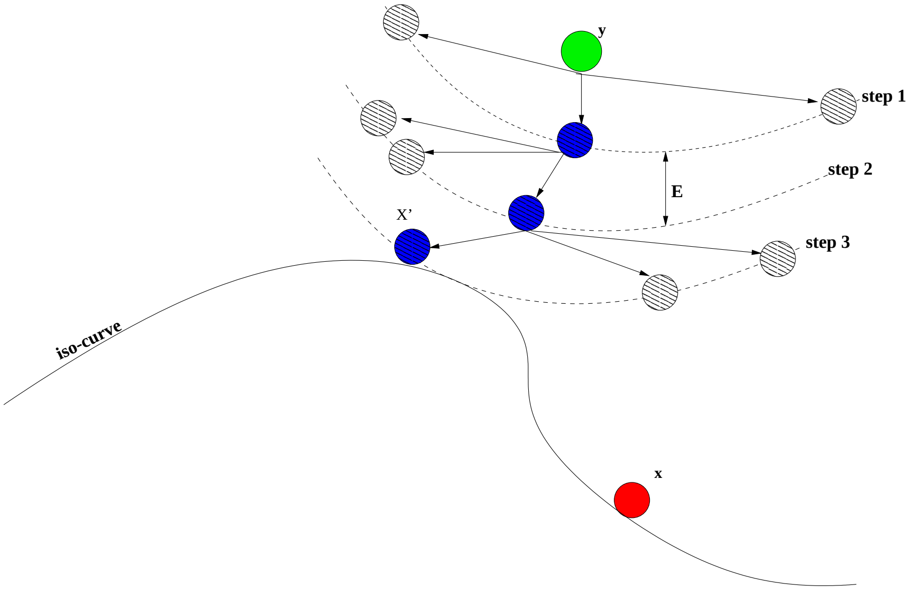

Figure 4.8: Schema of the trade-off algorithm with $N = 3$ and $S = 3$.

The algorithm, shown schematically in figure 4.8, starts at the contract y, the opponent's last offer, and moves towards the iso-curve (the solid line marked *iso-curve* in figure 4.8) associated with the agent's last offer, x. This approach to the iso-curve containing contract x is performed sequentially in S steps (three in figure 4.8). Each step starts by randomly generating N new contracts (three, one filled and two patterned ovals in figure 4.8) that have a utility E greater than the contract selected in the last step y^j (or $y^0 = y$ if it is the first step). N is referred to as the number of children. Each new contract y^{j+1} so generated satisfies $v(y^{j+1}) = v(y^j) + E$, and they all have the same utility to the agent (shown as the dotted line connecting all the children at each step). From the generated children contracts, the one that maximizes the similarity with respect to the opponent's contract y is selected (shown as the filled oval that becomes the parent of the next set of children in figure 4.8). E is computed as the overall difference between the value of x and y divided by the number of steps. That is, $E = \frac{v(x)-v(y)}{S}$. The overall effect of the algorithm is to sequentially explore a subset of the possible space of contracts and select for the next step the one that maximizes the similarity with respect to the other agent's contract offer.

Presented below is the algorithm responsible for generating a new random contract. This algorithm will thus be invoked N times at each step in order to compute the best trade-off contract (giving SN calls in total). The algorithm generates children by splitting the step gain in utility, E, randomly among the set of issues under negotiation.

This algorithm shows only the computations involved in making a single step, of size E in figure 4.8, towards the iso-curve specified by x. It functions as follows. Firstly, the maximum utility that can be gained for each issue is computed as the difference between the full aspiration of the agent's preferences and the utility of the contract that is being modified y (line 1). Note, at the first step of the algorithm iteration y will be the opponent's offered contract. In subsequent iterations the contract will be a sibiling of y. Each *weighted* individual utility gain is then summed to determine the overall weighted amount of utility that can be gained (line 2). Next, because the "consumption" of this utility gain has a random element (line 5), a degree of tolerance is set to allow for a degree of flexibility for the processes (steps between lines 4 and 7) that may "overstep" the iso-curve (line 3). The process of consumption of the available utility (computed in line 2) begins by allowing each issue to consume a random amount (line 5) between 0 and the limits computed in line 1. The store of the total amount consumed by each issue E_n is then updated as the addition of the old store and a linear weighted sum of each of the individually consumed utilities (line 6). The total amount that can be consumed is then recomputed given the newly consumed amount (line 7). If the amount consumed is less than the total amount E the process of consumption continues until the maximum (E or the step size in figure 4.8) is reached. The utility gain of each issue is then normalized to 1 once the issues have consumed all of the step utility gain E (line 8). Finally, the utility gained by each issue is remapped to actual values that correspond to the new utility (line 9).

```
inputs: $y^j$;                 /* last step best contract. $y^0 = y$ */
        $E$;                   /* step utility increase */
        $v()$;                 /* value scoring function */
output: $y^{j+1}$;             /* child of $y^j$ */
begin
(1)  $\overline{E}_i := 1 - v(y_i^j)$;               /* compute the maximum utility gain foreach issue*/
(2)  $E_{max} := \sum w_i \overline{E}_i$;           /* compute the total maximum utility gain*/
(3)  $\delta = 0.01 E_{max}$                         /* compute the average number of iterations*/
     if $(E_{max} > E + \delta)$ then
     begin
(4)    $k := 0; E_n := 0$;                           /* initialize number of steps and utility gain counters */
       while $(E_n < E)$ do
            $k := k + 1$;
(5)         $r_i^k := random(0, \overline{E}_i)$;     /* randomize utility gain for each issue */
(6)         $E_n := E_n + \sum_i w_i r_i^k$;          /* update utility gained in iteration $k$ */
(7)         $\overline{E}_i := \overline{E}_i - r_i^k$;  /* compute potential utility gain for next iteration */
       endwhile
(8)    $E_i := \left(\sum_{j=1}^{k} r_i^j\right) \frac{E}{E_n}$;   /* normalize the gains*/
(9)    $y_i^{j+1} := v_i^{-1}\left(v_i(y_i^j) + E_i\right)$;      /* compute value for each issue in new contract */
     end
     else $raise\ error$
end
```

Figure 4.9: The Trade-Off Algorithm

4.5.2.4 Algorithmic Complexity

When analysing the complexity of the trade-off algorithm the first thing to note is that it includes a call to a random number generator inside the main loop (step 5). This has a direct impact on the number of iterations, and hence on the time the algorithm will take. Assuming the random number generator is probabilistic in nature, a 'big-O' analysis of the complexity cannot be made (Aho, Hopcroft, & Ullman 1985). However, what can be computed is an "average case" assuming that the random generator is perfect.

Let n be the number of negotiation issues. Steps 1, 5, 6, 7, 8, and 9 all need a time which is $O(n)$ $(1 \leq i \leq n)$. The time used by the algorithm will be proportional then to the number of iterations, k, of the while loop, multiplied by the cost of each iteration (which, as said, is $O(n)$). That is, it will be proportional to kn. The possible magnitude of k is derived next. The while loop will terminate when E_n becomes bigger than E. It is known that before entering the loop for the first time $E_{max} = \sum_i \omega_i \overline{E}_i$ and $E_{max} > E+\delta$. E_n is the weighted addition of the portions r_i^k generated by each iteration. On average, and assuming perfect random number generation, at every iteration E_n will be incremented by half of each issue's maximum potential utility gain given to the random generator, that is, $\sum_i \frac{\overline{E}_i}{2}$. Thus, in the first iteration, the algorithm will consume a half of E_{max}, i.e. $E_n = 0 + \sum_i \omega_i \frac{\overline{E}_i}{2}$ which is $\frac{E_{max}}{2}$. In the second, a half of the remaining amount, that is a half of $\frac{E_{max}}{2}$, i.e. $\frac{E_{max}}{4}$. In general, the algorithm consumes $\frac{E_{max}}{2^k}$ at step k and leaves $\frac{E_{max}}{2^k}$ for the next step. That is, E_n at step k is $E_n = E_{max} - \frac{E_{max}}{2^k}$. The average value for k can then be computed as a function of the difference between E_{max} and E. Given that the algorithm stops when $E_n > E$, have $E_{max} - \frac{E_{max}}{2^k} > E$, that is, $E_{max} - E > \frac{E_{max}}{2^k}$. The step before had $\frac{E_{max}}{2^{k-1}} > E_{max} - E$. Taking this latter inequality, it is easy to see that $k < 1 + log\frac{E_{max}}{E_{max}-E}$. As $E_{max} - E > \delta$ is considered to be true, then $k < 1 + log\frac{E_{max}}{\delta}$. A policy to decide which value to assign to δ could be to fix its value as a percentage of E_{max}. For instance, making δ a 1% of E_{max} would mean that $k < 1 + log\frac{E_{max}}{0.01E_{max}}$, that is $k < 1 + log100 < 8$; eight iterations on average. Summarizing, if δ is fixed as a percentage c of E_{max}, it can be see that the average number of iterations is $k = 1 + log\frac{1}{c}$. Thus, on average the total time of the algorithm is proportional to $(1 + log\frac{1}{c})n$.

Thus, the average time the algorithm takes to complete is linear with respect to the number of issues in the negotiation. This linearity is a desirable property of the algorithm considering one of the aims of this research has been to develop decision mechanisms that respect the computational limitations of the agents. The trade-off mechanism can grow in complexity, although only linearly, with growing number of issues. However, an agent can reason explicitly about the time costs of engaging in trade-off negotiation given knowledge of the above analysis that the complexity grows linearly with the number of issues. Therefore, as complexity grows then agents can reason about what course of action to take. For example, if during the negotiation the number of issues grows to such an extent that the trade-off computation becomes too costly, then an agent wanting to implement a trade-off may use the issue-manipulation mechanism to remove some issues. This reduces the costs involved in the trade-off deliberation. Generally, the complexity levels of the trade-off algorithm can be used as triggers for initiating issue-manipulation mechanism that may help reduce the complexity of the trade-off algorithm. This decision can be made by the meta-strategy component of the agent architecture (section 4.7).

4.6 The Issue Set Manipulation Mechanism

The other deliberation mechanism is the issue set manipulation. One motivation behind the design of this mechanism has been the need to escape the problem of local minima in the social welfare function. This can be achieved through restructuring the problem. Recall that a local minima in the social welfare function refers to the negotiation context where the utility of the exchanged contracts is the same as the previous step—the agents are exchanging the same contracts, hence the joint utility of the possible deal given the exchanged contract, or the social welfare function, is constant.

At other times it is not the need to escape local minima that motivates modification of the issues involved in negotiation, but rather agents preferences over dimensions of services that can be *substituted*, *removed* or *added to*. Note that whereas the trade-off mechanism operates over the *complementary* dimensions of a service, the issue-set manipulation operates over the dimensions of a service that are *modifiable* (Topkis 1988). For example, in the telecommunication scenario (section 1.4.2), agents negotiate over a static set of issues, informally defined as core issues. However, the negotiation between *SPAs* and *NPAs* additionally consists of offers over non-core issues. For example, a *SPA* may begin QoS negotiation with a *NPA* specifying only $Bandwidth$. However, subsequently *NPA* may decide to include into the QoS negotiation a $packetloss$ issue with a high value if *SPA* has demanded a high capacity $Bandwidth$. Alternatively, *SPA* may decide to remove the $Bandwidth$ issue from the QoS negotiation with *NPA* if *IPCA* has changed its demand from a high quality video service to a standard audio service. Similarly, as shown in the example of agreement over the firing of the police commissioner by both the police office union and city hall (section 4.2.2), issues can also be removed when agents agree to their resolution.

4.6.1 Issue Manipulation Evaluation

The evaluation of a contract from the perspective of the issue manipulation mechanism is defined as:

$$I_a^{\text{issue-manipulate}}(t, x_{b\rightarrow a}^t) = \begin{cases} withdraw(a,b) & \text{If } t > t_{max}^a \\ accept(a,b,x_{b\rightarrow a}^t) & \text{If } V^a(S_{b\rightarrow a}^t) \geq V^a(S_{a\rightarrow b}^{t'}) \\ newset(a,b,S) & otherwise \end{cases}$$

where the content of the primitive *newset* (S in figure 4.2) is computed by the functions given in equations 4.7 that expand or equations 4.8 and 4.9 that reduce the set of negotiation issues (section 4.6.2). Note the similarity between this evaluation and the responsive (section 4.4.1) and trade-off mechanism's (section

4.5.1) evaluation functions. It terminates successfully if the utility of the new set of issues (and their corresponding values) is greater than the newset the agent is about to offer.

4.6.2 Issue Set Manipulation: A Formal Model

Negotiation processes are directed and centered around the resolution of conflicts over a set of issues J. This set may consist of one or more issues (distributed and integrative bargaining respectively). For simplification, the ontology of the set of possible negotiation issues, J, is assumed to be shared knowledge amongst the agents. It is further assumed that agents begin negotiation with a pre-specified set of "core" issues, $J^{core} \subseteq J$, and possibly other mutually agreed non-core set members, $J^{\neg core} \subseteq J$. Alterations to J^{core} are not permitted since some features such as the $Price$ of services are mandatory. However, elements of $J^{\neg core}$ can be altered dynamically. Agents can add or remove issues into $J^{\neg core}$ as they search for new possible, and up to now unconsidered, solutions.

If J^t is the set of issues being used at time t (where $J^t = \{j_1, \ldots, j_n\}$), $J - J^t$ is the set of issues not being used at time t, and $x^t_{a \to b} = (x[j_1], \ldots, x[j_n])$ is $a's$ current offer to b at time t, then issue set manipulation is defined through two operators: add and $remove$.

The add operator assists the agent in selecting an issue j' from $J - J^t$, and an associated value $x[j']$, that gives the highest score to the agent.

Definition 12 *The best issue to add to the set J^t is defined as:*

$$add(J^t) = \arg\max_{j \in J - J^t} \{ \max_{x[j] \in D^i_j} V^a(x^t \bullet x[j]) \} \tag{4.7}$$

where $\bullet$ stands for concatenation.

An issue's score evaluation is also used to define the $remove$ operator in a similar fashion. This operator assists the agent in selecting the best issue to remove from the current negotiation set J^t.

Definition 13 *The best issue to remove from the set J^t (from $a's$ perspective), is defined as:*

$$remove(J^t) = \arg\max_{j_i \in J^t - J^{core}} \{V^a(x)\} \tag{4.8}$$

with $x = (x^t[j_1], \ldots, x^t[j_{i-1}], x^t[j_{i+1}], x^t[j_n])$

The $remove$ operator can also be defined in terms of the aforementioned similarity function (section 4.5.2.2). This type of similarity-based $remove$ operator selects from two given offers x, from agent a to b, and y, from agent b to a, which issue to remove in order to maximize the similarity between x and y. Therefore, compared to the previous $remove$ operator, this mechanism can be considered as more cooperative:

Definition 14 *The best issue to remove from $a's$ perspective from the set J^t is defined as:*

$$remove(J^t) = \arg \max_{j_i \in J^t - J^{core}} \{sim(x', y')\} \tag{4.9}$$

with $x' = (x[j_1], \ldots, x[j_{i-1}], x[j_{i+1}], \ldots, x[j_n])$*, and* $y' = (y[j_1], \ldots, y[j_{i-1}], y[j_{i+1}], \ldots, y[j_n])$

It is not possible to define a similarity-based *add* operator since the introduction of an issue does not permit an agent to make comparisons with the opponent's last offer (simply because there is no value offered over that issue).

Another computational requirement of these mechanisms is the need for an agent to dynamically re-compute the issue weights. The re-computation of weights is defined by first specifying the importance of the added issue, I_j, *with respect* to the average importance of other issues. That is, the weight the new issue should have in the set of issues with respect to the weight of the other issues— $I_j = w_j / (\sum_{i \in J} w_i / n)$, where n is the new number of issues.Then:

Definition 15 *The weight of added issue j, w_j, is defined as:*

$$w_j = \frac{I_j}{(n-1) + I_j}$$
$$w'_i = (1 - w_j) w_i \quad \forall i \in \{i_1, \ldots, i_n\}, i \neq j$$

where w_j is the importance of the issue j, n is the new number of issues, w_i is the old weight for issue i and w'_i is its new weight after the inclusion of issue j. Thus computation of w'_i attempts to "fit" in the weights of other issues within the "space left over" when the new issue has been included.

Re-computation of weights when an issue is removed in turn is defined simply as re-normalizing the remaining weights:

Definition 16 *The weight of the remaining issues i after an issue j has been removed is defined as:*

$$w'_i = \frac{1}{1 - w_j} w_i$$

Agents deliberate over how to combine these *add* and $remove$ operators in a manner that maximizes some measure such as the contract score. However, a search of the tree of possible operators to find the optimum set of issues may be computationally expensive because the size of the search tree can grow to combinatorially large sizes. This problem is not addressed in this thesis and is postponed for future work by implementing anytime algorithms that produce closer to optimal search results when given increasingly more time, but nonetheless produce, possibly sub-optimal, results when they are stopped anytime (Aho, Hopcroft, & Ullman 1985). Then given these algorithms and the negotiation time limits it is possible to compute a, possibly sub-optimal, solution that increases some measure such as the contract score or social welfare.

4.7 The Meta Strategy Mechanism

The fact that there are three potential choices of mechanisms to use for generating a proposal poses another decision problem for the agent, namely which to use. This decision is referred to as the *meta-strategy* of the agent since the process involves making decisions about which of the decisions should be selected for the generation of the proposal. Recall the argument from section 2.2.8 for the need to develop not only computationally tractable search algorithms that can traverse problem state-spaces that may be deep with wide branching factors (figure 2.3) and can operate under strict time limits, but also the need for reasoning mechanisms about these different algorithms. This meta reasoning is needed because each algorithm carries different costs and benefits.

Another role of a meta strategy in negotiation, apart from a cost and benefit analysis of each mechanism in a given environment, can be described through an example that shows different “negotiation dances” (Raiffa 1982) implemented by the responsive and trade-off mechanisms (figure 4.10). Issue manipulation dynamics are not represented since the behaviour of this mechanism is to alter the space of possible deals. The filled ovals are the values of the offered contracts from agent 1 to agent 2 from agent 1’s perspective, and the unfilled ovals represent the converse, the value of the offered contracts from agent 2 to agent 1 from agent 2’s perspective. The filled oval at $(0.5, 0.5)$ represents the *reference* point (section 3.1.4).

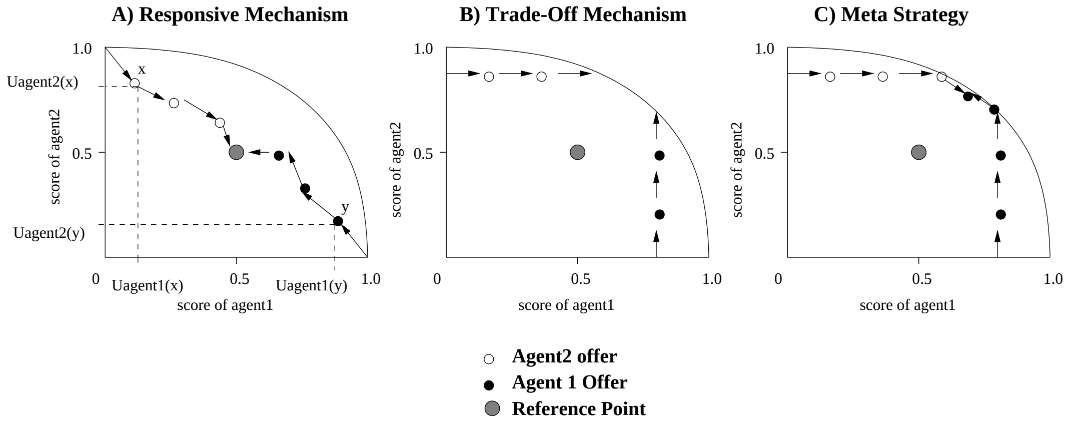

Figure 4.10: Negotiation Dances.

Figure 4.10 A represents one hypothetical execution trace where both agents generate contracts with the responsive mechanism. Each offer has lower utility for the agent who makes the offer, but relatively more utility for the other (movement towards the *reference* point). This process continues until the second condition of the responsive evaluation function (section 4.4.1) of one of the agents is satisfied $(V^a(x^t_{b\to a}) \geq V^a(x^{t'}_{a\to b}))$—referred to as the cross-over in utilities earlier. The responsive mechanism can select different outcomes based on the rate of concession adopted for each issue (the angle of approach to the *reference*

point in figure 4.10 A). Although in figure 4.10 A this final outcome is hypothetically represented as the *reference* point, it will be concretely shown in the next chapter that this is not necessarily the case if each agent assigns a different rate of concession according to the weight of the issues involved—responsive mechanisms *can* also reach better deals than *reference*.

Figure 4.10 B represents another hypothetical execution trace where both agents now generate contracts with the trade-off mechanism. Now each offer has the same utility for the agent who makes the offer, but relatively more utility for the other (movement towards the *pareto-optimal* line). The trade-off mechanism searches for outcomes that are of the same utility to the agent, but which *may* result in a higher utility for the opponent. This is schematically shown in figure 4.10 as a line of approach directed towards the pareto-optimal line. Once again, this is a simplification for purposes of the exposition—an offer generated by agent 1 may indeed have decreasing utility to agent 2 (arrow moving *away* from the pareto-optimal line) if the similarity function being used does not correctly induce the preferences of the other agent.

A meta strategy (figure 4.10 C) is then one that combines either "dance" towards an outcome. One rationale for the use of a meta-strategy mentioned above is reasoning about the costs and benefits of different search mechanisms. However, an additional rationale, observable from the example shown in figure 4.10 B, is to escape from the local minima of the social welfare function. If the social welfare function is taken to be the pareto-optimal line, which maximizes the sum of the individual utilities, then, because of the privacy of information (an important feature of many domains, section 1.4.3), agents can not make an interpersonal comparison of individual utilities in order to compute whether their offers do indeed lie on, or are approaching, the pareto optimal line which measures the global goodness of offers.[11] Given that the position of offers with respect to the pareto-optimal line can not be compared and the fact that the evaluation function of the trade-off mechanism (section 4.5.1) only terminates when the time runs out or there is a cross-over of utilities, then the agents enter a loop of exchanging the same contract with one another. That is they remain in a local minima. A solution is therefore needed to escape this local minima. Figure 4.10 C shows one such solution where the local minima is escaped by both agents switching to a responsive mechanism and conceding utility. This concession may, as shown in figure 4.10 C, indeed satisfy the second condition of the trade-off evaluation function where offers cross-over in utilities (thereby terminating the negotiation process). Alternatively, agents may resume implementing a trade-off algorithm until such a cross-over is eventually reached or time limits are passed. Alternatively, the meta-strategy may change the problem state-space by implementing the issue-manipulation mechanism which changes the set of possible outcomes through adding or removing issue(s).

[11] Indeed, another protocol may be to allow one agent to exchange points on its iso-curve and let the other agent select the one that maximizes its utility (Raiffa 1982). However, this protocol assumes agents will not only reveal their preferences, but will also do so honestly (assumptions which are not made in this thesis).

The above example shows how different combinations of mechanisms, by either both or the individual agents, leads to different final outcomes. For instance, a meta strategy which continuously switches between responsive and trade-off mechanisms creates a contract score trace that is similar to an ever decreasing step function. Conversely, a meta strategy that only permits the responsive mechanism to generate contracts results in a contract score trace which may (depending on the parameters of the responsive mechanism) decrease in a linear fashion. Note, that at the first time step in its negotiation an agent must choose the responsive mechanism. It then has a choice of other mechanisms in the course of negotiation. This is because the trade-off mechanism must have a previous contract to compute the iso-contract curve.

In general, the evaluation of which search should be implemented is delegated to a meta-level reasoner whose decisions can be based on factors such as the opponent's perceived strategy, the on-line cost of communication, the off-line cost of the search algorithm (or its path cost), the structure of the problem or the optimality of the search mechanism in terms of completeness (finding an agreement when one exists), the time and space complexity of the search mechanism, and the solution optimality of the mechanism when more than one agreement is feasible. A formal treatment of a meta-strategy is postponed for future work. However, the contributions of this work with respect to the meta-strategy are the identification of the computational role and rationale of meta-strategies in the dynamics of negotiation processes that often involve uncertainties and computational boundedness. Furthermore, the role and effect of candidate meta-strategies are also empirically analyzed in the next chapter.

4.8 Summary

A formal decision architecture of the wrapper framework and two protocols of interactions were presented in this chapter. The decision architecture is based on three mechanisms: responsive, trade-off and issue set manipulation. The rationale for their design was provided in terms of computational, information and motivational states of an agent. The responsive mechanism is computationally simple and requires only minimal information about the state of the other agent. An agent that implements a responsive strategy is motivated by pressing environmental needs to terminate negotiation and reach an agreement that has lower social welfare or joint utility. Conversely, deliberative mechanisms (trade-off and issue set manipulation) may increase the social welfare—hence an agent that implements a deliberation mechanism is said to be motivated by concern for social welfare. However, these mechanisms are computationally more complex and their operations require more information about their opponent.

The next chapter empirically analyses the behaviour of a number of concrete agent architectures that directly follow from the presented generic model. The aim of these experiments is to test the behaviour of the responsive and trade-off mechanisms in a number of different environments. Empirical analysis of the issue set manipulation mechanism is deferred to future work, since algorithms must first be designed.

Chapter 5

Empirical Evaluation

This chapter is a description of the evaluation phase of the research. The model presented in the previous chapter defines and formalizes a range of negotiation behaviours which can be implemented by the wrapper. However, which of these behaviours will be successful in which negotiation contexts cannot be predicted from the theoretical model alone. This is because: a) the developed model only specifies a negotiation framework that can be "tuned" to the needs of a negotiating agent designer, b) there are a large number of interrelated variables within the wrapper and a broad range of situations that need to be considered, and c) some parts of the model are heuristic in nature (for example, a meta-strategy that engages in trade-off mechanism always until a local minimum in the social welfare function is detected is a decision heuristic whose efficacy across different types of environments can not be determined a priori; see section 3.3). The designer who uses the wrapper needs additional information about the interaction profiles of the components of the wrapper and it is the "tuning" of these profiles which produces the results. Therefore the approach adopted in this research has been to empirically evaluate representative components of the wrapper with the final aim of determining the most successful behaviours in various types of situations. The experiments reported here are exploratory studies (Cohen 1995). In such studies, *general* hypothesis are formed that state the underlying intuitions about causal factors. Experiments are then conducted by creating a simulation "laboratory" that generates data, the observation of which either supports or refutes these general hypothesis. Manipulation studies, on the other hand, are more specific and investigate the system via *detailed* causal hypothesis. As Cohen notes, exploratory experiments help us to *"find needles in the haystack, whereas manipulation experiments put the needles under the microscope, and tell us whether they are needles and whether they are sharp"* (Cohen 1995), p.6.

5.1 The Experiment Set

Three sets of experiments are reported in this chapter. One set relates to the empirical evaluation of the responsive mechanism of the wrapper (sections 5.3, and 5.4), other to the trade-off mechanism (section

5.5) and final one to the meta-strategy mechanism (section 5.6). For the reasons outlined in section 4.8, the issue-manipulation mechanism is currently excluded from the analysis. The responsive experiments are divided into two complementary sections. In the first section (section 5.3) the investigation is focused on determining the behaviour and inter-dependencies of the responsive model's basic constituent elements, namely tactical decision making. This analysis will then lay the foundation for subsequent experimental work reported in section 5.4 which investigates strategic decision making. Throughout this chapter the former experiments will be referred to as either *non-strategic* or *pure-strategy* experiments because tactics are assigned a binary weight value for γ_{ij} of either 0 or 1, and this value is static throughout the negotiation thread. Alternatively, the latter experiments will be referred to as *strategic*, since the tactics' weights can be assigned any value in the interval $[0, 1]$. Strategic experiments are further subdivided into *static strategy* and *dynamic strategy* experiments, for experiments where the weight of a tactic is static throughout the negotiation or dynamically modified in the course of negotiation, respectively. Section 5.5 reports on the experimental procedure and outcomes of the empirical evaluation of the trade-off mechanism. Finally, section 5.6 details the empirical evaluation of the meta strategy mechanism.

Before this, however, the next section discusses the foundational principles of the design of the experiments.

5.2 Experimental Design Principles

A negotiation context can involve many issues and parties with different agent aspiration levels and time limits. To handle this environmental complexity experimental design consideration, together with a number of simplifying assumptions, are necessary for empirical analysis of the negotiation model that is embedded in such a complex environment. Experimental design principles define and categorize the variables of the "laboratory". These design principles are expanded on in this section.

Experimental variables can either be *independent* or *dependent* (Cohen 1995). Independent variables are defined as those variables whose values are under the control of the experimenter. Dependent variables, in turn, are defined as those variables whose values are not under the control of the experimenter. Instead, the values of these are observed by the experimenter as measurements. The type of either of these variables must be one of the following: i) categorical, ii) ordinal or iii) interval (Cohen 1995). With categorical variables, the measurement (for dependent variables) or assignment process (for independent variables) designates a category label to the variable. For example, the categorical dependent variable $outcome$ can be assigned a value $Accept$ or $Withdraw$ after making a measurement. Ordinal variables, on the other hand, can be ranked, but the distances between these points are meaningless. For example, the time deadline of negotiation for the experiments, t^a_{max}, is designed as an ordinal independent variable which can be assigned values $long$, $medium$ and $short$ term. Distances between ordinal scales are meaningless (it can not be said that the difference between $long$ and $short$ is equal to $medium$). Finally, with interval (or ratio) scales both the distances between variable points *and* the ratios between data sets are meaningful. For example, distances in the amount of utility a mechanism procures for an agent can be compared not only in a single trial but also across trials. A condition for ratio scale parameters is that the zero point is known.

Variables can also be transformed by mapping from one scale into another. Mapping information from one scale into another enables i) analysis of the *types* of environments and ii) statistical operations that were previously inaccessible (see description below for examples). Transformation of scale is useful because it can be used as a data abstraction tool since it allows analysis of *groups*, or *types*, of environments rather than individual, concrete environments. For example, transformation of negotiation deadlines from an interval scale into a ranked ordinal scale is an abstraction tool that ignores the actual differences within and across the groups of variables $long$, $medium$ and $short$ term deadlines and instead emphasizes the differences in rankings. Members that have a $long$ term negotiation deadline have values for t_{max} that are higher than $short$ term members. Nothing is said about their magnitudes.

5.3 Non-Strategic Experiments

The aim of this set of experiments is to investigate the behaviour of individual tactics (non-strategic) for decision making in a number of environments. A knowledge of how individually different pure tactics behave in different environments can then be captured as decision guidelines for the responsive strategic decision making component of the wrapper.

The experiments involve selecting a particular tactic, generating a range of random environments, then allowing the agent to negotiate using the chosen tactic against an opponent who employs a range of other tactics. Various experimental measures related to the negotiations are then recorded. In particular, section 5.3.1 defines the experimental environments and the tactics, section 5.3.3 describes the experimental measures, section 5.3.2 defines the experimental procedure, section 5.3.4 describes the experimental hypotheses and discusses the results, and finally section 5.3.4.4 summarizes the results and conclusions reached.

5.3.1 Experimental Independent Variables

The experimental independent variables are discussed in this section. In pure-strategy experiments, independent variables are defined in terms of i) environments of negotiation (section 5.3.1.1) and ii) the tactics available for decision making (section 5.3.1.2). The complete set of independent variables is shown in figure 5.1. The assignment of values to independent variables is under the control of the experimenter who is constrained by limiting the complexity of analysis. The variable scale denotes the type of the variable (either categorical or interval), variable range denotes the set of possible values available which can be assigned to the variable and variable transformation denotes the mapping from one scale to another.

Variable Name	*Variable Scale*	*Variable Ranges*	*Variable Transformation*
$Agent$	categorical	$\{2, \infty\}$	categorical=$\{$player,opponent$\}$
$\{J\}$	categorical	$\{1, \infty\}$	categorical=$\{$price$\}$
w_j^a	categorical	$[0, 1]$	categorical=1
κ^a	interval	$[0, 1]$	ordinal=$\{$high,low$\}$
$[min_j^a, max_j^a]$	interval	$[[0, \infty], [0, \infty]]$	ordinal=$\{$full-overlap,no-overlap$\}$
t_{max}^a	interval	$[1, \infty]$	ordinal=$\{$large,low$\}$
$Tactics$	categorical	$\{time, resource, behaviour\}$	ordinal=$\{$boulware,linear,conceder,impatient,steady,patient,relativetitfortat,randomtitfortat,averagetitfortat$\}$

Figure 5.1: Pure Strategy Experimental Independent Variables

5.3.1.1 Environments

Environments, in these experiments, are characterized by the number of agents they contain, the issues which are being discussed, the deadlines by when agreements must be reached and the expectations of the agents. Since there are infinitely many potential environments (infinite number of agents and issues), selecting a representative and finite subset of environments is necessary to find a means of assessing an agent's negotiation performance. To this end, experiments are conducted between only two agents, categorically labelled as $client$ and $server$, negotiating over only a single issue, *price*. The last simplification is relaxed in the next set of experiments where agents negotiate over a number of issues. Since there is only one issue, its weight (w_j^a) can only be assigned the value of 1. The position of the initial offer on the reservation values (κ^a, section 4.4.2.1) is transformed from an interval independent variable to an ordinal scale of high and low initial offers (see section 5.3.4.3 for details of the transformation).

The negotiation interval, $[min_j^a, max_j^a]$, is also an interval valued independent variable whose scale is infinite. To overcome this problem, an agent's reservation values are transformed to an ordinal scale whose actual scale is computed as follows. The difference between the agent's minimum and maximum values, for price, is computed using two variables: θ^a (the length of the reservation interval for an agent a) and Φ (the degree of intersection between the reservation intervals of the two agents; ranging between 0 for full overlap and 0.99 for virtually no overlap). In this case, for each environment, the independent variable min_{price}^c is assigned value 10 ($min_{price}^c = 10$), Φ is set to 0 ($\Phi = 0$), θ^a is randomly selected between the ranges of $\{10, 30\}$ for both agents, and the negotiation intervals are computed as $max^c = min^c + \theta^c; min^s = \theta^c \Phi + min^c; max^s = min^s + \theta^s$. Note, these values for computing the interval lengths of the interval value are chosen arbitrarily because the scoring function of the offers models the ordinal and not the cardinal relationships between the reservation values.[1]

The independent variable t_{max}^a, which assigns the negotiation deadline of the experiments for each agent, is transformed from the interval to an ordinal scale of $short$ and $long$ term deadlines. This transfor-

[1] Note the server's minimum reservation value is never lower than the client's minimum. This is because degenerate negotiations in which offers are immediately accepted are not interesting. This method of generating reservation values also means a deal is always possible since there is always some degree of overlap.

Tactic Family	*Tactic Name*	*Abbreviation*	*Tactic Ranges*	*Description*
Time-dependent	Boulware	B	$\beta \in \{0.01, 0.2\}$	Increased rate of approach to reservation as β increases
Time-dependent	Linear	L	$\beta = 1.0$	
Time-dependent	Conceder	C	$\beta \in \{20.0, 40.0\}$	
Resource-dependent	Impatient	IM	$\mu = 1, \mathrm{n} = 1$	Decreasing rate of approach to reservation as μ increases
Resource-dependent	Steady	ST	$\mu \in \{1, 5\}, \mathrm{n} = 1$	
Resource-dependent	Patient	PA	$\mu \in \{5, 10\}, \mathrm{n} = 1$	
Behaviour-dependent	Relative tit for tat	RE	$\delta = 1$	Percentage imitation of last two offers
Behaviour-dependent	Random tit for tat	RA	$\delta = 1 \ \mathrm{m} \in \{1, 3\}$	Fluctuating absolute imitation of last two offers
Behaviour-dependent	Average tit for tat	AV	$\gamma = 2$	Average imitation of last four offers

Figure 5.2: Experimental Tactic Key

mation facilitates the analysis of outcomes in *groups* of deadlines, ignoring the differences within a group and emphasizing the differences across the groups. The group $long$ term deadlines is defined as samples within the values of $30 - 60$ ticks of a discrete clock. $Short$ term deadlines are defined as samples within values $2 - 10$ ticks of a discrete clock.

Given this situation, the experimental environment is uniquely defined by the following variables: $[t^c_{max}, t^s_{max}, \kappa^c, \kappa^s, min^c_{price}, max^c_{price}, min^s_{price}, max^s_{price}]$.

5.3.1.2 Tactics

The second simplification involves selecting a finite range of tactics, since the model allows for an infinite set (e.g the range of β is infinite which means there are infinitely many time dependent tactics). For analytical tractability, the tactics are divided into nine groups (see figure 5.2); three each from the time, resource and behaviour dependent families. An equal number for each family is chosen to ensure the results are not skewed by having more encounters with a particular type of tactic. The three members of the time-dependent family are chosen to correspond to behaviours that concede in time in a boulware, linear and conceder fashion. These categories of behaviours are chosen since they represent extreme behaviours ($boulware$ and $conceder$) as well as an in-between control rate ($linear$) which concedes linearly. These categories of time-dependent tactics correspond to the transformation of interval values for β into the ordinal scale $0.01 - 0.2$ for the $boulware$ category, 1.0 for the $linear$ category and $20 - 40$ for the $conceder$ category. The three members of the resource-dependent family are also chosen that correspond to a decreasing rate of concession as the rate of resources used increases. These categories of resource-dependent tactics correspond to the transformation of interval values for μ into the ordinal scale 1 for the $impatient$ category, $\{1, 5\}$ for the $steady$ category and $\{5, 10\}$ for the $patient$ category. Finally, the three members of the behaviour-dependent family are also chosen to correspond to the different types of imitation according to the given sub-family parameters.

5.3.2 Experimental Procedure

The experimental procedure consists of sampling each tactic group for *every* environment since the subject of interest is the behaviour of tactic families rather than single, concrete tactics. For each environment e_k, k indexes the environments, two matrices are defined to represent the outcomes of the client, $game_c^{e_k}$, and the server, $game_s^{e_k}$, when playing particular tactics. The client's tactics are indexed by the rows i and the server's by the columns j, so $game_c^{e_k}[i,j]$ is the outcome of the client when playing tactic i against a server playing tactic j. Each tactic plays against all other tactics in each environment, hence $1 \leq i, j \leq 9$.

To produce statistically meaningful results, the experimental measures described below are averaged over a number of environments and summed against all other tactics for each agent. Therefore this analysis is based on the performance of a tactic family across all other tactic families. The precise set of environments is sampled from the parameters specified in section 5.3.1 and the number of environments used is 200. This ensures that the probability of the sampled mean deviating by more than 0.01 from the true mean is less than 0.05. The experiments were written in Sicstus3.7.1 Prolog and ran on *SunOs 4.5* Unix machines.

5.3.3 Experimental Dependent Variables

To evaluate the effectiveness of the tactics, the following measures are considered which calibrate: i) the intrinsic benefit of the tactic family to an agent (section 5.3.3.1); ii) the cost adjusted benefit which moderates the intrinsic benefit with some measure of the cost involved in achieving that benefit (section 5.3.3.2) and iii) the performance of the intrinsic utility relative to a control condition (section 5.3.3.3).

5.3.3.1 Intrinsic Agent Utility

The intrinsic benefit is modeled as the agent's utility for the negotiation's final outcome, in a given environment, independently of the time taken and the resources consumed (Russell & Wefald 1991). This utility, $U_a^{e_k}$, is calculated for each agent for a price x using a linear scoring function:[2]

$$U_c^{e_k}(x) = \frac{max_{price}^c - x}{max_{price}^c - min_{price}^c} \qquad\qquad U_s^{e_k}(x) = \frac{x - min_{price}^s}{max_{price}^s - min_{price}^s}$$

If no deal is made in a particular negotiation, then the value zero (the conflict point, see section 3.1.4) is assigned to both $U_c^{e_k}$ and $U_s^{e_k}$. However, by defining the utilities in this manner no distinction can be made between deals made at reservations and no deals. Therefore in certain experiments the intrinsic utility is only computed for cases in which deals are made.

The outcome of the negotiations, as presented in the previous subsection, is represented in the matrix $game_a^{e_k}$. Hence the utility for a client c when negotiating using a tactic i against a server s using tactic j in environment e_k is $U_c^{e_k}(game_c^{e_k}[i,j])$.

[2] The simplicity of this utility function is acknowledged, but the intention here is to investigate the properties of the model and not the utility functions per se. The role of the utility function is evaluated in section 5.5

5.3.3.2 Cost Adjusted Benefit

In addition to knowing the intrinsic utility of a tactic to an agent, the relationship between an outcome's utility and the costs involved in achieving it is also useful information in making strategic or meta-strategic decisions about the costs of a given mechanism (see argument in section 4.7). The type of cost considered in these experiments is on-line, as opposed to off-line cost, because the former are more machine or resource independent than the latter. For example, calculating the off-line computational cost of a mechanism may require calibration of performance with respect to memory usage, speed and time which is machine architecture dependent. On-line costs, on the other hand, are not dependent on the architecture of the agent, but rather the load the agent's reasoning process places on the communication infrastructure.

The cost adjusted benefit (B) of tactic pairs i and j in environment e_k is defined as follows:

$$B_a^{e_k}[i,j] = U_a^{e_k}[i,j] - C_a^{e_k}[i,j]$$

To define the on-line cost function, C, the notion of a *system* is introduced. A system, in these experiments, is a set of resources that can be used by the agents during their negotiations. The usage of these resources is subject to a tax $\mathcal{T}$ which is levied on each message communicated between the agents. Therefore, the greater the communication between the agents, the greater the cost to the agents. So:

$$C_c^{e_k}[i,j] = C_s^{e_k}[i,j] = \tanh(|X_{c_i \leftrightarrow s_j}| * \mathcal{T})$$

where $|X_{c_i \leftrightarrow s_j}|$ is the length of the thread at the end of negotiation between a client using tactic i and a server using tactic j, $\tanh$ is an increasing function that maps the real numbers into $[0,1]$ and $\mathcal{T}$ determines the rate of change of $\tanh()$. $\mathcal{T}$ is sampled between the ranges of $[0.001, 0.1]$. In short, the greater the taxation system, the more costly the communication and the quicker the rate at which the cost rises to an agent for each message.

The system utility, on the other hand, is coarsely defined as the total number of messages in negotiation which indirectly measures the communication load the tactics incur at the agent level.

5.3.3.3 Experimental Controls

The control conditions for these experiments are based on the arguments from cooperative game theory, presented in section 3.1.4. The outcome attained by a pair of tactic families is compared with the regular Nash solution (equation 3.1 and figure 3.2 A, section 3.1.4), implemented by a protocol in which agents declare their true reservation prices (an incentive compatible and direct protocol, section 3.1.8) at the first step of negotiation and then share the overlap in the declared reservation values. This choice is both fair (i.e. is Nash) and pareto optimal (in that the outcome is beneficial to both agents and any deviation results in an increase in utility for one at the cost of a decrease in utility to the other). For example, consider a client agent c and a server agent s having price reservation values $[min^c_{price}, max^c_{price}]$ and $[min^s_{price}, max^s_{price}]$

respectively and $max^c_{price} \geq min^s_{price}$. The control outcome $\mathcal{O}$ for a given environment e_k is then defined as:

$$\mathcal{O}^{e_k} = \frac{max^c_{price} + min^s_{price}}{2}$$

Applying the definitions of utility presented earlier, the utility of the control game, $U^{e_k}_a(\mathcal{O}^{e_k})$, for agent a can then be computed. Given this, the comparative performance of agents using the responsive mechanism of the wrapper with respect to the one shot protocol, is defined as the difference between the intrinsic agent utility and the utility the agent would have received in the control protocol:

$$Gain^{e_k}_a[i,j] = U^{e_k}_a(game^{e_k}_a[i,j]) - U^{e_k}_a(\mathcal{O}^{e_k})$$

5.3.4 Hypotheses and Results

The experiments considered here relate to two main components of the negotiation model: i) the amount of time available to make an agreement, t^a_{max} and ii) the relative value of the initial offer, κ^a. These two factors are chosen because the parameters which influence the behaviour of the tactics (with the exception of resource-dependent tactics for N number of agents) are dependent on the available time limits and the initial offers, rather than the number of agents, the number of issues, their weights or their reservation values (note that these variables are constant in these experiments).

To test the effects of varying deadlines on agreements, the experiments are classified into environments where the time to reach an agreement is large (section 5.3.4.1) and those where it is small (section 5.3.4.2). Likewise for initial offers; there are environments in which the initial offer is near the minimum of the agent's reservation values and those where it is near the maximum (section 5.3.4.3). The reservation values are computed as described in section 5.3.1 with $\theta^c = \theta^s = 30$ and $\Phi = 0$ (refer to figure 5.2 for the key to the experimental tactics). Each abbreviation is further postfixed by the agent's role (e.g BC and BS denote a client and a server playing tactic B respectively).

5.3.4.1 Long Term Deadlines

The hypotheses about the effect of long term deadlines are:

> **Hypothesis 1:** *In environments where there is plenty of time for negotiation, tactics which slowly approach their reservation values will gain higher intrinsic utilities than those which have a quicker rate of approach. However, they will make fewer deals.*
>
> **Hypothesis 2:** *The utility to the system will be high when tactics have long deadlines since large numbers of offers will be exchanged. Consequently, there will be a large difference between a deal's intrinsic and cost adjusted utilities.*

Concrete values need to be provided for the experimental variables to evaluate these hypotheses. In this case, an environment with long term deadlines is defined as one in which the values of t^c_{max} and t^s_{max}

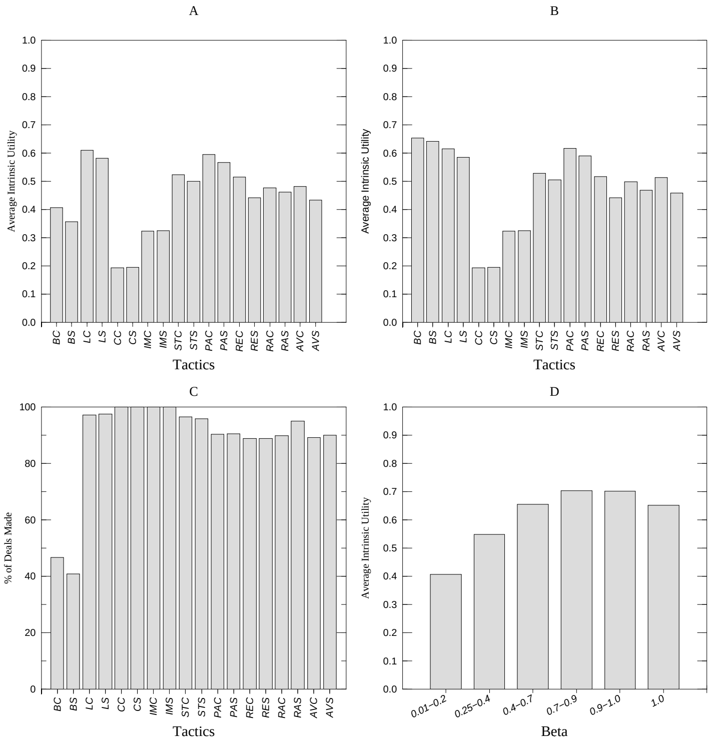

Figure 5.3: Average Intrinsic Utilities and Deals Made for Pure-Strategy Experiments in Long Term Deadlines: A) Average Intrinsic Utility For Both Deals And No Deals, B)Average Intrinsic Utility For Deals Only, C) Percentage of Deals Made, D) Average Intrinsic Utility For Both Deals and No Deals for Increasing Values of β.

are sampled within thirty and sixty ticks of a discrete clock. Note that $t^c_{max} \geq t^s_{max}$ and $t^c_{max} < t^s_{max}$ are permitted. Since high values of κ^a over-constrain the true behaviour of tactics, the value of κ is set to 0.1 for both agents. In each environment, the order of who begins the negotiation process is randomly selected.[3] Considering hypothesis 1 first. It was predicted that a tactic which approaches reservations at the slowest rate (i.e a Boulware) should attain the best deals. However, from figure 5.3.A the observation is that the most successful tactics are Linear, Patient and Steady. These tactics are characterized by the fact that they concede at a steady rate throughout the negotiation process. The next most successful group are the behaviour dependent tactics. Note, these imitative tactics never do better than other tactics; the best they

[3] The initiator of a bid is randomly chosen because in earlier experiments it was found that the agent which opens the negotiation fairs better, irrespective of whether the agent is a client or a server. This is because the agent who begins the negotiation round reaches $\alpha^a_{price} = 1$ *before* the other agent, hence deriving more intrinsic utility. See section 2.2.5 for further arguments concerning the (dis)advantages of the opening bid.

do is gain equal utility to the best tactic (Axelrod 1984). The worst performing tactics are Conceder and Impatient, both of which rapidly approach their reservation values.

The observation that Boulware tactics make significantly fewer deals than all the other tactic families (figure 5.3.C) helps explain Boulware's unexpectedly poor performance. Taking this into account, the average intrinsic utility for only those cases in which deals are made (figure 5.3.B) was examined. This shows that when Boulwares do make deals, they do indeed receive a high individual utility (as predicted).

It is hypothesized that the reason why Boulware tactics perform poorly is caused by the imitating responses of the behaviour dependent tactics, thereby effectively increasing the numbers of Boulwares in the population. To test this, the final average intrinsic utility for deals only of Boulware tactics is compared across: i) *all* other tactics and ii) all other tactics *apart* from behaviour dependent tactics. It is found that the success of Boulware tactics increased by 10% in the latter case.

From these observations, it can be concluded that the initial hypothesis does not hold because of the composition of the tactic population. It is predicted that in an environment in which there is plenty of time to reach a deal, Boulware should rank higher than tactics that approached reservation values quickly. However, for Boulwares to prosper in the experimental environment, they should adopt a value for β which is between 0.7 and 1.0 (figure 5.3.D).

Moving onto the second hypothesis. Figure 5.4.A confirms the results for the first part of this hypothesis; the tactic that uses the most system resource is Boulware and the least is Conceder. In addition, although Boulware tactics have higher intrinsic agent utilities than conciliatory tactics (Conceder and Impatient), when the the cost of communication is taken into consideration the converse is true (figures 5.4.B). This accords with the intuitions in the second part of hypothesis 2. The cost adjusted utilities of the remaining tactics are approximately similar. The reason for this is that cost adjusted benefit, which is the product of the intrinsic utility and a function of the number of exchanged messages, is sensitive to large fluctuations in the product and assigns similar utilities to non-extreme values.

Finally, it can be observed that the comparison of the tactics with respect to the controls follows the same broad pattern as the intrinsic agent utility (figure 5.4.C). Steadily conceding type tactics (Linear, Steady and Patient) on average perform better than the controls, the conciliatory types (Conceder and Impatient) perform worse. This is to be expected, since the closer the tactic's selected deal to the deal which is the mid-point of the reservation intersection (intrinsic utility of 0.5—because of the complete overlap of the reservation values), the closer to zero the differential between the intrinsic utility and the control utility becomes. As can be seen from figure 5.3.A, the only tactics which approach or exceed an average intrinsic utility of 0.5 are those which concede at a steady rate.

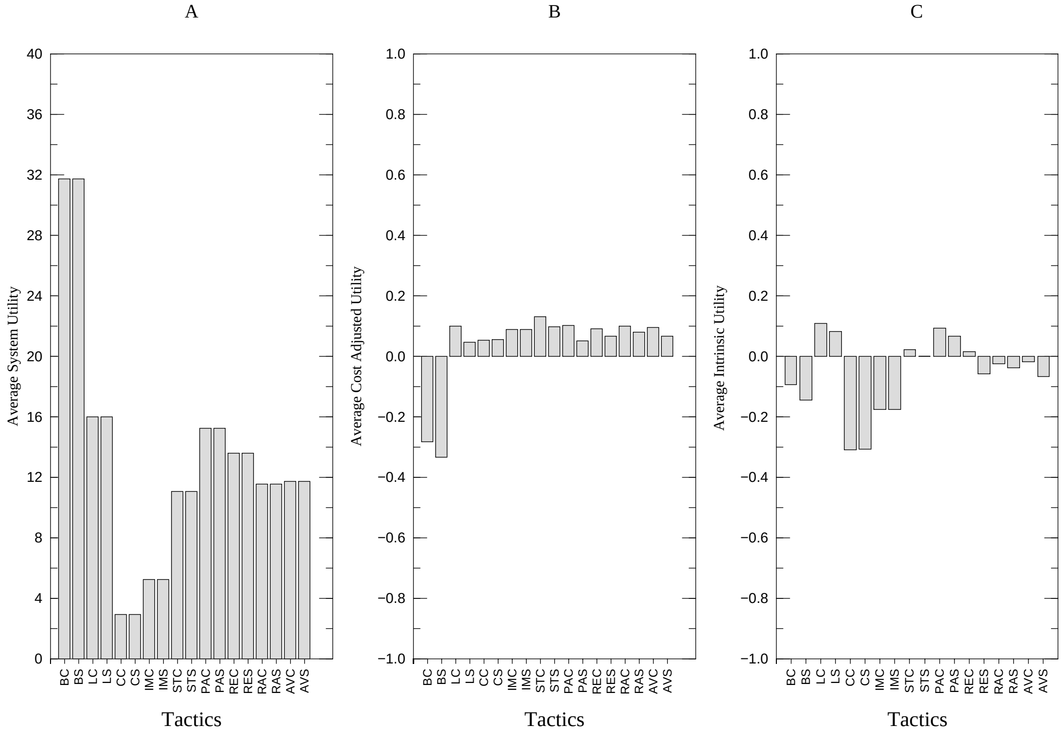

Figure 5.4: Average Non-Intrinsic Utilities and Control Utilities for Pure-Strategy Experiments in Long Term Deadlines: A) Average System Utility, B)Average Cost Adjusted Utility, C) Comparisons to Control.

5.3.4.2 Short Term Deadlines

Changing the environmental setting can radically alter the successfulness of a particular family of tactics. Therefore, an experiment is carried out to investigate the behaviour of tactics in cases where deadlines are short. For this case, the hypotheses are:

Hypothesis 3: *When there is a short time frame to negotiate, tactics which quickly approach their reservation values will make more deals.*

Hypothesis 4: *Since deadlines are short, the number of messages exchanged to reach a deal will be small. Consequently the system utility will be low.*

In this context, short term deadlines are obtained by sampling values for t^c_{max} and t^s_{max} between two and ten ticks of a discrete clock. The remainder of the experimental setup is as before. Figure 5.5 shows the results obtained for these experiments. The first observation is that for most tactics, the overall intrinsic utility, the system utility and the number of deals made (figures 5.5 A, C and B respectively) are significantly lower

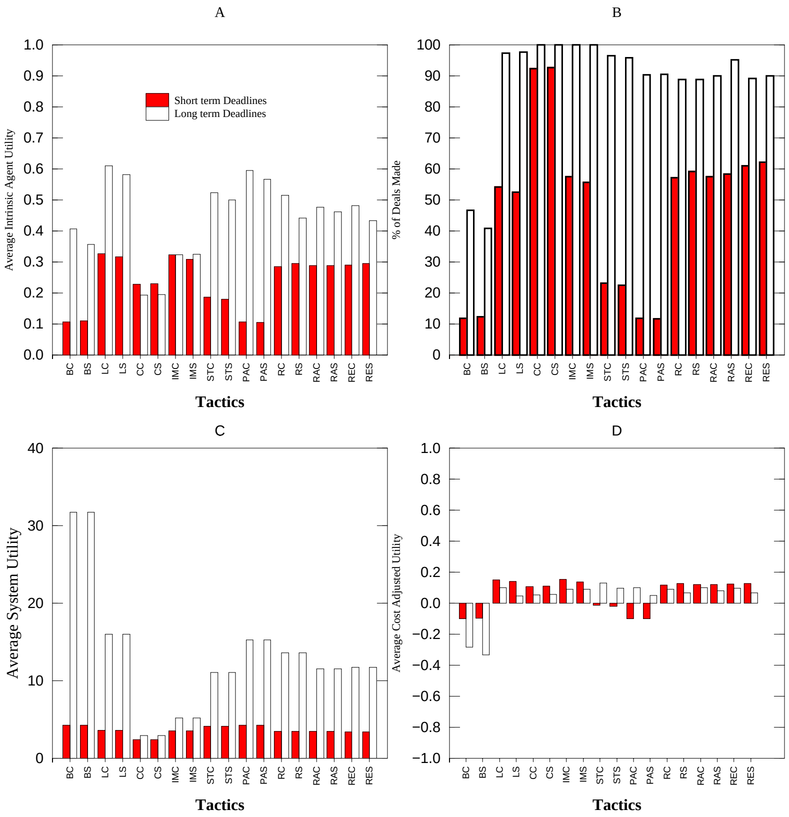

Figure 5.5: Comparative Data For Intrinsic, System and Cost-Adjusted Utilities And Deals Made For Pure-Strategy Experiments in Long And Short Term Deadlines. A) Average Intrinsic Utility, B) Percentage Number of Deals C) Average System Utility, D) Average Cost Adjusted Utility.

than the respective measures for the long deadline experiments. A lower system utility is expected since fewer messages can be exchanged in the allocated time. Note that since Conceder and Impatient are quick to reach agreements, their utilization of system resources is independent of the time constraints. Also, because fewer messages are exchanged, the agents pay less tax and, consequently, keep a greater percentage of their derived intrinsic utility (figure 5.5.D). These findings are all in line with the predictions in hypothesis four. However, the other measures require further analysis.

With long term deadlines, most tactics, apart from Boulware, make deals approximately 90% to 95% of the time, whereas with short term deadlines only Conceder makes anything like this number. This reduction is either because the tactics are insensitive to changes in their environment (e.g resource dependent tactics) or because they have a slow rate of approach to reservation values (e.g Boulware). Time insensitivity means the other tactics fail to make many deals when interacting with these tactics. Because the length of the thread is independent of the deadline, the resource dependent tactics cannot distinguish between short and long term deadlines. This claim is supported by the observation that Impatient gains equivalent intrinsic utility independently of deadlines (figure 5.5A). Furthermore, resource dependent tactics are differentiated with respect to μ, the amount of time an agent considers reasonable for negotiation. If an agent does not reason about deadlines *and* erroneously assumes a value for μ which is close to or above t_{max}, then it will be unsuccessful in environments where deadlines are important. The relatively low intrinsic utility of Patient and Steady (ranked 9th and 7th respectively—figure 5.5A) supports this claim. When the deadline is long, resource dependent tactics with $\mu > 1$ gain large intrinsic utility because they approach reservation values in a steady way. However, the same behaviour in short term deadlines is less successful. The imitative tactics also exhibit a reduction in average intrinsic utility. This is to be expected since these tactics imitate the relatively larger rate of concession of other tactics (especially time dependent tactics) when the deadline is shorter.

Hypothesis three is supported by the relative reductions in intrinsic utility for Boulware, Steady and Patient and by the comparative increase for Conceder and Impatient. Whereas in long term deadlines, Boulware, Steady and Patient ranked higher than the conciliatory tactics, the reverse is true for short term cases. With short term deadlines, tactics that quickly approach their reservation values gain higher intrinsic utility than those which are slower.

Again, it is observed that the dominant tactic is one which concedes at a steady rate (i.e Linear), suggesting that the best tactic, independent of time deadlines, is one that approaches reservation values in a consistent fashion. The behaviour dependent tactics also gain relatively high utilities in both cases, ranking third and fourth for short and long term deadlines respectively. Thus, whereas most tactics have large fluctuations in rankings across environments, the behaviour dependent family maintains a stable position, indicating its general robustness and usefulness in a wide range of contexts. This is because these tactics stick firm to avoid exploitation and reciprocate concession.

5.3.4.3 Initial Offers

In the formal model, an agent's reservation values are private. This means no other agent has any knowledge of where in the range of acceptable values an opponent begins its bidding process, nor where it is likely to end. Given this constraint, an agent must decide where in its reservation ranges it should begin *its* negotiation offers. That is, what should be the value of κ^a in the face of this uncertainty? To help answer this question, the following hypothesis is formed: [4]

> **Hypothesis 5:** *When the deadline for agreements is not short, making initial offers which have values near the maximum of U^a_{price} leads to deals which have higher intrinsic agent utilities*

[4] Note: U^s_{price} increases and U^c_{price} decreases with increasing price offers.

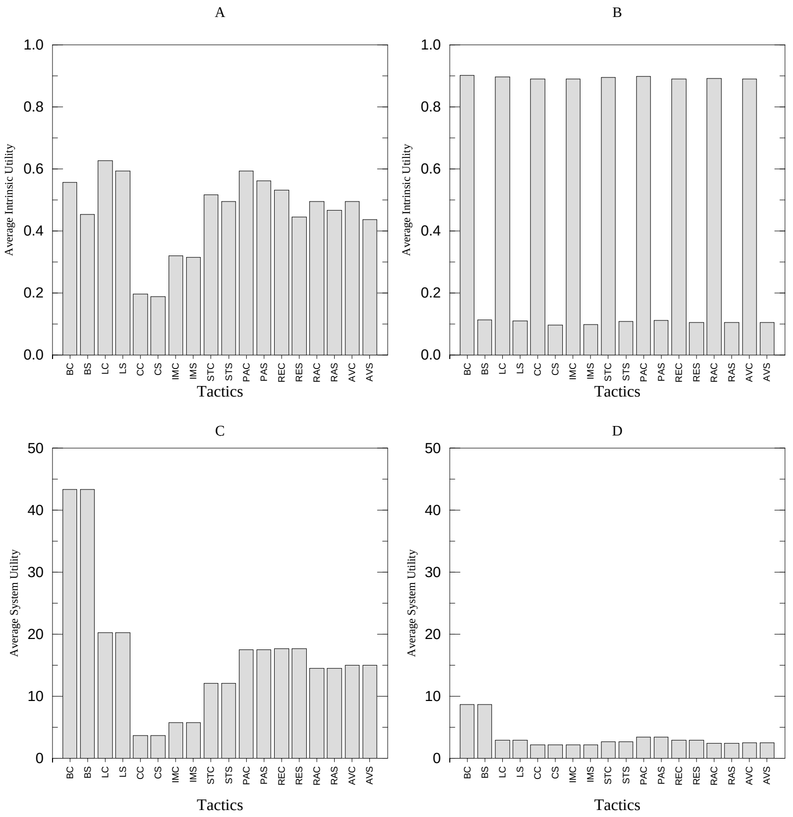

Figure 5.6: Average Intrinsic And System Utilities For Pure-Strategy Experiments With Low And High Initial Offers: A) Average Intrinsic Utility For $\kappa^s \in \{0.01, 0.2\}$, B) Average Intrinsic Utility for $\kappa^s \in \{0.8, 0.99\}$, C) Average System Utility For $\kappa^s \in \{0.01, 0.2\}$ and D) Average System Utility For $\kappa^s \in \{0.8, 0.99\}$. $\kappa^c = 0.1$ For All Cases.

> *than initial offers near the minimum of U^a_{price}. In other words, a server that starts bidding close to max^s_{price} is more likely to end up with deals that have a higher utility than a server who starts bidding close to min^s_{price}. The converse is true for the client.*

To test this hypothesis, both agents are allowed to have reasonably long deadlines, $t^c_{max} = t^s_{max} = 60$, and κ^c is made a constant at 0.1 (i.e the client is cautious in its first offer). Therefore, the single independent variable is κ^s, which is sampled between the values $[0.01, 0.2]$ for high initial price offers and $[0.8, 0.99]$ for low initial offers. All other environmental variables are chosen as in previous experiments. Figure 5.6 confirms the prediction that a server which begins bidding at values near the maximum of U^s_{price} (figure 5.6.A) has a higher average intrinsic utility than a server that begins bidding at values near the minimum of

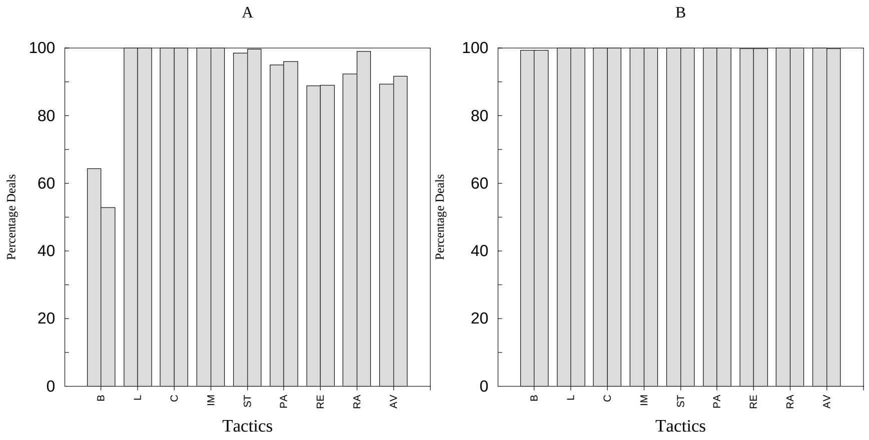

Figure 5.7: Percentage of Successful Deals For Low and High Initial Offers: When $\kappa^c = 0.1$ And A) $\kappa^s \in \{0.01, 0.2\}$, B) $\kappa^s \in \{0.8, 0.99\}$.

U^s_{price} (figure 5.6.B). Moreover, if κ^s is close to κ^c (the client starts bidding at low values and the server begins with high offers), then both agents gain equivalent utility in most cases and take many rounds of negotiations before a deal is found (figure 5.6.C). This is because the tactics begin their negotiation at some distance from the point in the negotiation space where bids have values which have a mutually acceptable level.

Conversely, if κ^s is not close to κ^c (both the client and server start bidding at low values), then the client benefits substantially more than the server. This is because the initial offers of the server are now immediately within the acceptance level of the client (confirmed by the number of messages exchanged before a deal is reached (figure 5.6.D)). Thus, the client gains relatively more utility than a server, since the initial offers of both agents are low and deals are made at low values.[5] The influence of κ on the behaviour of tactics can be further explained from the observations shown in figure 5.7. κ^a is used by all tactics for generating the initial offer but, for exposition purposes, only the results with respect to the Boulware tactic family are discussed (since this offers the greatest difference in behaviour). When κ^s is low, Boulwares have a lower percentage of deals relative to other tactics (figure 5.7.A). Conversely, when κ^s is high, Boulware almost equals all other tactics in the percentage of deals they make (figure 5.7.B). This is because at low values of κ^s, the shape of the acceptance level for Boulware is almost a step function, whereas when κ^s is high it is a straight line near to or at min^s. Thus a server playing a Boulware tactic makes a small number of high utility deals when the acceptance levels tend towards being a step function (compare figures 5.7.A

[5] When κ^s is distinctly different from κ^c there is little differentiation among intrinsic utilities. This is why $\kappa^a = 0.1$ for both agents in sections 5.3.4.1 and 5.3.4.2.

and 5.6.A), but makes larger number of lower utility deals when the acceptance level is almost a straight line (figures 5.7.B and 5.6.B). Therefore, as the value of κ increases, the likelihood of a deal increases, but the utility of the deal decreases.

5.3.4.4 Summary of Non-Strategic Experiments

It has been formally shown elsewhere that agents are guaranteed to converge on a solution in a number of very constrained situations (e.g. when two agents implement a time-dependent tactic, then the negotiation over an issue is guaranteed to converge if there is an overlap in the joint reservation values of that issues) using the tactical component of the wrapper's responsive mechanism (Sierra, Faratin, & Jennings 1997). The aim of the sections above was to extend these results empirically and to evaluate the non-strategic part of the responsive mechanism of the wrapper in a wider range of circumstances. To this end, a number of basic hypotheses were defined about negotiation using the tactical component of the wrapper. In particular, with respect to tactics the following were discovered: (i) irrespective of short or long term deadlines, it is best to be a linear type tactic, otherwise an imitative tactic; (ii) tactics must be responsive to changes in their environment; and (iii) there is a tradeoff between the number of deals made and the utility gained which is regulated by the initial offers.

The aforementioned results confirmed (and rebutted!) a number of basic predictions about negotiation using the tactical component of the wrapper. Next, the analysis is extended to strategic interactions.

5.4 Strategic Experiments

The aim of the previous experiments was to investigate the effects of non-strategic decision making. The aims of the experiments in this subsection are to empirically explore the causal relationships between *strategic* decision making on the dynamics and outcomes of negotiation. The overall aim is to empirically evaluate the postulate that consideration of a number of environmental factors *and* changes of these considerations (or dynamic strategies), lead to better negotiation outcomes than considering a number of environmental outcomes but not changing this initial consideration (static strategies). In addition to this, it is postulated that static strategies, in turn, leads to better negotiation outcomes than considering only one environmental factor (pure strategies). As will be shown below, better outcomes are defined as ones that maximize the joint utility of outcomes (a global measure). Therefore, from a global perspective, $dynamic\ strategies \succ static\ strategies \succ pure\ strategies$, where $\succ$ is should be read as the "better" operator.[6] Furthermore, the objective of the experiment is to show that changing of strategies *per se* is more beneficial than non-adjustment. Therefore, the objective is not to analyze the behaviour of different types of $f()$ given in equation 4.2, but rather the relative performance of a single strategic decision making

[6] Note, strictly speaking only the dynamic strategies are strategies as defined in section 4.4.3. However, for terminological simplicity throughout this chapter static consideration of one or a number of environmental factors will be referred to as strategies.

Variable Name	*Variable Scale*	*Variable Ranges*	*Variable Transformation*
$Agent$	categorical	$\{2, \infty\}$	categorical={player,opponent}
$\{J\}$	categorical	$\{1, \infty\}$	categorical={price,quality,time,penalty }
w_j^a	interval	$[0, 1]$	categorical={[0.1,0.5,0.25,0.15],[0.5,0.1,0.05,0.35] }
$[min_j^a, max_j^a]$	interval	$[[0, \infty], [0, \infty]]$	ordinal={perfect,partial}
t_{max}^a	interval	$[1, \infty]$	ordinal={large,low}
$Tactics$	categorical	$\{time, resource, behaviour\}$	categorical={boulware,linear,conceder,titfortat}
$sim_{w_j}^a$	interval	$[0, 1]$	categorical={perfect,partial,imperfect,uncertain,market}
h_j^a	categorical	$\{1, \infty\}$	
ϵ_j	interval	$[0, 0.5]$	value=0.1
γ	interval	$[0, 1]$	categorical={tough,linear,conceder,titfortat}

Figure 5.8: Strategy Experimental Independent Variables

compared to a non-strategic decision making.

The methodology of the experiments is similar to previous experiments—evaluation of a number of hypotheses in various *types* of environments as opposed to concrete cases. To this end, sections 5.4.1 introduce the data abstraction methodology and statistical methods necessary for definition of environments. Section 5.4.2 then defines the experimental measures, section 5.4.3 details the experimental procedures and, finally, section 5.4.4 presents the hypotheses and the discussion of results.

5.4.1 Experimental Independent Variables

This section introduces the set of experimental independent variables for the strategic experiments that are under the control of the experimenter. Like the non-strategic experiments, the set of experimental independent variables collectively define the environment of negotiation (section 5.4.1.1) and the tactics available for decision making (section 5.4.1.2). However, in the experiments reported in this section there is an additional set of variables, the strategy variables (section 5.4.1.3), which define the available strategies in negotiation. These experimental independent variables are introduced in figure 5.8. As before, the assignment of values to these variables is under the control of the experimenter whose main objective is to choose values for these variables that lower the complexity of the analysis. Note, in general throughout the experiments the *actual* concrete values of the independent variables mean very little in themselves. It is the *relative relationship* of an independent variable's value with respect to others that is important. Therefore, throughout the following exposition the actual values of independent variables are no longer justified and their values should be interpreted in comparison to other dependent variable values.

5.4.1.1 Environments

In these experiments, like the previous pure-strategy experiments, an environment is defined by the number of agents, the number of issues involved in negotiation, the deadlines to reach a settlement and the aspiration levels of agents. In these experiments negotiations are conducted between only two agents, categorically labelled as $player$ and $opponent$. However, in the pure-strategy experiments agents negotiate over multiple quantitative issues $\{price, quality, time, penalty\}$. The set of negotiation issues is expanded from one to four so as to facilitate a comparative analysis with the results of the trade-off mechanism experiments (which requires a minimum of two issues, section 5.5). This analysis is also restricted to quantitative issues, because the behaviour of both the responsive and trade-off mechanisms are less smooth with qualitative issues. This, in turn, masks the underlying behaviour of the model. For example, concession over qualitative issues produces scoring function outputs that are "bumpy", containing discrete points (since qualitative issues are naturally discrete valued objects). Likewise, the trade-off of a qualitative issue with a quantitative one often produces a transfer of score from one issue to another which may require the introduction of an auxiliary issue into the trade-off consideration to accommodate the correct score that needs to be transfered in trade-off. For example, consider a client of a service negotiating over a quantitative issue $price$ and a qualitative issue $colour$. Let the reservation values of the issue $price$ be $[10, 20]$, with score value ranges between $[0, 1]$, dictated by a continuously decreasing scoring function for increasing values over $price$. Let the reservation values of $colour$ be $[red, blue, green]$ with an associated score of $[0.8, 0.4, 0.1]$ respectively. Let the previous offer of the agent about to make a trade-off offer be $[20, green]$. Further assume that the iso-value is set at $\theta = 0.3$ (section 4.5.2.2), meaning that a score of 0.3 must be re-distributed among the two issues. One such re-distribution may be to decrease the score on the issue $price$ by 0.1 (thus the agent should offer less than 20 for the next offer over $price$) and increase the score on $colour$ by 0.2. However, an increase of 0.2 to the score of $colour$ will map to an offer of between $green$ and $blue$, which is not permitted. Another issue may have to be introduced to accommodate this residue score. Alternatively, the loss in score over $price$ can be computed *given* the gains that can be obtained from $colour$. However, this last solution is not satisfactory since it gives higher precedence to qualitative issues, and fails in cases where offers straddle, or are close to, the reservation values. Again, this masks the behaviour of the mechanisms and since the aim of the experiments is to analyze the underlying mechanisms, agents negotiate over quantitative issues only.

The other independent variables are as follows. The importance level for each negotiation issue is assigned concrete values $\{0.1, 0.5, 0.25, 0.15\}$ for the $player$ and $\{0.5, 0.1, 0.05, 0.35\}$ for the $opponent$. These weights are chosen because they allow comparative analysis of results with trade-off mechanisms, since they permit operation of the latter mechanism. For practical purposes, similar to pure-strategy experiments, the issues' interval values are converted from an interval to an ordinal scale which specifies both the length of the interval for each issue and the degree of overlap between the respective interval values for each issue (see section 5.3.1 for a more in-depth discussion of the methodology for computing interval values). The type of intervals considered in these experiments are those where the lengths of the interval values are equal and perfectly overlapping for each issue for both agents and are assigned the following values: Again, similar to pure-strategy experiments, the length of the interval value for each issue is chosen arbitrarily because the score of the offers models the ordinal and not the cardinal relationships between the interval values. Furthermore, to simplify the overall problem and reduce the complexity of analysis, the

$$\begin{aligned} min_{price} &= 10, max_{price} = 20 \\ min_{quality} &= 5, max_{quality} = 30 \\ min_{time} &= 20, max_{time} = 50 \\ min_{penalty} &= 1, max_{penalty} = 10 \end{aligned} \tag{5.1}$$

same interval values are assigned to both agents (hence a perfect overlap in interval values). The implication of this design are: i) that in bi-lateral negotiations between agents that *both* use a linear scoring function the *reference* point (or the most equitable outcome) is exactly at the mid point of an issue's interval value, with a score of exactly 0.5 for each agent and ii) a deal always exists. Note, that the actual concrete values for the intervals are insignificant and any values that obey the perfect overlap requirement will suffice. If one or both agents implement a non-linear scoring function then this mid point must "shift" along the utility scale. Fixed interval values with perfect overlap permits analysis of results with respect to a known reference point. Sampling interval values and the degree of overlap leads to a more complicated analysis of results because the location of the reference point can only be ascertained on an average basis.

The independent variable t^a_{max}, is assigned the same values as the previous pure-strategy experiments. The group $long$ term deadlines is defined as samples within the values of $30 - 60$ ticks of a discrete clock. Short term deadlines are defined as samples within $2 - 10$ ticks of a discrete clock.

5.4.1.2 Tactics

The other independent variables that are subject to transformation are the responsive tactics. To reduce the complexity of the analysis task, experiments are conducted using only the time-dependent and behaviour-dependent tactics (since time is a resource and time-dependent families model time sufficiently). The parameters of these tactics are randomly sampled. The same three members of the time-dependent family are chosen as for the pure-strategy experiments (figure 5.2); these correspond to behaviours that concede in time in a boulware, linear and conceder fashion. Again, to reduce the complexity of the experiments, only the relative-titfortat sub-family (section 4.4.2.4) is chosen to represent behaviour-dependent tactics. This category is defined as the transformation of interval values for δ into concrete value of 1. That is, relatively mimicking every last offer of the other agents. When the length of the negotiation thread is below δ (i.e insufficient offers have been exchanged between the agents) the titfortat default behaviour is to be a $conceder$ with a β value that is sampled within values of $[1.0, 3.0]$—a conceder tactic that is more conceder than a linear, but within certain limits of concession. A concrete tactic is chosen for each negotiation experiment by sampling within the range of the specified ordinal scale of that tactic.

5.4.1.3 Strategies

In these experiments an agent's strategy amounts to i) the initial assignment of relative importance weights for all issues (or computing the matrix Γ^0, see section 4.4.3) given the four experimental categories of tactics $T \in \{boulware, linear, conceder, \textit{titfortat}\}$, and ii) the modification of this initial consideration. An element of the Γ matrix is indexed by γ_{ij}, the weight of tactic j for an issue i. A row of the Γ matrix is indexed by γ_i, the tactics weight *array* for an issue i. The *relative differences* in the assignments of values to each of γ_{ij} in the γ_i array defines the agent's strategy for an issue in negotiation. For continence these strategies are labelled as follows. Given a set of tactics $j \in \{T\}$, a strategy for the issue i in negotiation can be one of the following:

- tough: where $j = boulware$ and γ_{ij} is assigned a higher weighting than other tactics $k, j \neq k$
- linear: where $j = linear$ and γ_{ij} is assigned a higher weighting than other tactics $k, j \neq k$
- conceder: where $j = conceder$ and γ_{ij} is assigned a higher weighting than other tactics $k, j \neq k$
- titfortat: where $j = \textit{titfortat}$ and γ_{ij} is assigned a higher weighting than other tactics $k, j \neq k$

As a simplification, *the same strategy is applied to all issues*. That is, the γ_i arrays for all the issues are the same. For example the Γ matrix:

$$\begin{array}{c} \\ price \\ quality \\ time \\ penalty \end{array} \left[\begin{array}{cccc} boulware & linear & conceder & \textit{titfortat} \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{array} \right]$$

specifies a strategy that assigns the boulware tactic the highest weight *for all issues*. Again, this simplification is intended as a measure to reduces the total number of free experimental variables and hence reduce the complexity of analysis. Therefore, the exposition will be described with reference to a single issue only (γ_i array). Application of the same strategy to each issue throughout the negotiation can serve as a base-case for future experiments that are more complicated and whose analysis is made more accessible from the base-case results. Note also, that the *strategy label* is derived from the highest weighted tactic, not to be confused by the tactic itself. Thus, a γ_i array with a value of $[0.7, 0.1, 0.1, 0.1]$ denotes a tough strategy. Conversely, a γ_i array with a value of $[0.1, 0.7, 0.1, 0.1]$ denotes a linear strategy, and so on. Since the aim of these experiments is to evaluate the differences between non-strategic and strategic decision making, the agents' strategies are evaluated in three classes of experiments:

- pure strategies
- mixed1 strategies
- mixed2 strategies

The differences between the classes of experiments are defined by i) the *magnitude* of the initial Γ^0 matrix and ii) the presence or absence of *change* in this initial Γ^0 matrix. A pure strategy simply consists of the assignment of binary values for γ_{ij} to the available tactic set. For example, for each issue a pure and tough strategist in the experiment consists of assignment to the tactic set $\{boulware, linear, conceder, \textit{titfortat}\}$ the γ_i array values $[1, 0, 0, 0]$ which does not change throughout the negotiation. Likewise, a pure and conceder strategist in the experiment would consist of γ_i assignment $[0, 0, 1, 0]$ to all issues which does not change throughout the negotiation. Therefore, pure strategies are the same as the base experiments where an agent's strategy consists of a static assignment of value 1 to one of the available tactic independent variables corresponding to the desired strategy.

A mixed1 strategy, on the other hand, consists of the assignment to the same tactic set of continuous, as opposed to binary, γ_{ij} values which also do not change throughout the negotiation. For example, a value of $[0.8, 0.066, 0.066, 0.066]$ for all issues in Γ denotes a mixed1 tough strategist. Thus, whereas pure strategies model the use of a *single* tactic in generating an offer, mixed strategies use a *combination* of tactics to generate offers (see section 4.4.3). Unlike pure strategies, because γ_{ij} is an interval valued variable, with the constraint that $\gamma_{ij} \in [0, 1]$ and $\sum_{j \in T} \gamma_{ij} = 1.0$ for all i, there can be an infinite number of values of γ_{ij} that implement the given strategy. However, the value of γ_{ij} has to obey an additional constraint that its value is within the range $[0.25, 0.9]$. This constraint restricts the range of possible values of γ_{ij} for a given strategy to be below a pure-strategy (hence 0.9 and not 1.0) and above the level where the tactic has equal weighting with the other tactics (since there are four tactics, the lower bound of the constraint is 0.25). For example, a γ_i array value of $[0.8, 0.066, 0.066, 0.066]$ specifies a tougher mixed1 strategist than a comparative γ_i value of $[0.5, 0.166, 0.166, 0.166]$. In the former case, the $boulware$ tactic has more of an input into the decision of the next offer generation than the other tactics, whereas in the latter case the other tactics have relatively more of an input in the decision making. Thus a tactic's influence on the final decision can range from no influence to fully dictating the decision (the case for a pure strategy). Therefore, to investigate different initial magnitudes of γ_{ij}, the degree of a tactic's magnitude/decision strength is made an independent variable Ω_{ij}, defined as the *initial* strength of the γ_{ij}^0 of issue i for tactic j at time 0. Assignments of *initial* values for each Ω_{ij} (for each issue and each tactic) then define Γ^0, the initial strategy of an agent at time 0 for all issues.

A mixed2 and tough strategist is similar to a mixed1 strategy, but now the initial Γ^0 array is dynamically modified throughout the negotiation. For example, a tough mixed1 strategy for an issue may correspond to the γ_i^0 array value of $[0.8, 0.066, 0.066, 0.066]$ (the agent considers the time factor to be important and does not change this consideration). However, these initial values of the γ_i^0 array are subject to *change* throughout the negotiation in the case of mixed2 experiments. Thus mixed2 strategies model not only the *combination* of tactics for generating an offer (same as mixed1 strategies), but also the *transition* in this combination during the course of negotiation (see section 4.4.3). This transition is formally specified as the $f()$ function (equation 4.2) that maps Γ^{t_n} to $\Gamma^{t_{n+1}}$, where t_n denotes the current time. However, like interval valued variables, there can be an infinite number of such mappings. In the case of these experiments the modification of the initial Γ^0 for all issues i is dictated by the following policy (equation 5.2) based on the notion of *similarity* (see equation 4.6):

$$\begin{array}{ll}
\text{If } 0.9 \leq sim(x,y) \leq 1.0 & \text{then } increase(\gamma_{i,boulware}, \Delta) \\
\text{If } 0.7 \leq sim(x,y) \leq 0.9 & \text{then } increase(\gamma_{i,titfortat}, \Delta) \\
\text{If } 0.4 \leq sim(x,y) \leq 0.7 & \text{then } increase(\gamma_{i,linear}, \Delta) \\
\text{If } 0.0 \leq sim(x,y) \leq 0.4 & \text{then } increase(\gamma_{i,conceder}, \Delta)
\end{array} \tag{5.2}$$

where x and y are the agent's and the opponent's last offer respectively, and $sim(x,y)$ is the similarity between the two contracts. There can be any number of modification policies, but rule 5.2 is chosen because it is simple and easily adjustable for experimental purposes (through modification of either the conditions of the rule or the action of the rule). Furthermore, since the objective of the experiment is to show that changing of strategies *per se* is more beneficial than non-adjustment, any reasonable rule which implements a modification of Γ would suffice.

The modification rule encodes the heuristic that if the agent believes that the two contracts x and y are very close then it should adopt a more $boulware$ strategy (since large changes, by being conceder, for example, may move the point of cross over of offers to positions where deals are less beneficial). On the other hand, if the two contracts x and y are believed to be dissimilar then a $conceder$ strategy should be adopted since movements in concessions may lead to the approaching of the zones of cross over of offers. In between these two extremes, a $linear$ and *titfortat* strategy should be adopted. Since for most strategies (especially with long term deadlines) the initial offers in negotiation are unlikely to be near the cross over of an issue interval (recall the results in section 5.3.4.3), the overall effect of the rule is to initiate a rate of concession to the crossover and then begin to lower this rate as crossover is approached. However, the consequence of rule 5.2 is to change the strategy of the agent independently to a new state, making the behaviour of mixed2 strategies an experimental variable that can not be manipulated. To overcome this problem, another variable (Δ) is added that modifies the behaviour of the rule under the control of the experimenter. The effect of Δ is to regulate the amount existing strategies change (i.e. it is a form of "resistance" to change). Thus, whereas the initial magnitude of the Γ^0 matrix completely defines mixed1 strategies, mixed2 strategies are defined by both the initial magnitude of Γ^0 *and* the dependent variable Δ, which specifies the percentage of change permitted to the initial Γ^0 matrix by rule 5.2. For example, a tough strategy for an issue i can be defined as $\gamma_i^0 = [0.8, 0.066, 0.066, 0.066]$ in mixed1 experiments. The same strategy in mixed2 experiments is then defined as a combination of the initial γ_i^0 array, $[0.8, 0.066, 0.066, 0.066]$ and the degree to which this tough strategy is allowed to be changed by rule 5.2. The degree of modification is given in percentile form, where the given γ_{ij} is *increased* by the specified percentile. The amount increased is removed equally from all other tactics, since $\sum_{j \in T} \gamma_{ij} = 1.0$ (section 4.4.3). Thus a Δ value of 80% over would specify a *tougher* mixed2 negotiator than a value of 5%, because a 80% change modifies to a greater extent the initial value of the tough strategy (0.8) than a 5% change. It should be noted that higher numbers for Δ result in dynamics of change in Γ that quickly reach the state where they are pure strategies.

The weights used for the similarity computation for the precondition of the update rule ($sim^a_{w_j}$ in figure 5.8, equation 4.5 and section 4.5.2.2) are $[0.25, 0.25, 0.25, 0.25]$, reflecting the agent's *uncertainty* about the other agents' issue importance evaluation (see section 5.5 for an explanation of other choices). The choice of criteria function (h^a_j in figure 5.8) is likewise infinite. The discriminatory power—the magnitude of the difference between the input and output—of the criteria function (equation 4.6) is set so that it exhibits two properties. Firstly, that it has more discrimination within the issues' interval values (as compared to values outside this range), since all of the negotiation will take place in this region. Thus, maximal discrimination should be between an issue's min and max values. This interval value requirement is parameterized by the independent variable ϵ. When ϵ is low, the function should be maximally discriminative for values within the issue's interval limits (*mutatis mutandis* when ϵ is high). Secondly, different discriminatory power *within* the interval range is also desired, to support different similarity measures for different issues (for generality and extension of these functions to trade-off experiments). For example, for one issue it may be desirable to have maximal discrimination at the center of the interval values, whereas for another issue maximal discrimination may be desired at the extremes of the interval values. This requirement is parameterized using the variable α. When α is high, more discrimination is placed towards the maximum of the interval values (*mutatis mutandis* when it is low). The following function satisfies these two requirements:

$$h(x) = \frac{1}{\pi} atan \left[\left(\frac{2 \mid x - min \mid}{x - min} \left| \frac{x - min}{max - min} \right|^{\alpha} - 1 \right) \; tan(\pi(\frac{1}{2} - \epsilon)) \right] + \frac{\pi}{2} \tag{5.3}$$

Figure 5.9 shows the effect of varying ϵ. Thus the discrimination power of the function decreases with increasing values of ϵ. In these experiments, in order to be quite discriminatory, ϵ is fixed at 0.1 for all issues. For all issues, α values are fixed to be equal: $\alpha^{price} = \alpha^{quality} = \alpha^{time} = \alpha^{penalty} = 1$, so as to have linear criteria functions $(h_i's)$, having equal discrimination power across the issue's interval values. ϵ

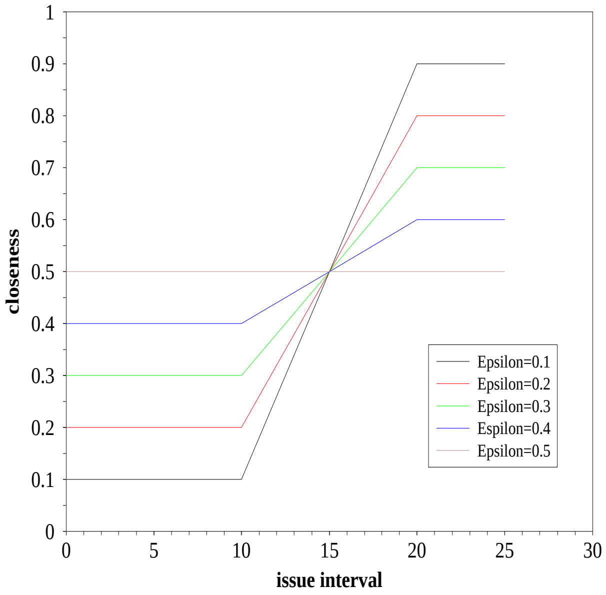

Figure 5.9: Criteria Functions For An Issue $Min = 10, Max = 20$

and α are made constant to reduce the number of free variables in the experiments. However, normally the setting of values for ϵ and α reflects the agent's domain knowledge.

The values for the γ_i array used for the strategy of each issue for each experiment class are shown in figures 5.11, 5.12, 5.13 and 5.14, corresponding to benchmark, increased Ω_{ij} for the $opponent$, increased Ω_{ij} for the $player$ and decreased Ω_{ij} for both the $opponent$ and the $player$ respectively. Recall that an increase (or decrease) in the initial values of an issue i strategy for tactic j at time 0 (γ^0_{ij}) across experiments is denoted as an increase (or decrease) in Ω_{ij}. Note also that the top row of each experiment class denotes the strategies of the $player$ and the bottom row of each experiment class denotes the strategy of the $opponent$. The benchmark experiments are included to establish a comparison criteria on the effect of increasing either the $opponent's$ or the $player's$ γ^0_{ij}, or, conversely, decreasing both agent's γ^0_{ij} levels, on the dependent variables. Following the same indexing convention as before, Δ_{ij} is the value tactic j can be changed for issue i. Furthermore, Δ_i arrays and γ_i arrays are identical for each issue $[price, quality, time, penalty]$.

It may be useful for the forthcoming discussion of results to imagine different tactics as different *forces* that attempt to "move" the score of the contract to a mutually acceptable point, the contract score at the cross over of offers. Figure 5.10 presents this analogy schematically, for one issue (for example $price$). Imagine the agent is a client. Therefore lower prices are preferred to higher prices. A $boulware$ tactic therefore attempts to generate prices that are distributed close to the minimum, whereas on the other extreme a conceder tactic generates price offers that reach the maximum quicker. Other tactics generate

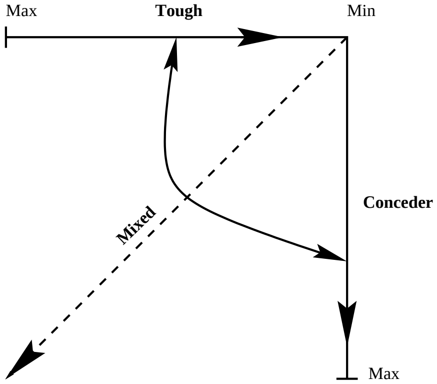

Figure 5.10: Analogy of Tactics As Forces.

offers on this minimum maximum continuum. A pure-strategy can then be envisaged as a mechanism that views a single force to reach the focal point. A mixed1 strategy, on the other hand, can be envisaged as considering a combination of forces to reach this convergence point. This is shown in figure 5.10 as the resultant force, the dotted line labeled *mixed*. The exact combination mixture (where the resultant line lies) is controlled by Ω_i. A mixed2 strategy can then be envisaged as a resultant force that not only considers a combination of forces, but also modifies the considerations as the environment changes. Note that there can be an infinite number of mixed1 resultant forces (mixtures) in between the tough and conceder strategies, corresponding to infinite values for Ω_i. However, whereas a mixed1 is a concrete selection and adherence to only one of these infinite possibilities, a mixed2 strategy also permits the "movement" of the resultant (the diagonal line in figure 5.10) along the tough-conceder axis (controlled by the independent variable Δ). Figure 5.11 shows values for the experimental independent variable γ_i array that are used as a benchmark for the other experiments which manipulate Ω_i (the magnitude of the initial strategy, or γ_i^0) for an issue i. For pure experiments, the strategies are simply assigned the value of 1.0 for the appropriate strategy for both the $player$ and the $opponent$. In mixed1 experimental classes, the value of the dominant tactic ($\{tough, linear, conceder, \mathit{titfortat}\}$) is assigned a value proportionally higher (three times) than the rest of the other tactics. Again, it is the ordinal, rather than cardinal, relationship between the variables that is of interest. The value of the dominant tactic is computed to be in the range $[0.25, 0.9]$ (as discussed in section 5.4.1). Since the values of the independent variables shown in figure 5.11 form the evaluation benchmark for the experiments that manipulate Ω_{ij}, the γ_{ij} for mixed1 of the dominant tactic j is set to 0.5 (within the constraint $[0.25, 0.9]$).

The remaining γ_i array for the other tactics is simply computed as the distribution of the residue weights according to the policy $(1 - \gamma_{ij})/3$. This policy is chosen because the aim of the experiments is to evaluate the relative and not the absolute differences in γ_i array. Mixed2 strategies, as mentioned above, are defined in terms of the two independent variables: initial γ_i array and the percentage permissible change of this value Δ_i by the weight update rule 5.2. The initial γ_{ij} for the dominant tactic of the strategy is set at 0.625 and the values of Δ_i array are, respectively, set to $[5, 25, 50, 40]$ for boulware, linear, conceder and titfortat strategies. These values reflect the relative persistence of the initial γ_i array in the course of negotiation. That is, for all issues, at each step in negotiation, a tough strategy permits only a 5% change to $\gamma_{boulware}$, a linear permits relatively more changes to γ_{linear}, conceder most of all, and titfortat in between linear and conceder strategies. The value of γ^0_{ij} for mixed2 experiments is higher than mixed1 experiments (0.625 and 0.5, respectively). A higher value for γ^0_{ij} is chosen because the update rule (especially in the case of conceder strategies) can reduce γ^0_{ij} too quickly to below mixed1 levels, thereby making it difficult to discriminate the results of mixed1 and mixed2 experiments. Thus the strategy in the mixed2 experiment classes is defined through the magnitude of the initial γ^0_{ij} and the *relative* permissible changes to this value through Δ_{ij}.

Note that the strategies of both the $player$ and the $opponent$ are constant and the same for all the experimental classes in the benchmark experiments. Generally, results are sought for *types* of environments. Therefore, γ_{ij} should ideally have been statistically sampled, allowing evaluation of contexts where γ_{ij} is not fixed. However, this methodology is not adopted because one of the aims of the experiments is to investigate the effect of Ω_i (or the strength of the strategy) on the dependent variables. To investigate the effect of Ω_i, the γ_{ij} distribution would have to be divided into bin sizes over the interval $[0.25, 0.9]$ (corresponding to the constraint above). Collecting values of γ_{ij} into small bin sizes and then statistically sampling each bin size would have resulted in distributions of γ_{ij} with similar values since the bin size is significantly small.

The independent variables shown in figures 5.12 and 5.13 show the experimental variables where the isomorphism between the $player$ and $opponent$ benchmark strategies is broken. Together with the independent variables shown in figure 5.14, these environments directly evaluate the effect of varying Ω_i. These variables are assigned these values to investigate the effect of *either* the $opponent$ or the $player$ increasing the value of Ω_i respectively. Note that since pure strategies are binary valued variables they cannot be included in Ω_i experiments. Thus, in figure 5.12 the $player$ dependent variables are unmodified from the benchmark experiments shown in figure 5.11. However, the values of γ_{ij} and Δ_{ij} are increased for the $opponent$. γ_{ij} of the dominant tactic is increased from 0.5 to 0.8 ($\Omega_{ij} = 0.3$). The implication of this change is that the $opponent$ in this environment is much more tough, linear, conceder or titfortat in its strategies. Likewise, the value of Δ_{ij} is relatively higher than the benchmark case, resulting in strategies that allow rule 5.2 to more freely modify γ_{ij} according to the distance to crossover in offers. Figure 5.13 shows the converse of 5.12, where the dependent variables for the $opponent$ are the same as the benchmark

Experiment Class	*tough*	*linear*	*conceder*	*titfortat*
pure	[1,0,0,0]	[0,1,0,0]	[0,0,1,0]	[0,0,0,1]
	[1,0,0,0]	[0,1,0,0]	[0,0,1,0]	[0,0,0,1]
mixed1	[0.5,0.166,0.166,0.166]	[0.166,0.5,0.166,0.166]	[0.166,0.166,0.5,0.166]	[0.166,0.166,0.166,0.5]
	[0.5,0.166,0.166,0.166]	[0.166,0.5,0.166,0.166]	[0.166,0.166,0.5,0.166]	[0.166,0.166,0.166,0.5]
mixed2	[0.625,0.125,0.125,0.125] Δ=5	[0.125,0.625,0.125,0.125] Δ=25	[0.125,0.125,0.625,0.125] Δ=50	[0.125,0.125,0.125,0.625] Δ=40
	[0.625,0.125,0.125,0.125] Δ=5	[0.125,0.625,0.125,0.125] Δ=25	[0.125,0.125,0.625,0.125] Δ=50	[0.125,0.125,0.125,0.625] Δ=40

Figure 5.11: Benchmark Strategy Experiments

Experiment Class	*tough*	*linear*	*conceder*	*titfortat*
mixed1	[0.5,0.166,0.166,0.166]	[0.166,0.5,0.166,0.166]	[0.166,0.166,0.5,0.166]	[0.166,0.166,0.166,0.5]
	[0.8,0.06,0.06,0.06]	[0.06,0.8,0.06,0.06]	[0.06,0.06,0.8,0.06]	[0.06,0.06,0.06,0.8]
mixed2	[0.625,0.125,0.125,0.125] Δ=5	[0.125,0.625,0.125,0.125] Δ=25	[0.125,0.125,0.625,0.125] Δ=50	[0.125,0.125,0.125,0.625] Δ=40
	[0.8,0.066,0.066,0.066] Δ=10	[0.066,0.8,0.066,0.066] Δ=40	[0.066,0.066,0.8,0.066] Δ=100	[0.066,0.066,0.066,0.8] Δ=80

Figure 5.12: $player$ With Benchmark Strategy And $opponent$ With Increased Ω_{ij}

Experiment Class	*tough*	*linear*	*conceder*	*titfortat*
mixed1	[0.8,0.06,0.06,0.06]	[0.06,0.8,0.06,0.06]	[0.06,0.06,0.8,0.06]	[0.06,0.06,0.06,0.8]
	[0.5,0.166,0.166,0.166]	[0.166,0.5,0.166,0.166]	[0.166,0.166,0.5,0.166]	[0.166,0.166,0.166,0.5]
mixed2	[0.8,0.066,0.066,0.066] Δ=10	[0.066,0.8,0.066,0.066] Δ=40	[0.066,0.066,0.8,0.066] Δ=100	[0.066,0.066,0.066,0.8] Δ=80
	[0.625,0.125,0.125,0.125] Δ=5	[0.125,0.625,0.125,0.125] Δ=25	[0.125,0.125,0.625,0.125] Δ=50	[0.125,0.125,0.125,0.625] Δ=40

Figure 5.13: $opponent$ With Benchmark Strategy And $player$ With Increased Ω_{ij}

Experiment Class	*tough*	*linear*	*conceder*	*titfortat*
mixed1	[0.3,0.23,0.23,0.23]	[0.23,0.3,0.23,0.23]	[0.23,0.23,0.3,0.23]	[0.23,0.23,0.23,0.3]
	[0.3,0.23,0.23,0.23]	[0.23,0.3,0.23,0.23]	[0.23,0.23,0.3,0.23]	[0.23,0.23,0.23,0.3]
mixed2	[0.3,0.23,0.23,0.23] Δ=5	[0.23,0.3,0.23,0.23] Δ=5	[0.23,0.23,0.3,0.23] Δ=5	[0.23,0.23,0.23,0.3] Δ=5
	[0.3,0.23,0.23,0.23] Δ=5	[0.23,0.3,0.23,0.23] Δ=5	[0.23,0.23,0.3,0.23] Δ=5	[0.23,0.23,0.23,0.3] Δ=5

Figure 5.14: Strategies For Both Agents Decreased Ω_{ij}

Variable Name	*Variable Scale*	*Variable Ranges*
$cycles$	interval	$[1, t_{max}]$
$V^a(outcome)$	interval	$[0, 1]$
$V^a(reference)$	value	0.5
$V^a(pareto)$	interval	$[0, 1]$

Figure 5.15: Experimental Dependent Variables

case in figure 5.11 and it is the $player$ that has increased magnitude of strategy.

Finally, the effect of varying Ω_i for *both* the $player$ and the $opponent$ from the benchmark is shown in figure 5.14. γ_{ij} is decreased from 0.5 to 0.3 resulting in strategies that, although they are still defined as strategies, have nonetheless a lower influence on the final decision. This allows other tactics to have relatively more strength (than the benchmark case) in the final decision. Likewise, Δ_{ij} for the mixed2 experiment class is uniformly lowered to a 5% level for all strategies, resulting in an environment where the γ_i array is modified smoothly across all strategies.

5.4.2 Experimental Measures

The previous section described the independent variables that can be manipulated by the experimenter and their effects observed on the dependent variables. Figure 5.15 shows the experimental dependent variables, one calibrating the *process* of negotiation ($cycles$), and three others for measuring the *outcome* of negotiation. Each dependent variable are described in more depth in the sections below.

5.4.2.1 Communication

A much simpler form of on-line cost, compared to the pure-strategy experiments, is defined by the independent variable $Cycles$. $Cycles$ calibrates the total number of messages exchanged in the course of a *single* negotiation run of the experiment (or the communication message load a strategy places on an agent). This simple form of on-line cost is used to disassociate the costs from the intrinsic utility of the strategy (methodology of the pure-strategy experiments) so that the agent can make decisions about the communication cost of the strategy, rather than the resulting cost-adjusted utility. The statistics used for $Cycles$ are simply the average number of messages exchanged for a strategy pairing across all experimental runs.

5.4.2.2 Intrinsic Utility

$Outcome$ is the categorical variable that measures the final outcome of negotiation in terms of success ($Accept$) or failure ($Withdraw$). Given an outcome the intrinsic utility of a deal, $V^a(outcome)$, is the *individual* agent utility of the deal. The form of the utility function is the same as the one given in pure-strategy experiments reported in section 5.3.3.1, defined as the linear scoring function:

$$V^a(x) = \sum_{1 \leq j \leq n} w_j^a V_j^a(x_j)$$

where x is the outcome, n is the total number of issues, and the value of the individual issue j to agent a, $V_j^a(x_j)$, is computed as:

$$V_j^a(x_j) = \begin{cases} \frac{max_j^a - x_j}{max_j^a - min_j^a} & \text{if decreasing} \\ \frac{x_j - min_j^a}{max_j^a - min_j^a} & \text{if increasing} \end{cases}$$

where $increasing$ and $decreasing$ refer to the direction of change in score as the value of that issue increases. For example, increasing the $price$ of the service decreases the score for a client, but increases it for a seller. Like $Cycles$, the statistics for $V^a(x)$ are simply the average utility of the deal when using a strategy across all experimental runs.

5.4.2.3 Experimental Controls

The analysis of the observed average utility data distribution will be made with respect to three reference points: i) the constant-sum line (see section 2.2.3), ii) the reference point and iii) the pareto-optimal line. See figure 3.1 for an explanation of each of these points. Recall from section 2.2.3 that the significance of the constant-sum line is that outcomes that lie on this line result in individual agent utility whose joint score adds up to 1.0—that is $V^{player}(outcome) + V^{opponent}(outcome) = 1.0$. This line is used as a control because outcomes that lie on it represent distributive bargaining situations and conversely, integrative bargaining for the outcomes that lie above it. Indeed, in negotiation over a single issue (distributive negotiation) the sum of utilities of an outcome *has* to be equal to 1 when the scoring functions of both agents are linear—an outcome with a utility of 0.8 for one agent determines the maximum the other agent can receive for this outcome is 0.2. In fact, for single issue negotiations the constant-sum line *is* the pareto-optimal line—there is no other deal that *both* agents prefer without one agent being worse off. It is by introducing multiple issues that the sum of individual utilities can be different to 1.0. Therefore the strategies could be evaluated with respect to the integrative and distributed bargaining dimension. However, as will be shown below, the experimental choice to assign the same Γ matrix to each issue results in the responsive mechanism selecting, at best, outcomes that lie on the constant-sum line, and, at worst, outcomes that lie below this line. The constant-sum line is included, together with the pareto-optimal line, for comparative analysis of the results obtained with the trade-off mechanism. Note, for multi-issue and differentially weighted issues, outcomes can lie below the constant-sum line, representing outcomes whose joint utility is lower than 1.0.

Outcomes that lie on the constant-sum line represent one set of possible distributions of utilities, or ways of “dividing the utility pie”. These outcomes are not equitable (recall that equitable is defined as equal distribution of utilities)—a utility distribution of $(0.8, 0.2)$ and $(0.1, 0.9)$ both equivalently maximize the sum of the individual utilities, but the first outcome is more favorable for the first agent and the second outcome is more favorable for the second agent. As mentioned in section 3.1.4, the Nash point is an equitable outcome, computed as the deal that maximized the *product* of the final utilities (see figure 3.1). However, recall the argument presented in section 3.1.4 against the use of the Nash solution for multi-dimensional negotiation—whereas computation of the Nash solution is straightforward for distributive (or single issue) negotiations (indeed, the Nash solution was the control measure in the previous non-strategic experiments), the same is not true for integrative negotiations involving different importance levels and intervals for each issue. For these reasons, the Nash solution control outcome is replaced with the reference outcome, simply computed as the intersection at the mid point of each agent's interval value for all issues. Unless stated otherwise, the reference point for a pair of linear scoring functions is specified as the utility coordinate point $(0.5, 0.5)$ and is constant in the experiments because the interval values of agents overlap perfectly and do not change.

The Pareto-optimal measure is included for comparative analysis of data *across* the responsive and trade-off experiments. Pareto-optimality ($V^a(pareto)$) is computed as the outcome that maximized the *sum* of the deals. Five pareto-optimal outcomes are computed and a line that joined the utility value points of these five deals is used as a control line of the closeness of the experimental outcome to a pareto-optimal outcome (see (Raiffa 1982), pp.163–165). The first pareto optimal deal is simply a value of 1 for the $player$ and 0 for the *opponent*, $(1, 0)$. The second is the converse $(0, 1)$. The third pareto optimal outcome is computed by selecting the values for each issue x_j in negotiation that maximizes the *combined* value of all the issues for both agents:

$$\sum(w_j^{player} * V(x_j^{player})) + \sum(w_j^{opponent} * V(x_j^{opponent}))$$

where w_j is the weight of issue j. The fourth pareto optimal outcome is computed by selecting the values for each issue which maximizes $player$ utility plus *half* the *opponent* utility. This gives the *opponent* less weight:

$$\sum(w_j^{player} * V(x_j^{player})) + 0.5 \sum(w_j^{opponent} * V(x_j^{opponent}))$$

The final pareto optimal contract is computed by selecting the values for each issue that maximizes $player$ utility plus *twice* the *opponent* utility. This gives the *opponent* more weight:

$$\sum(w_j^{player} * V(x_j^{player})) + 2 \sum(w_j^{opponent} * V(x_j^{opponent}))$$

The pareto-optimal line, in turn, is indicated in the figures of results as the solid line that connects these five points.

Where appropriate, statistical averages and standard deviation of averages *across* strategies will be given, respectively, to represent the center of the density and the variation of a group of outcome distributions with respect to the reference point. For example, four different strategies that result in a sum total utility average of 0.5 and a standard deviation of 0.0 identify a distribution of different strategy outcomes which lie exactly on the reference point. Variations in the averages then indicate the distance of the final average outcome from the reference and the standard deviation measures the degree of variation of the averages from the reference point. Averages and standard deviations of a group of strategies will be presented only for the $opponent$ since the distribution of outcomes for the $player$ is simply one minus the average of the distribution of the $opponent$.

5.4.3 Experimental Procedure

The experimental procedure consists of games between each pairing of $player$ and $opponent$ strategies (tough, linear, conceder, titfortat) for each of the Ω settings in figures 5.11, 5.12, 5.13 and 5.14 and for each of the experiment classes (pure, mixed1, mixed2). This procedure is shown algorithmically in figure 5.16.

Two strategies are paired to begin negotiation by selecting initial Ω_i levels for all issues for both the $player$ and $opponent$ for each type of experiments (line 12). A game then consists of playing the $player$ strategy against the $opponent$ strategy N times (line 13). On each run $first, \beta_{boulware}, \beta_{conceder}, \beta_{linear},$ $\Gamma_{controls2}$ and $\delta_{titfortat}$ (where $first$ is the agent that proposes the first contract and $\Gamma_{controls2}$ is a random sampling of γ_{ij}, described more below) are sampled for each agent (lines 8, 9, 10 and 11). N is set at 300 runs which means that the probability of the sampled mean deviating by more than 0.01 from the true mean is less than 0.05. At the end of each run, the depedent variables $V^{player}(outcome)$, $V^{opponent}(outcome)$ and $cycles$ are measured (lines 14, 15 and 16). After N runs, the *averages* for all the dependent variables are computed (lines 18–22). Note the difference in the analysis between these experiments and the previous pure-strategy experiments reported in section 5.3. In the latter set of experiments the analysis was at the *collective* level, where the final average measure of dependent variable (such as utility) of a strategy was summed and averaged across all other strategies. However, the analytical unit of this set of experiments is the average of dependent variable measure for a *pair*, rather than a collection of strategies.

For the mixed1 experimental class there are two additional $opponent$ strategies for each of the $player$ strategies, corresponding to the controls (line 10). The $player$ in the mixed1 experimental class plays not only against the $opponent$ strategy, but also a $control1$ $opponent$ (where the opponent's strategy is simply the γ_i array $[0.25, 0.25, 0.25, 0.25]$ for all issues and all tactics) and a $control2$ $opponent$ (which corresponds to a random sampling of Γ). $Control1$ is included to evaluate the performance of various mixed1 strategies against a strategy that behaves linearly across all tactic sets and thus reflects an $opponent$ that is uncertain about which strategy to choose. Note that the Γ matrix of $Control1$ is almost the same as the Γ matrix of both agents in experiments where Ω is decreased linearly for both agents (mixed1 strategies in figure 5.14). Therefore, these controls are only significant in other experimental Ω_i levels. $Control2$, on the other hand, is included to evaluate the performance of strategies against a random benchmark. Controls are not possible for the pure experimental class since the values of γ_{ij} are binary. Mixed2 strategies do not

```
$\Omega_i := \{[0.5, 0.5], [0.5, 0.8], [0.8, 0.5], [0.3, 0.3]\}$;             /* number of changes in magnitude of strategy */
$strategy^{player} := \{tough, linear, conceder, titfortat\}$;          /* player's strategy */
$strategy^{opponent} := \{tough, linear, conceder, titfortat\}$;        /* opponent's strategy */
$class := \{pure, mixed1, mixed2\}$;                                    /* classes of experiments */
$N$;                                                                    /* number of experimental runs */
$k :=| class |; m :=| \Omega |; p :=| strategy^{player} |; o :=| strategy^{opponent} |$;
$l := 0; n := 0; i := 0; j := 0; r := 0; N := 300$;
begin
  $reference := argmax_x\{V^{player}(x) * V^{opponent}(x)\}$;
  $pareto := argmax_x\{V^{player}(x) + V^{opponent}(x)\}$;
  while($l < k$) do $l := l + 1$;
      while($n < m$) do $n := n + 1$;
          while($i < p$) do $i := i + 1$;
              while($j < o$) do $j := j + 1$;
                  while($r < N$) do $r := r + 1$;
                  $env_{player} := sample(t^{player}_{max}, \beta^{player}_{boulware}, \beta^{player}_{conceder}, \beta^{player}_{linear}, \delta^{player}_{titfortat})$;
                  $env_{opponent} := sample(t^{opponent}_{max}, \beta^{opponent}_{boulware}, \beta^{opponent}_{conceder}, \beta^{opponent}_{linear}, \delta^{opponent}_{titfortat})$;
                  $control_1 := [0.25, 0.25, 0.25, 0.25]; control_2 := random(0, 1)$;
                  $first := random(player, opponent)$;
                  $pairs^r_{ij} := (strategy^{player}_i, strategy^{opponent}_j)$;
                  $(thread^r_{ij}, outcome^r_{ij}) := play(pairs^r_{ij}, first, env_{player}, env_{opponent}, control_1, control_2)$;
                  $V^{r^{player}}_{ij} := V^{player}(outcome^r_{ij})$;
                  $V^{r^{opponent}}_{ij} := V^{opponent}(outcome^r_{ij})$;
                  $cycles^r_{ij} := length(thread^r_{ij})$;
                  endwhile
                  $\overline{v}^{player}_{ij} := \sum_{r:=1}^{N} V^{r^{player}}_{ij} /N$;
                  $\overline{v}^{opponent}_{ij} := \sum_{r:=1}^{N} V^{r^{opponent}}_{ij} /N$;
                  $\overline{cycles}_{ij} := \sum_{r:=1}^{N} cycles^r_{ij} /N$;
              endwhile
          endwhile
      endwhile
  endwhile
end
```

Figure 5.16: Experimental Procedure Algorithm

	pure	*mixed1*	*mixed2*
pure	●		
mixed1		●	
mixed2			●

Figure 5.17: Experimental Class Execution Order

encounter any other control strategies either since the aim of these experiments is to show that modification of strategies per se is better than non-modification. Thus, the best one can achieve is interactions between a mixed2 strategy and a highly stylized negotiator (mixed1 and pure strategies in these experiments), as opposed to a random or another purposeful mixed2 strategist.

The experiments are also restricted to games between similar experimental class (see figure 5.17). Thus strategies are evaluated for cases when *both* the $player$ and the $opponent$ are pure, mixed1 or mixed2 strategists. Encounters between, for example, a pure $player$ and a mixed1 or a mixed2 $opponent$ (and vice-versa) are excluded because the generated data set in the latter case would be very large. In the former case, the number of generated data points is 224 (number of $player$ strategy $*$ number of $opponent$ strategy $*$ number of Ω_i experiments= $((4*4*4)+(4*6*4)+(4*4*4))$). In the latter case the generated data set is of size 736 making the analysis complex. The experiments were written in Sicstus3.7.1 Prolog and ran on *HP* Unix parallel machines at the Centre de Supercomputació de Catalunya *CESCA* (Barcelona), utilising four CPUs, 7MB of memory and lasted 1112.41 seconds.

5.4.4 Hypotheses and Results

The experimental hypotheses and results are presented in this section. Because the aim of the experiments is to investigate the benefits of dynamic strategic decision making over static and pure strategies (and not necessarily the causal relationship between a given strategy type and a *combination* of any number of dependent variables), results are presented and discussed for each experimental class (pure, mixed1 and mixed2) and their effects on the individual dependent variables: i) the final average utilities for outcomes, ii) the communication load and iii) the number of successful outcomes. Thus the aim is not so much an analysis of the effects of, for example, a pure-strategy on the final average utility of an outcome *and* its relationship with the communication costs, but rather the differential effects of pure, mixed1 and mixed2 strategies on a single dependent variable, in this example, the final average outcome. Note, that all the hypotheses for the effects of strategies on final average utilities will be quantitatively represented as the relationship between the expected outcome utilities and i) the reference point representing the maximum joint gain that is also equitable and ii) the constant-sum line outcomes representing maximum joint utility that may not be equitable.

Before presenting the individual hypotheses and results a side-effect, observable in all of the forthcoming data, is identified, directly resulting from the choice of assigning the *same* γ_i array to all the issues. For example, a tough strategy specifies a tough strategy for all the issues in negotiation. The observation from all the data is that the best **joint** outcome any combination of strategies, in either pure, mixed1 or mixed2 experimental classes, can attain is a contract score at the mid point of the cross over of the agents' interval values (or the reference point), independently of the pairing of strategies. This is so for the following reason. The independent variable $[min_j^a, max_j^a]$ of both agents has been designed to be perfectly overlapping, for all the issues $[price, quality, time, penalty]$. The weights of the $player$ and $opponent$ for each of the issues are $[0.1, 0.5, 0.25, 0.15]$ and $[0.5, 0.1, 0.05, 0.35]$, respectively. These weights mean that the $player$ views quality to be the most important issue, followed by time, followed by penalty and finally, least important issue, price. The $opponent$, on the other hand, views price as the most important, followed by the penalty, followed by the quality and finally time. Given these interval values, importance weights and the linear scoring function of section 4.2.1, the value of the reference point $[15, 17.5, 35, 5.5]$ (mid point of each issue, section 5.4.1.1) for the $player$ is computed as:

$$(0.1 * 0.5) + (0.5 * 0.5) + (0.25 * 0.5) + (0.15 * 0.5) = 0.5$$

It is trivial to show that the same score (0.5) will result for the $opponent$ for the same reference point $[15, 17.5, 35, 5.5]$. Now consider another contract, $X^{'}$, in the space of possible deals, $[18, 20, 35, 5.5]$. This contract will be more beneficial to *both* agents, because:

$$V^{player}(X^{'}) = (0.1 * 0.2) + (0.5 * 0.6) + (0.25 * 0.5) + (0.15 * 0.5) = 0.52$$

$$V^{opponent}(X^{'}) = (0.5 * 0.8) + (0.1 * 0.4) + (0.05 * 0.5) + (0.35 * 0.5) = 0.64$$

Thus increasing the values for the issues price and quality from the reference contract values to the $X^{'}$ contract value is more beneficial to both agents (i.e moving north-easterly in the direction of the pareto-optimal line). However, in these experiments the responsive mechanism is a concession protocol which can not support increase in utility scores where agents begin the negotiation from the reference point and then move towards more pareto-optimal contracts. Furthermore, agents are assumed to be unaware of one another's interval values, making the computation of the reference contract ($[15, 17.5, 35, 5.5]$) impossible. One way agents can reach $X^{'}$, or better, is to select one outcome from the space of possible outcomes. Next the agents assign a *different* Γ matrix to each issue. In this example this means that the $player$ concedes *more* on the price and less on quality of a service. Conversely, the $opponent$ can concede more on quality than on price. The combination of these two Γ matrices means different concession rates on different issues in such a way as to reach $X^{'}$, or better. However, this policy of making strategic decisions (assigning γ_i arrays for each issue) dependent on the weight of an issue (for example, a more important issue will be

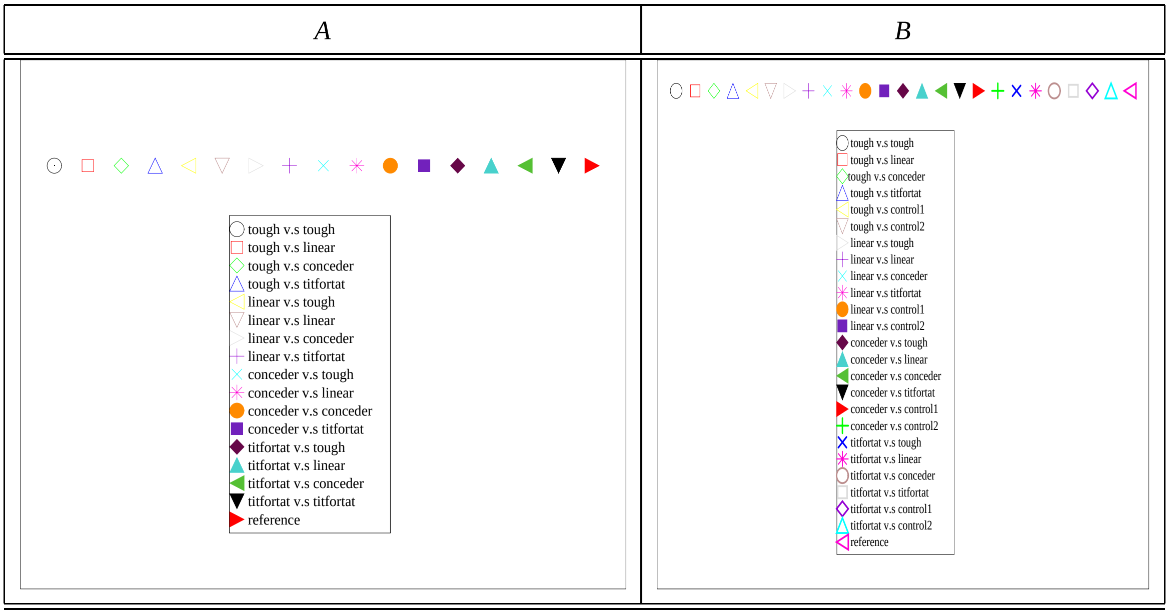

Figure 5.18: A) Key For Pure and Mixed2 Strategy Pairings. First Entry of Label Specifies The *opponent* Strategy And The Second The *player*. B) Key For Mixed1 Strategy Pairings. First Entry of Label Specifies The *opponent* Strategy And The Second The *player*.

assigned a higher γ_{ij} value to boulware tactic) is not adopted in the experiments because, as was mentioned in section 5.4.1.3, of the need to control the number of free experimental independent variables. Indeed, these experiments are viewed as base-case strategic experiments which form the basis for the design of future strategic experiments.

5.4.4.1 Pure-Strategy Utility Results

The effects of a pure-strategy on the set of dependent variables has already been discussed in section 5.3. However, the methodology of analysis is different (see section 5.4.3) hence the experiments are repeated here in these new environments for comparative reasons.

The expectation for the results of these experiments are summerised by the following hypothesis:

> **Hypothesis 7:** *Pairings of two pure strategies that approach their interval values less quickly will result in final average outcomes that are lower in* **joint** *utility than pure strategy pairs where at least one strategy approaches the interval faster.*

The hypothesis states the intuition that an encounter between, for example, two tough strategies will result in a group utility that is worse than when at least one of the strategies concedes (since concession increases the other's share of the utility). For the discussion of average utility results see figure 5.18 A for the key of each strategy pair for the average utility data for pure and mixed2 experiments and figure 5.18 B for mixed1

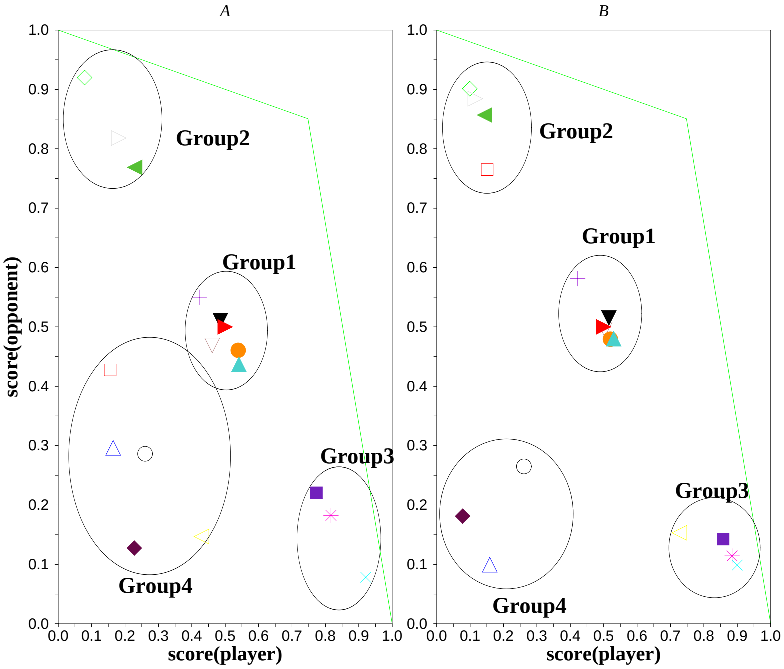

Figure 5.19: Comparative Final Joint Average Utility For Pure Strategies. A) Average Intrinsic Utility For Short Term Deadline, B) Average Intrinsic Utility For Long Term Deadline.

experiments (which include the two control conditions).

Figures 5.19 A and B show the observed average outcome utilities for the $player$ (x axis) and the $opponent$ (y axis) of the pure-strategy benchmark experiments with the independent variables shown in figure 5.11.

The first observation is that the argument in section 5.4.4 (that because the γ_i arrays for each issue are the same the responsive mechanism can not do better than outcomes lying on the constant-sum line) is supported by the observations of outcome utilities in both short term and long term environments. No strategy pair does significantly better than the reference point, by moving north easterly towards the pareto-optimal line, independently of the time limits.

Hypothesis 7 is also supported by the observed data in figure 5.19. The data in figure 5.19, A is clustered into roughly four groups. The first group (shown as $group1$), are the best outcomes, in that they are closest to the reference point, thus resulting in a more equal distribution of final utilities. $group1$ members are the strategy pairings $(linear, linear)$, $(conceder, conceder)$ $($*titfortat*,*titfortat*$)$, $(linear,$ *titfortat*$)$, and $($*titfortat*$, linear)$. These strategies correspond to the cases where *both* agents adopt a concessionary approach to the cross over of the interval values. The group's total mean and standard deviation is 0.485 and

0.04 respectively (recall that outcomes with a perfect coincidence with the reference point will have a mean and standard deviation of 0.5 and 0.0 respectively). The observed standard deviation of $group1$ statistically represents the tightest cluster of these outcomes around the reference point in figure 5.19, A.

The next two groups of outcomes, group 2 and group 3, also lie on, or close to, the constant-sum line, but the distribution of individual outcomes is less uniform compared to $group1$. $group2$ members are the strategy pairings $(tough, conceder)$, $(linear, conceder)$, $(\textit{titfortat}, conceder)$, which lie on the north west sector of the outcome grid (resulting in higher valued outcomes for the $opponent$, since *only* the $player$ is conceder in all their strategies). Conversely, mirroring $group2$ outcomes is $group3$, whose members are the strategy pairings $(conceder, tough)$, $(conceder, linear)$, $(conceder,\textit{titfortat})$. These lie on the south east sector of the outcome grid resulting in higher valued outcomes for the $player$, since $opponent$ is conceder in all its strategies.

Finally, $group4$ is the clustering of outcomes that do not lie on the constant-sum line (south west sector of the outcome grid) and occur with the strategy pairings $(tough, tough)$, $(tough, linear)$, $(tough, \textit{titfortat})$, $(\textit{titfortat}, tough)$ and $(linear, tough)$. $group4$ outcomes are the worst outcomes because they result in final joint average utilities that are lower than all other outcomes. These four groups of observations support hypothesis 7 —in groups $1, 2$ and 3 there is at least one strategy that approaches its interval faster than the others. However, in $group4$ both are either $tough$ or imitate a tough strategy or are linear.

Roughly four groups are once again observed when the environment is changed from short term to long term deadlines, figure 5.19, B. However, this time there are less members in $group4$—$(tough, linear)$ and the converse member $(linear, tough)$ now belong to $group2$ and $group3$ respectively. This further supports the stated hypothesis since $group4$ is now purely composed of tough strategies. However, although a strategy that approaches its interval value slowly does *individually* badly, *collectively* (similar methodology as the previous tactic experiments, when the results are averaged across all other strategies) there is an increase in final average utility. Results show that, for example, a $tough\ player$ strategy gains an average of 5% of utility when utilities are averaged across all other strategies in long term deadlines. It is interesting to note that the performance of a $tough$ strategist is lowered when more time is given for negotiation when encountering a *titfortat* strategist. Statistically the total average of outlying data decreased from 0.247 to 0.182 with a standard deviation of 0.067. This result is explained by the fact that the *titfortat* strategy is a $conceder$ until it can begin to imitate other's responses. Therefore, under short term deadlines the strategy concedes (hence moves closer to constant-sum line), whereas in longer term deadlines it has more opportunity to imitate the other's strategy ($tough$ in this consideration) and as such becomes $tough$ too (hence a deal is only possibly reached in the last few moments of negotiation). This pushes the outcomes further away from the constant-sum line. In general, for all the experiments described below noticeable effects of time limits on strategies are more observable for data that calibrate the *process* (the costs of communication) and less on the *outcome* of negotiation. Note, this is not to be confused with the observation of the previous pure-strategy experiments where there was a significant difference across deadlines. As was shown in the results of $group4$ in long term deadlines, the *collective* final average utility of a strategy when summed and averaged across all other strategies (methodology of the pure-strategy experiments) *does* increase. However, the analytical unit of these experiments are average *joint* utilities for a *pair*, rather than a collection of strategies.

5.4.4.2 Mixed1 Strategy Utility Results

The expectations for the results of these experiments are summerised by the following hypotheses:

> **Hypothesis 8:** *A weighting policy that allows all tactics an input into the decision making results in a larger number of outcomes that are closer to being equitable, than one that only considers a single tactic.*
>
> **Hypothesis 9:** *The more equal this weighting of each of the tactics for* both *agents: i) the more equitable the final outcome and ii) the fewer the number of outcomes that lie off the constant-sum line. That is, variation of tactic weightings by either party results in less fair outcomes and more outcomes that lie off the constant-sum line.*

Hypothesis 8 states the intuition that in decision making a combination of tactics (a mixed1 strategy) is better than a single tactic (pure-strategy). The argument is as follows. In the given set of tactics (or "forces"), $boulware, linear, conceder$ and *titfortat*, two ($linear$ and $conceder$) concede at different rates (possibly three, *titfortat* given the other is a $conceder$ or $linear$) and one (possibly two—*titfortat* encountering a $boulware$) does so at a relatively much slower rate. Therefore, if equity, or some fair joint utility, is required then, as was shown in the results of the previous section, only encounters between a few pairs of *pure* concessionary strategies will achieve this expectation. Indeed, overall, encounters between all of the pure strategies will lead to outcome utilities that have a distribution within the space of possible deals that is more variable since the outcomes between agents will be based on *individual* tactics. For example, a pairing of tough and tough pure-strategies will result in a final joint utility that is significantly different to a pairing of conceder, conceder pure-strategies. Variability in the final average utilities is to be expected (since some pure strategies will reach the reference point, but encounters between others will not). On the other hand, encounters in mixed1 strategies are expected to be comparatively less varied, since they are no longer between unique tactics, but a *combination* of tactics. For example, to reach a fair solution an agent in mixed1 experiments does not have to "wait to meet" an agent who is adopting a pure $conceder$ or $linear$ strategy; $conceder\ and\ linear$ pure strategies are present, to some degree, in *all* mixed1 strategies of the other agents.

The expectation over the effects of this relative weighting of a tactic compared to the other tactics (Ω) is given in hypothesis 9. This hypothesis captures the expectation that the more equal the weighting of all the tactics by *both* agents (a resultant force that lies equally between the tough and conceder tactics, by both agents) then the more likely the final outcome is to be an equitable one. This is expected because when the distribution of tactic weights are less varied by both agents, then when these two agents meet the $boulware$, *titfortat* component of their strategy pairs are more resistant to concession. However, this resistance is compensated for by the $conceder, linear$, *titfortat* components which approach the cross over of offers at a quicker rate. Derivable from this argument is the expectation that inequality in the weight of tactics, by either party, should result in more variation of outcomes, similar to pure strategies—departure from an equal weighting of tactics, by either party, should result in outcomes that resemble more closely the results from pure-strategies. Variation in these experiments will be quantified with respect to the reference point and the departure of outcomes from the constant-sum line.

Figure 5.20 shows the final average agent utilities for mixed1 strategies in short term deadlines. Figure 5.20 A represents the observed final average utility outcomes for the benchmark $player$ and $opponent$ independent variables (figure 5.11). Compared to the pure-strategy results of figure 5.19, two patterns can be observed from the collected data that support hypothesis 8. Firstly, the center of the distribution of outcomes is closer to the reference outcome. Statistically the center of the distribution of points lying on or close to the constant-sum line is of a higher value of 0.479 with a lower standard deviation of 0.0077 as compared to a standard deviation of 0.25 for the points in the pure experiments. Almost all the points lie on the constant-sum line—compared to pure strategies, the number and magnitude of points lying off the constant-sum line is much lower (there are no longer any groups of outcomes). Specifically, the largest magnitude “breakaway” is for encounters between a $tough\ player$ and a $tough\ opponent$. Other previously breakaway outcomes (($tough$, *titfortat*), (*titfortat*, $tough$) and ($linear, tough$)) are much closer to the constant-sum line and the reference point than the results observed for pure experiments. Therefore, in such an environment, a combination of tactics (hypothesis 8) does indeed appear better than using a single tactic in generating offers in responsive mechanisms. Thus agents do not have to wait to “meet” a concessionary strategy to reach fair deals since all strategies have some degree of concession incorporated in them.

Hypothesis 9 is tested by changing the environment from a benchmark $player$ and $opponent$ to an $opponent$ with a higher Ω value (figure 5.20 B, as specified by the independent variables of figure 5.12). Two patterns are observable in the collected data. Firstly, the center of the distribution moves away from reference point towards the player in the south easterly direction (this increase of the distribution variation along the constant-sum axis will be referred to as “elasticity” of data points). The overall direction of the observed shift is towards higher utilities for the $player$. Statistically this corresponded to the total average of points lying on or close to constant-sum line value of 0.436 with a standard deviation of 0.11. Thus

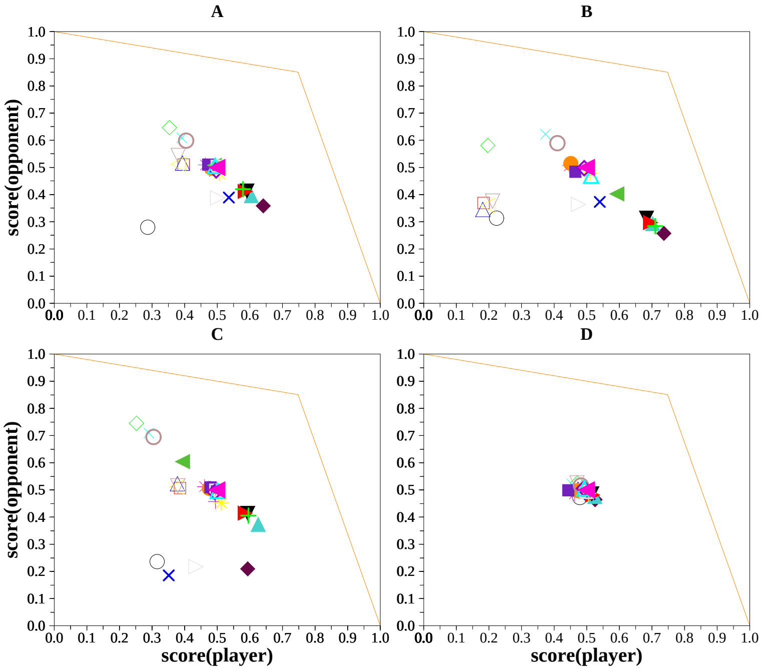

Figure 5.20: Comparative Average Utility For Mixed1 Strategies in Short Term Deadlines. A) Benchmark B) Opposition Increased Ω_t, C) Player Increased Ω_t D) Both Decreased Ω_t.

the outcomes that lie on the constant-sum line are of relatively higher value to the $player$. This pattern is expected from hypothesis 9 since with higher Ω values the $opponent$ is more concessionary for conceder strategies ($linear, conceder, titfortat$). This means that a shift should occur away from the $opponent$ reference point and towards the $player$.

The second observation is that the change in the environment (increased Ω_t for the $opponent$) produces more breakaway final outcome utilities from the constant-sum line than the benchmark experiments. Furthermore, the observation closely resembles the outcome distribution of pure-strategy experiments, supporting the second part of hypothesis 9—unequal weightings of tactics by either agent result in more outcomes that lie off the constant-sum line. Again, like the pure experiments the breakaway points consist of encounters between a $tough$ $opponent$ and a $tough$, $linear$ or *titfortat* $player$.

Hypothesis 9 is given symmetric support when the $player$ has a higher Ω value and the $opponent$ is specified by the benchmark values. This environment is described by the independent variables in figure 5.13. Results are shown in figure 5.20 C, the converse of 5.20 B. Once again, there is an elasticity of data points, but in an opposite manner to the previous environment. However, the relative movements are more towards the $opponent$ this time ($opponent$ statistical average for outcomes lying on or close to the

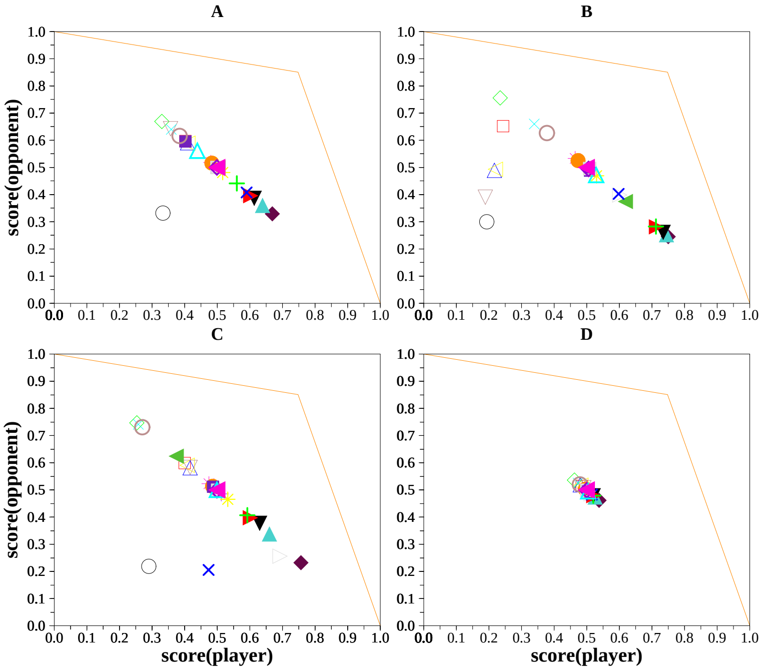

Figure 5.21: Comparative Final Joint Average Utility For Mixed1 Strategies in Long Term Deadlines. A) Benchmark B) Opposition Increased Ω_t, C) Player Increased Ω_t D) Both Decreased Ω_t.

constant-sum line increased from 0.436 to 0.51 with a standard deviation of 0.109). As before, the only breakaway points are encounters involving a $tough\ player$.

Hypothesis 9 is positively supported when *both* agents' environments are changed from the benchmark environment to one where the value of Ω_t is decreased (from the benchmark level) to a level where all other tactics have more of an input in decision making (independent variables shown in figure 5.14). The final outcomes across all strategies almost converge to the reference point (figure 5.20 D), corresponding to a total final joint average utility of 0.497 with a standard deviation of 0.015, the lowest standard deviation in the results thus far. Thus, the more equal the weighting of all tactics, by both agents, the more closer the final agreement is to the mid-point of the intervals. This is because some tactics function to reach the minimum of the interval values ($conceder, linear$), whereas others function to remain at the maximum of interval values ($tough, titfortat$). The resultant position reached is the mid-point of the interval.

Overall, the results imply the causal relationship that i) a combination of tactics outperforms pure strategies and ii) a near equal combination of the possible set of tactics by both agents results in better social outcomes (figure 5.20,D) than a differential combination policy of tactics (figure 5.20, A, B, C).

Figure 5.21 shows the same set of experimental environments as figure 5.20, but now the deadline to reach an agreement is extended from a short to a longer term. For the benchmark cases the outcomes further support hypothesis 8 and 9. The results are now more evenly distributed along the constant-sum line and with less breakage (with the exception of $(tough, tough)$ encounters) than short term deadlines (summed total average of 0.51 with a standard deviation of 0.096). The implications of this observation are that: i) when given enough time to negotiate, the strategies are almost uniquely distinguished by the solution point they reach (hence a more even distribution of outcomes) and ii) the sum of these deals are all almost 1.0. Whereas in pure-strategy experiments an increase in time deadline (figure 5.19 A and B) does not significantly increase the sum of the joint outcomes (resulting in outcomes that lie below the constant-sum line), the same change in the environment results in better joint outcomes.

The same elasticity pattern is observed as previously when either the $opponent$ or the $player$ negotiates with higher Ω values, shown statistically by an increase in average utility for the $opponent$ from 0.414 (standard deviation of 0.079) to 0.50 (standard deviation of 0.13) between figures 5.21 B and C respectively. Once again, the outlying outcomes are encounters between strategies that are slower to reach the cross over point of offers.

Finally, once again the best outcomes are observed when *both* agents' environment are changed from the benchmark environment to one where the value of Ω is decreased (from the benchmark level). Again, the final outcomes across all strategies almost all converge to the reference point (figure 5.21 D), corresponding to a total final joint average utility of 0.499 with a standard deviation of 0.017. These combined observations shown in figures 5.20 D and 5.21 D imply that outcomes cannot be distinguished when both agents adopt an almost equal weighting of possible tactics (low values of Ω for strategy magnitudes). This means that outcomes are independent of the strategies the agents select (a collapse of *all* points on to the reference). This is because all strategies in this environment are defined as an almost equal weighting of tactics, where the difference between the weightings for each strategy is insignificant. Hence all strategies are almost equal with small variations (shown in the data by the magnitude of the standard deviation of the results). The expectation for this result is stated in hypothesis 8; in this environment the point of the crossover between the offers is reached by almost *equally* combining the suggestions of all tactics. Thus whereas the $boulware$ tactic may suggest different offers, its input is approximately only one quarter or, at maximum, a third of the final decision ($[0.23, 0.3]$, see figure 5.14). On the other hand, the concessionary tactics may suggest concession rates that are very different to a $boulware$ tactic, but nonetheless they are also only a quarter or a third part of the final decision. The overall effect of the strategies in this environment is an *equal* integration of the suggestions of different tactics into a single concession rate. In so doing, each of the individual differences between tactics are ignored and a new combined concession rate is computed. As will be shown later, this hypothesis, that in this environment strategies integrate different concession rates into a single concessionary rate, is supported by an almost constant communication load across all strategy pairings shown in figure 5.26.

5.4.4.3 Mixed2 Strategy Utility Results

The expectation for the results of these experiments are summerised by the following hypothesis:

> **Hypothesis 10:** *Modification of a strategy during the course of negotiation will result in higher valued and fairer social outcomes than non-modification.*

This hypothesis states that the combination of i) considering a number of tactics in decision making and ii) modifying this consideration, should result in outcomes that maximize the equity joint utility of the outcome. This is expected because the update rule should change the weights of each tactic in such a way as to reach the cross over in the contract score according to how close the offers are to one another. Thus, at the beginning of negotiation it is expected that offers are dissimilar. The degree of dissimilarity in turn depends on the starting position of the resultant shown in figure 5.10 (or the initial Γ matrix). However, the resultant is incrementally adjusted (according to rule 5.2 whose actions are dependent on the evolving similarity between offers) towards the tough end of the spectrum (Ω) by both agents as offers become more similar to one another. This process continues until offers converge. Thus, if both agents are implementing rule 5.2 for update of weights and their interval values are perfectly overlapping, then final outcomes should be closer to the reference point than mixed1 strategies. The observed final joint average utility outcomes are shown in figure 5.22 for dynamic strategies (mixed2) in short term deadlines. The overall observation for all the Ω variations (independent variables shown in figures 5.11, 5.12, 5.13, 5.14) is that *all* of the outcomes are distributed on the constant-sum line with no breakaway points. The average of the utility distributions along the constant-sum line are now $0.50, 0.489, 0.533$ and 0.488 for benchmark, $opposition, player$ and both decreased Ω levels respectively, with standard deviations of $0.0866, 0.123, 0.10$ and 0.035 respectively. The same averages for mixed1 strategies were $0.479, 0.384, 0.436$ and 0.497 for benchmark, $opposition, player$ and both decreased Ω levels respectively. The combined observations that there are no outlying breakaway outcomes (hence all outcomes are maximized) and there is an increased final joint average utility distribution around the reference point (hence higher equitable outcomes) gives support to hypothesis 10. Thus, changing strategies in short time deadlines results in better joint outcomes than a mixed1 strategy. For example, a $tough$ mixed1 strategy throughout the negotiation results in breakaway points, but changing from being concessionary to a tough type strategy resulted in better social outcomes.

Finally, figure 5.23 shows the results for the same set of environments but for longer term deadlines. Once again there are no breakaway outcomes with average distributions along the constant-sum line values of $0.509, 0.448, 0.557$ and 0.499 for benchmark, $opposition, player$ increased Ω and both decreased Ω levels respectively, with standard deviations of $0.04, 0.082, 0.089$ and 0.0012. The interesting point to note

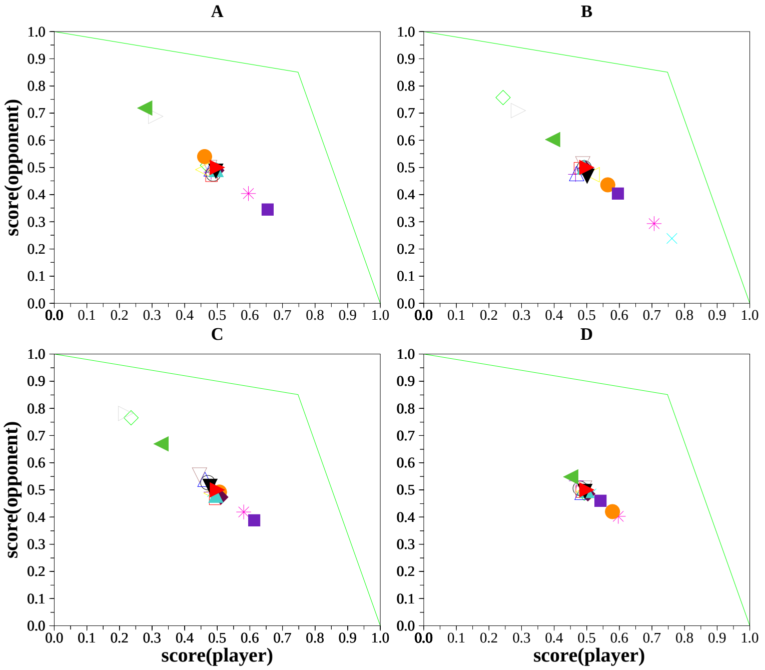

Figure 5.22: Comparative Final Joint Average Utility For Mixed2 Strategies in Short Term Deadlines. A) Benchmark, B) Opponent With Increased Ω , C) Player With Increased Ω, D) Both With Decreased Ω.

is that when all tactics are weighted almost equally by both agents (figure 5.23 D), the final outcomes converged exactly to the reference point (average distribution of 0.499 and standard deviations of 0.0012, the lowest in the experiments). Thus in environments where both agents weight their tactics uniformly (figure 5.23 D) the final outcome is independent of the individual strategies. This result is expected from the combination of hypothesis 9 of mixed1 strategies and the behaviour of the update rule. That is, when both agents weight each tactic almost equally, then the initial concession rate to the cross over of values is computed as the combination of both concessionary and non-concessionary tactics, into a unique concession rate that is the resultant of the combination. This initial concession rate is then updated by the rule given in equation 5.2 independently of the type of strategy, selecting a convergence policy to the cross over of offers which is dependent of the context (the similarity) of negotiation.

5.4.4.4 Pure-Strategy Cost Results

In this section the hypotheses and observations over the dependent variable $cycles$ are presented for the pure strategies. Recall that, unlike the previous pure-strategy experiments reported in section 5.3, the analytical unit of these experiments is average cost for a *pair* of strategies, rather than a collection of strategies.

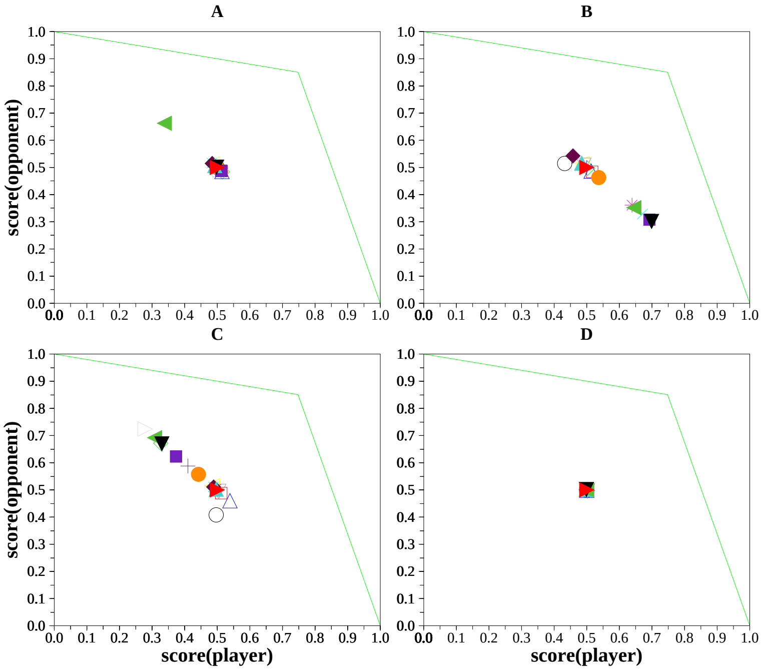

Figure 5.23: Comparative Final Joint Average Utility for Mixed2 Strategies in High Time Deadlines. A) Benchmark B) Opposition Increased Ω_t, C) Player Increased Ω_t D) Both Decreased Ω_t.

The dependent variable $cycles$ directly measures the communication load a strategy incurs during the negotiation. The results for mixed1 and mixed2 strategies are presented in the two subsequent sections. Due to legend space restrictions, the strategy labels on the x axis have been abbreviated to b, l, c, t for $tough$, $linear$, $conceder$ and *titfortat* strategies.

The intuitions and expectations about the communication load of a pure strategy are captured by the following hypothesis:

> **Hypothesis 11:** *Pure strategies that concede comparatively less slowly will result in correspondingly higher communication costs.*

This hypothesis is simply based on the fact that some tactics ($boulware$ or *titfortat* when it encounters a $boulware$) approach the minimum of the interval values less slowly, thereby prolonging the negotiation thread. The support for hypothesis 11 is given in the observed results of figure 5.24 A and B, showing the observed communication load for different pure-strategy pairings in short and long term deadlines respectively. The first support for the hypothesis is deduced from the inverse observation that encounters between any strategy and a $conceder$ result in fewer exchanges of offers than other combinations, indepen-

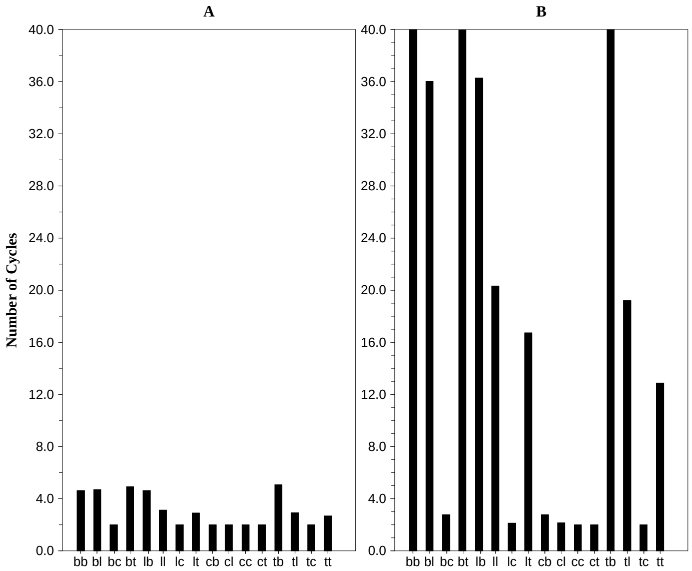

Figure 5.24: Communication Loads For Pure Strategies. A) Short Term Deadlines B) Long Term Deadlines.

dently of the environment. Furthermore, more offers are exchanged in longer term deadlines. However, whereas in most pairings the amount of communication increases with increasing time limits, encounters with a *conceder* result in almost constant communication load. That is, encounters with a *conceder* result in the same number of offer exchanges independently of the time limits. Finally, positive confirmation of hypothesis 11 is obtained with the observation that the highest number of offers exchanged is between the *tough* and *titfortat* pairings, the same pairings in the final joint average utility observed data that exhibited breakaway patterns from the constant-sum line (figure 5.19). Taken together, these results indicate that encounters between pure strategies that have a slower rate of approach to the interval values not only result in poorer social outcomes, but also incur a high communication overhead.

5.4.4.5 Mixed1 Strategy Cost Results

The intuitions and expectations about the communication load of a mixed1 strategies are captured by the following hypothesis:

> **Hypothesis 12:** *In the general case, a strategy that combines tactics will result in an increased number of negotiation rounds. Specifically, the amount of communication used is a function of the amount of mixture involved between tactics that reach intervals slowly or rapidly.*

The above hypothesis is based on the expectation that when only a single tactic is selected for generation of offers (a pure-strategy) then, as confirmed in the previous section, those tactics that have a slower concession rate to the interval values will result in a higher number of offer exchanges. However, when tactics with a different concession rate to the interval values are combined, then the number of exchanges will generally be greater than the pure strategy case. A higher number of exchanges are expected because concessionary tactics are now combined, to some degree, with less concessionary tactics like $boulware$ and *titfortat*. Therefore, since each strategy has an element of less concessionary behaviour then more communication is to be expected. This is a general hypothesis since the specific number of offers exchanged depends on the "amount of this mixture" (or Γ matrix) policy of the strategy. The overall expectation is that fewer exchanges of offers are likely when both agents "move" the resultant force (figure 5.10) of their tactic combination from a $boulware$ tactic to a $conceder$ tactic.

Figure 5.25 A, B, C and D show the observed results for the communication load of pairings of mixed1 strategy types in short term deadlines (for benchmark, $opposition, player$ increased Ω_t levels and both decreased Ω_t levels respectively). Hypothesis 12 is not supported in short term deadlines. The observed data suggests, similarly to $pure$ strategies, that in short term deadlines virtually all the encounters between all the different types of strategies take the same number of cycles to complete. This is also observed for $mixed2$ experiments (see figure 5.27 A, B, C and D) which are described in the next section. This result is due to the small window of opportunity constraining the time within which strategies must reach a deal (this sub-hypothesis is supported by the observation that in comparatively longer term deadlines strategies are differentiated, see figure 5.26). Because this "window" is small all strategies use almost all of the limited time to search for deals. A short term deadline is defined as $2 - 10$ ticks of a discrete clock. Therefore, strategies have on average 4 ticks of a clock to reach a deal. As shown in figure 5.25, nearly all strategies "consume" this available time. Therefore, a better differentiator of strategies in short term deadlines is not the communication load of the strategies, but rather the number of deals reached or their utilities, or a combination of both. This result is carried over to other strategies, where in short term deadlines the number of $cycles$ in negotiation is independent of not only the pairings of the strategies within a given type of strategy (pure, mixed1 or mixed2), but also across different types of strategies. Therefore, communication load can not be used as a decision criteria in short term deadlines. The agent may rely instead on other relevant criteria such as the intrinsic utility of the outcome or the number of successful deals reached. For example, if the utility of deals is used as a decision criteria for which strategy to select then, as shown by the results in figure 5.22, a mixed2 strategy can lead to better social outcomes. A significant effect of strategy pairings on the communication load is observed in patterns of data for longer term deadlines for both mixed1 and mixed2 experiment types (figures 5.26 and 5.28 respectively). The claim that the number of exchanges in a mixed1 strategy will *generally* be greater than the pure-strategy case is supported by an increased total average number of cycles across all strategies. Quantitatively, the total average number of cycles across all strategies are 17.18 for pure strategies (figure 5.24 B) and 22.58 for benchmark mixed1 strategies (figure

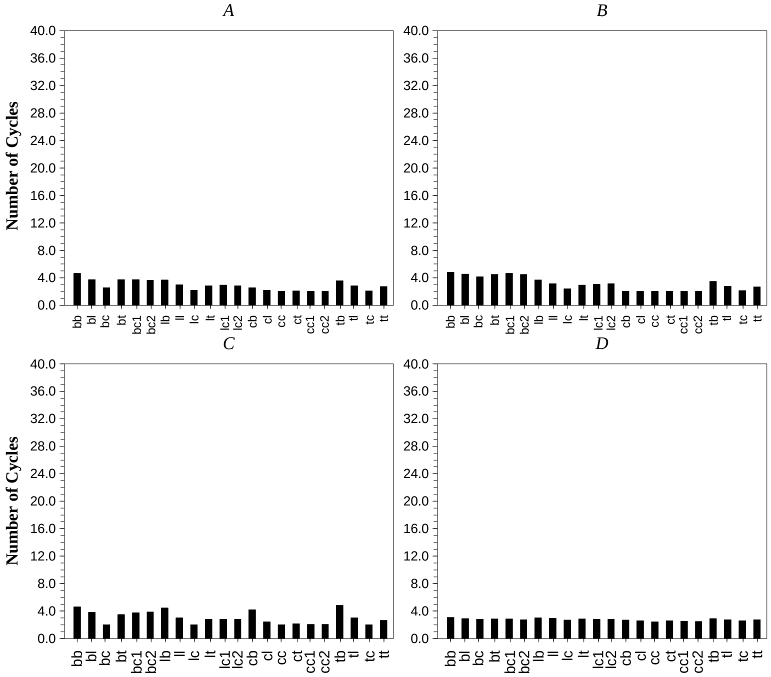

Figure 5.25: Communication Loads For Mixed1 Strategies in Short Term Deadlines. A) Benchmark B) Opposition Increased Ω_t, C) Player Increased Ω_t D) Both Decreased Ω_t

5.26 A). Hypothesis 12 is further supported by the observations when the $opponent$ (or conversely the $player$) had a higher Ω_t level than the benchmark environment (figure 5.26 B and C respectively). This environment tests the proposition that the specific number of offers exchanged depends on the "amount of mixture" involved (or Γ matrix) policy of the strategy. Compared to the benchmark case, increasing γ_{ij} (moving towards a γ_{ij} array distribution that resembles more closely the pure-strategy γ_{ij}), causes tactics that approach their interval quickly (or slowly) to decrease (or increase) the communication loads. For example, increasing the γ_{ij} of a $conceder$ tactic from the benchmark case results in a lower number of exchanges in negotiation (figure 5.26 B). Conversely, increasing the γ_{ij} of a $boulware$ tactic from the benchmark case results in an increased number of exchanges in negotiation (figure 5.26 B). Note also that the latter encounters are the group of pairings that exhibited breakaway from the constant-sum line (figure 5.21 B).

The results of mutual and uniform integration of concessionary tactics with less concessionary tactics by both agents in long term deadline environments is shown in figure 5.26 D. The expectation that the communication load of mixed1 strategies specifically depends on the amount of "mixture" of tactics is positively supported in figure 5.26 D, where an equal combination of concessionary and non-concessionary tactics re-

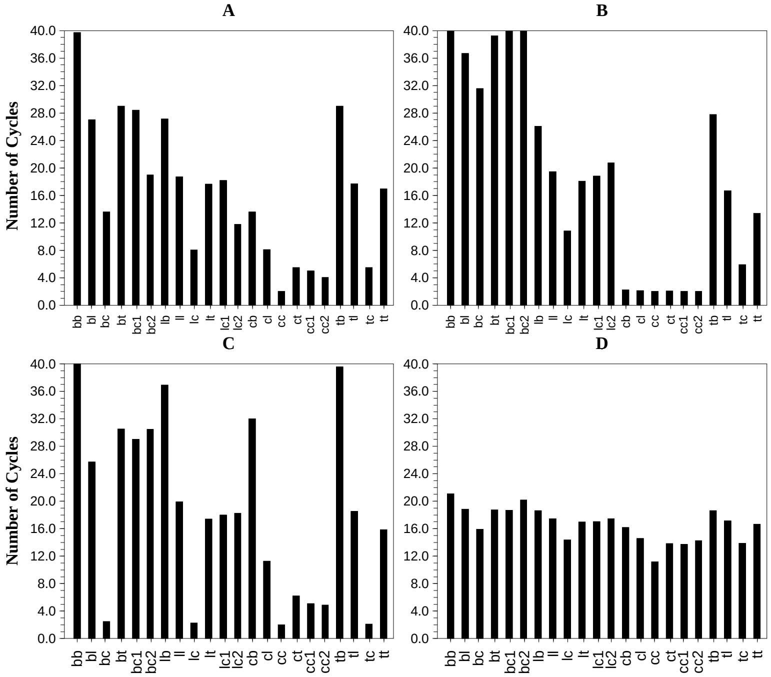

Figure 5.26: Communication Loads For Mixed1 Strategies in Long Term Deadlines. A) Benchmark B) Opposition Increased Ω_t, C) Player Increased Ω_t D) Both Decreased Ω_t.

sult in an increase in communication costs of conceder type strategies and a decrease in communication costs of tougher strategies.

5.4.4.6 Mixed2 Strategy Cost Results

The intuitions and expectations about the communication load of mixed2 strategies are captured by the following hypothesis:

> **Hypothesis 13:** *In the general case, dynamically changing strategies in the course of negotiation, according to some subjective function, will result in fewer negotiation rounds than static strategies.*

Hypothesis 13 has essentially the same form as hypothesis 12. However, the difference in the prediction is that in the general case a mixed2 strategy will result in fewer exchanges of offers. That is, in the types of environments considered in these experiments, the modification of the Γ matrix according to some subjective function (here the perceived closeness between offered contracts) should result in fewer exchanges of offers since the interval values of agents are perfectly overlapping. If the interval values are perfectly overlapping and agents begin their offers at the maximum of their interval values, then subsequent offers should quickly

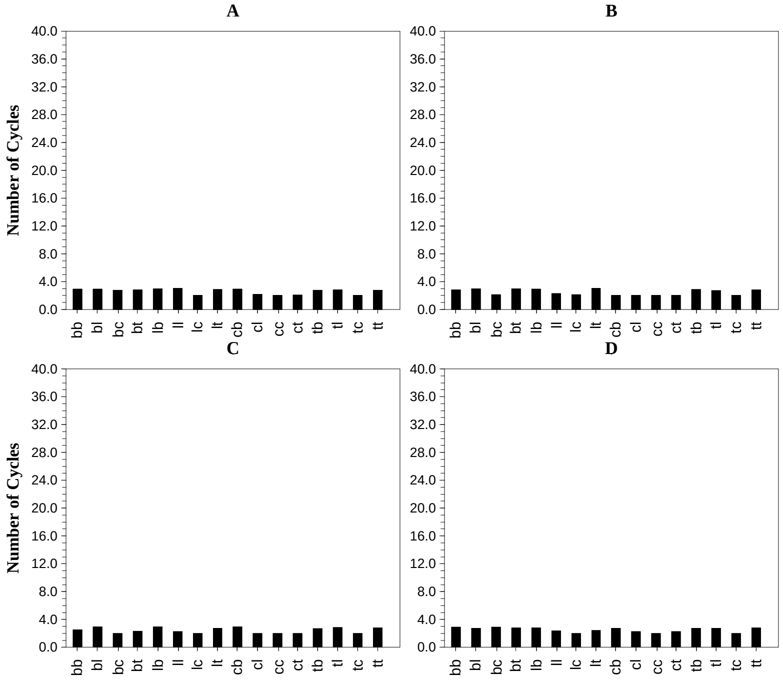

Figure 5.27: Communication Loads For Mixed2 Strategies in Short Term Deadlines. A) Benchmark B) Opposition Increased Ω_t, C) Player Increased Ω_t D) Both Decreased Ω_t.

become more similar when at least one agent makes a concession. Offers become similar quickly because the update rule 5.2 gives higher weightings to concessionary tactics when offers are not close to one another. In essence the update rule modifies the behaviour of each strategy with another tactic (concessionary or retaliatory) according to the perceived closeness of offers. If distances between contracts are large then a tactic that concedes is given higher importance. As the offers approach one another the similarity between offers increases, resulting in a higher weighting for $boulware$ tactics. The overall effect of these two rates of approach is to quickly approach the mid-point of the intervals, followed by a slower rate of concession until a cross over of offers occurs. In a mixed1 strategy, on the other hand, the rate of approach to mid-point is constant. For example, a tough strategy in mixed1 consists of approaching the interval at a rate that is constant and slow. This should naturally result in more exchanges of offers than an equivalent tough mixed2 strategy whose behaviour is to concede initially (because contracts are dissimilar—rule 5.2), but become tough as offers become more similar.

The observations and explanation of the results for the short term deadline environment (figure 5.27) have already been described in the section above. Figure 5.28 shows the final observed communication results for the mixed2 strategies with long term deadlines. The comparative data for benchmark cases of

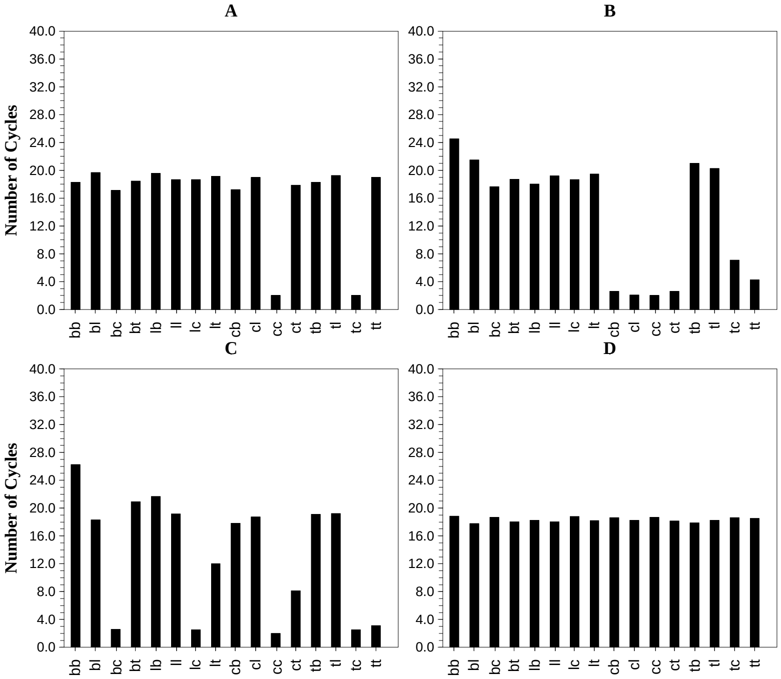

Figure 5.28: Communication Loads For Mixed2 Strategies in Long Term Deadlines . A) Benchmark B) Opposition Increased Ω_t, C) Player Increased Ω_t D) Both Decreased Ω_t.

mixed1 and mixed2 (figures 5.26 A and 5.28 A, respectively) supports hypothesis 13. For example, a tough mixed2 strategy engages in less communication than an equivalent tough mixed1 strategy. In *general*, a mixed2 strategy reaches an outcome in fewer rounds of negotiation. Statistically the final sum average of communication cycles for all benchmark mixed1 strategies (the general case) is 22.75, compared to the final sum average of 16.3 for the benchmark mixed2 strategies. This pattern is also repeated for cases when Ω_t of either the $opponent$ or the $player$ is increased, figures 5.28 B and C respectively. Finally, there is no significant observed difference in communication usage between mixed1 and mixed2 strategies when both agents weight the tactics smoothly and almost equally (figures 5.26 D and 5.28 D respectively). This result, in combination with others shown in figure 5.28 A, B and C, suggests that when tactics are mixed equivalently, offers are closer to the mid-point of the cross over (supported by the final joint utility observations in figure 5.23 A, B, C and D, where the final outcomes are very close to the reference point). Hence the update rule modifies all strategies slowly (a $boulware$ tactic) until cross over is achieved. This suggest that deliberation over which combination of tactics to use will result in better social outcomes (figure 5.23 A, B, C and D) than a static policy, and this can be achieved at the same communication cost.

5.4.4.7 Summary of Strategic Experiment Results

The above results for the three experimental classes confirm the initial proposition of the experiments—that $dynamic\ strategies \succ static\ strategies \succ pure\ strategies$, for the experimental dependent variables intrinsic utility and cycles.

The utility results show that decision making using pure strategies, when viewed from a global perspective (the equity or maximization of joint utilities represented by the reference point), results in the most variable set of utility outcomes. However, when tactics are mixed, but constant (mixed1 strategy), there are significantly lower variations in final average utilities. Furthermore, a more equal weighting by both agents results in final outcomes that most increase the maximization of equitable outcomes. In sum, as the mixture of tactics is made more equal by both agents, then the closer the final outcome gets to the reference point. Finally, changing this initial consideration (mixed2 strategy) results in the highest maximization of equitable outcomes.

Once again the variability of the communication load of the strategy is highest in the pure case. This can be seen by the fact that conceder pure strategies result in less communication load and, conversely, a tough strategy results in relatively more communication. In the case of mixed1 strategies, on the other hand, this variability in communication across strategies becomes dependent on the amount of mixture of the tactics. Thus the results show that when an agent places higher weighting on concessionary tactics, the communication load is minimal (also independently of time limits). Conversely, almost all of the communication resource is used by agents when they place more weight on the less concessionary tactics. Medium communication load, and less variability across strategies, is observed when both agents weight each tactic equally. Finally, a dynamic strategy according to the policy that the concession tactic be given more weight when offers are not similar to one another, results in the least overall communication resource usage.

Overall the implications of these results, from the perspective of configuring an agent, using the wrapper with the current set of available tactics, is that the agent designer should expect the following:

- Pure-strategies have the largest effect on the interactions. Specifically, if an agent is configured to interact with a pure-strategy then variability should be expected in: i) the final utility of outcomes, with only a few combinations of pure-strategies resulting in better social outcomes, and ii) the overall communication costs.

- If an agent is configured to interact with a mixed and static strategy then the designer should expect: i) less variability in the final utility of outcomes with relatively more pairings of mixed strategies resulting in better social outcomes, but ii) with a higher overall communication cost than pure-strategies because concessionary and non-concessionary tactics are now mixed (thereby increasing the overall communication cost).

- If an agent is configured to interact with a mixed and dynamic strategy (given by the update rule 5.2) then the designer should expect: i) the least variability in the final utility of outcomes with relatively more pairings of mixed strategies resulting in better social outcomes than pure or static strategies and ii) an invariant, and almost average, overall communication cost when compared with pure or static strategies.

5.5 Trade-off Experiments

The previous two sections empirically investigated the behaviour of the responsive mechanism. In this section the trade-off component of the wrapper is empirically evaluated. The aim of these experiments is to evaluate the kernel of the trade-off algorithm (presented in section 4.5.2.3) by investigating its *parameters* in generating a *single* offer. Therefore these experiments are intended to discover the behaviour of the algorithm and assist negotiating agent designers by providing guidelines about the possible outputs of the algorithm given the inputs that need to be supplied by the designer. This input is the information an agent has about the other agent and it needs to be provided by the designer as knowledge in the acquaintance model (*AM*) component of the wrapper, shown in figure 1.1. These experiments will be referred to as *single-offer* experiments.

The next section, in turn, reports on the experiments that evaluate the *process* of negotiation when agents use a combination of, through the use of meta-strategies, trade-off and responsive mechanisms. The process of both agents solely making trade-offs can not be investigated because negotiation will always be unsuccessful. Making trade-offs means offers have non-diminishing scores, hence cross over of offers, a condition for accepting an offer, can not occur. Therefore the designer of an agent is provided with a higher level interaction analysis of the behaviour of the trade-off mechanism when it interacts with a combination of other mechanisms. These latter experiments are referred to as the *meta-strategy* experiments.

Whereas the aim of the single-offer experiments is to investigate the kernel of the trade-off algorithm, in the meta-strategy experiments the subject of the investigation is the dynamics of the trade-off algorithm when interacting with other mechanisms.

5.5.1 Experimental Independent Variables

The experimental independent variables are reported in this section. Both the single offer and meta-strategy experiments share a common set of independent variables, therefore, to avoid repetition in the next section, the set of shared independent variables is presented in section 5.5.1.1 below. Next the independent variables unique to the single-offer experiments are presented in section 5.5.1.2.

5.5.1.1 Experimental Independent Variables for Both Single-Offer and Meta-Strategy Experiments

The negotiation environment is left unaltered from the dependent variables described in the strategy experiments (figure 5.8) in order to assist the comparison of the results between the trade-off and meta-strategy experiments with the responsive experiments, presented earlier in section 5.4.4. Briefly, the environment in the single-offer and meta-strategy experiments consists of bi-lateral negotiations between agents categorically labelled as $player$ and $opponent$, who negotiate over multiple quantitative issues $[price, quality, time, penalty]$. The interval values for these issues are perfectly overlapping (see equation 5.1). The $player$ assigns $[0.1, 0.5, 0.25, 0.15]$ and the $opponent$ assigns $[0.5, 0.1, 0.05, 0.35]$ as the importance of these issues.

The other input variables of the trade-off algorithm are the discriminatory power and the magnitude of the difference between the input and output of the criteria function (equation 4.6). The criteria function used (equation 5.3) is the same as the one presented for the responsive Γ update rule 5.2. Like the responsive experiments, ϵ is also fixed at 0.1 for all issues in order to be quite discriminatory. Also, different α values are fixed to be equal for all issues, $\alpha^{price} = \alpha^{quality} = \alpha^{time} = \alpha^{penalty} = 1$, so as to have linear criteria functions $(h_i^\prime s)$, having equal discrimination power across the issue's interval values.

5.5.1.2 Single-Offer Experimental Independent Variables

The independent variables that are specific to the single-offer experiments are:

1. the number of children generated at each step in hill-climbing to the iso-curve (N in the trade-off algorithm, section 4.5.2.3)

2. the number of steps taken to reach the iso-curve (S in the trade-off algorithm, section 4.5.2.3)

3. the information that is available to an agent regarding the importance (or weight) the opponent places on each issue in computing the contract's value (equation 4.5) and

4. the $opponent's$ and $player's$ last offers (x and y in equation 4.4).

Values for the first and second variables control the amount of search performed by the trade-off algorithm. Experiments are run where the number of children are selected from the set $\{5, 100, 200\}$. The number of steps to the iso-curve is selected from the set $\{1, 40\}$. The concrete numbers for both the number of children and the number of steps to the iso-curve individually signify very little. However, the significance of these values is the relative relationship between them. Thus more computation is involved when the trade-off algorithm generates 200 rather than 5 children at each iteration, or when it takes a *larger* number of steps to the iso-curve. The expectation, as will be shown below, is that more computation should result in better outcomes.

The third independent variable attempts to calibrate the relationship between the performance of the trade-off algorithm (in particular, how similarity is computed) given an agent's subjective estimates of the likely importance weightings of the other agent. This subjective estimation over others' weights is stored as information in the *AM* component of the wrapper. Thus, to compute whether two offers are similar, an agent has to make some subjective, and possibly incorrect, decision about how the other views the importance of an issue. Specifically, in single-offer experiments an agent can have either perfect, partial, imperfect or uncertain information on how the other agent weights the issues that are input into its similarity function (equation 4.5). The agent chosen to perform the single-offer tradeoff is the $player$. Then, in experiments with perfect information, the algorithm, in computing similarity, is given the $opponent's$ weights for different issues (i.e. $[0.5, 0.1, 0.05, 0.35]$, cardinally correct information). Partial information games are where the algorithm is given the correct order of importance but not the actual issue weights (i.e. $[0.7, 0.09, 0.01, 0.2]$, ordinally correct information). Imperfect games represent the situation where the algorithm is given incorrect information about the other's weights (i.e. $[0.1, 0.2, 0.5, 0.2]$, incorrect information). Finally, uncertain information games represent cases where the algorithm is given undifferentiated weights for each issue, in this case $[0.25, 0.25, 0.25, 0.25]$. The output of the trade-off algorithm can then be assessed when supplied with different types of information.

The final independent variables in these experiments are the input contracts x and y (see equation 4.4) representing the $player's$ and the $opponent's$ last offer respectively. Given the interval values in equation 5.8, contract x is set to $[15, 28, 25, 8]$ and y to $[18, 10, 45, 3]$. Given each agent's weights and their linear scoring function (described in section 5.4.2.2), the agent's valuation of these two contracts are:

$$V^{player}(x) = 0.835, V^{player}(y) = 0.195$$

$$V^{opponent}(x) = 0.344, V^{opponent}(y) = 0.8$$

meaning that negotiation can continue since there is no cross over of offers yet, each agent still prefers their own offer over the other's latest offer.

5.5.2 Experimental Procedure

The experimental procedure consists of inputting two contracts, representing x and y, into the algorithm under different combination of the other three independent variables (number of children, number of steps to the iso-curve and the information levels) and observing the utility execution trace of the algorithm for *an* offer from the $player$ to the $opponent$. All input contracts (x and y) are subject to the general constraint that $V^{player}(y) < V^{player}(x)$ and $V^{opponent}(x) < V^{opponent}(y)$. This ensures trade-off is possible by ruling out all those contracts that are already of a higher value to either party. A control set is also generated by choosing the preferred child randomly at each step approaching the iso-curve (as opposed to using the similarity criteria).

5.5.3 Hypotheses and Results

The hypothesis in a single-offer experiment is given in terms of the input and output of the trade-off algorithm. The input is the set of importance weights of the other agent (perfect, partial, imperfect and random) and the output is a contract that has the same score to the agent, but some other score to the other agent. Specifically, the hypothesis is:

> **Hypothesis 15:** *The greater the exploration of the space of possible deals, the better the output of the algorithm from the perspective of the other agent.*
>
> *Furthermore, the quality of the algorithm's output (the score of the contract to the $opponent$) is directly dependent on the quality of information input—the better the information, the better the outcome quality.*

The hypothesis simply states the intuition that a more refined search of the possible space of contracts should result in selecting and offering a contract that has more value to the other agent. Furthermore, this search should be directly affected by the information the algorithm has about the other's issue importance rankings.

Figure 5.29 and the top row of figure 5.30 show the results of varying, under different information inputs, the number of children generated in single-offer experiments when the number of steps to the iso-curve is set to 40. The bottom row of figure 5.30 represents the case where the number of children is set to 100, but the trade-off algorithm computes the iso-contract in a single step. The dot-dash line represents the execution trace of the random control, the solid line emanating from y the similarity based trade-off execution trace, and the line joining $(0,1)$ to $(1,0)$ the pareto-optimal line. The output of the algorithm, $x^{'}$, is shown in figures 5.29 and 5.30 as an unfilled circle and square for the algorithm that selects the next child in each step based on similarity or random criteria respectively.

Three major patterns are observed that directly and indirectly support hypothesis 15. Direct support is given by the first observation that when moving to the iso-curve if the space of possible contracts is not explored sufficiently, 5 children (figure 5.29 top row) or 1 step (figure 5.30 bottom row), then the gains of the $opponent$ are at best insignificant and at worst negative. More specifically, only when the $player$ has perfect information about the $opponent$'s evaluations and the trade-off mechanism operates in 1 step with 100 children will the mechanism improve the offer (from the $opponent$'s perspective) (figure 5.30 E). The next best contract for the $opponent$ is when it has the same value as x (figure 5.29 A). All other contracts generated by the $player$ when not fully exploring the search space (figures 5.29 B,C,D and 5.30 F) have lower value to the $opponent$ than x.

However, the $opponent$'s benefit increases as the algorithm performs more search (from 5 to 200 children in 40 steps—figure 5.29 top row [5 children], bottom row [100 children], and figure 5.30 top row

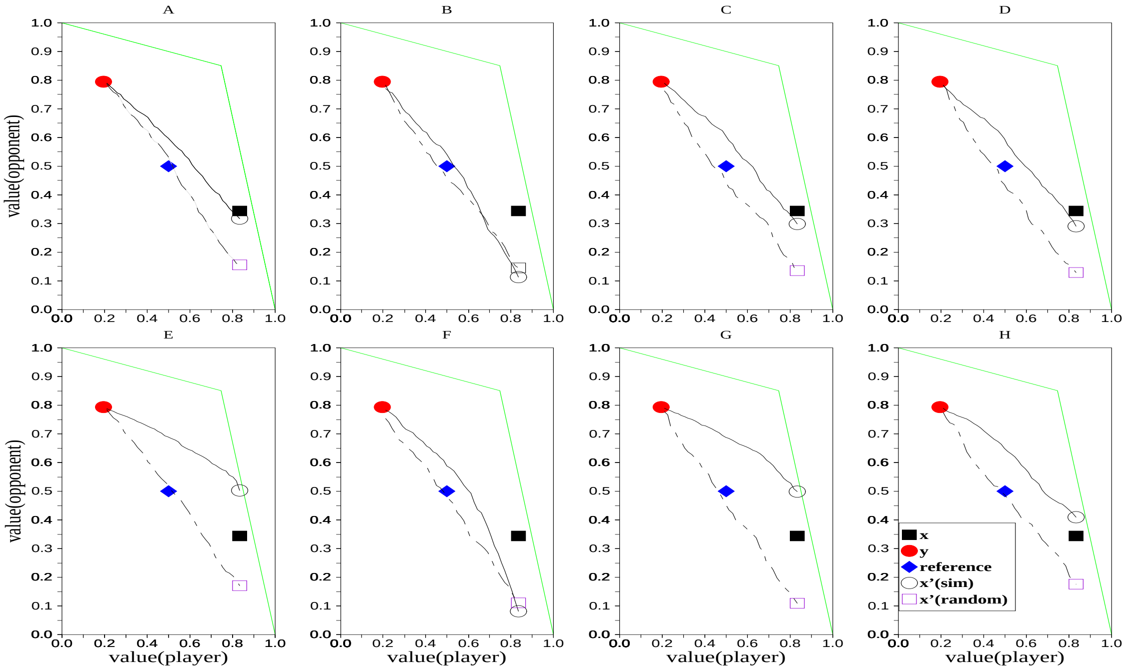

Figure 5.29: Tradeoff Algorithm Experiment: Data for 5 Children in 40 Steps (First Row) and 100 Children in 40 Steps (Second Row). A) & E) Perfect Information, B) & F) Imperfect Information, C) & G) Partial Information, D) & H) Uncertain Information.

[200 children]). Thus, generating more children does indeed increase the utility of the opponent. However, the data suggests there is a point above which generation of more children does not increase the utility of the opponent. This is observed in the lack of any significant difference between perfect and partial information outcomes within either the 100 and 200 children (40 steps) result categories (compare figures 5.29 E, F, G and H with 5.30 A, B, C and D). Furthermore, the expectation, as stated by hypothesis 15, that the more accurate the information about the weights of the *opponent* are, the better the contract score for the *opponent* is supported by the observation that the utility to the *opponent* is indeed increased when the algorithm is increasingly supplied with more correct information about the $opponent's$ weights (seen as increasing utility) from incomplete to uncertain information classes. However, the hypothesis is rebutted for perfect and partial information cases (compare 5.29 E with G or 5.30 A with C). This lack of significant differences between contracts selected under perfect and partial information conditions indicates that the algorithm requires only partial ordering information, rather than perfectly cardinal orderings, in order to compute outcomes that are better for the *opponent*. This is because the absolute differences in magnitude between the perfect and partial information classes is small ($[0.5, 0.1, 0.05, 0.35] - [0.7, 0.09, 0.01, 0.2] = [0.2, 0.01, 0.04, 0.15]$), resulting in input variables that are not significantly different. The chosen value

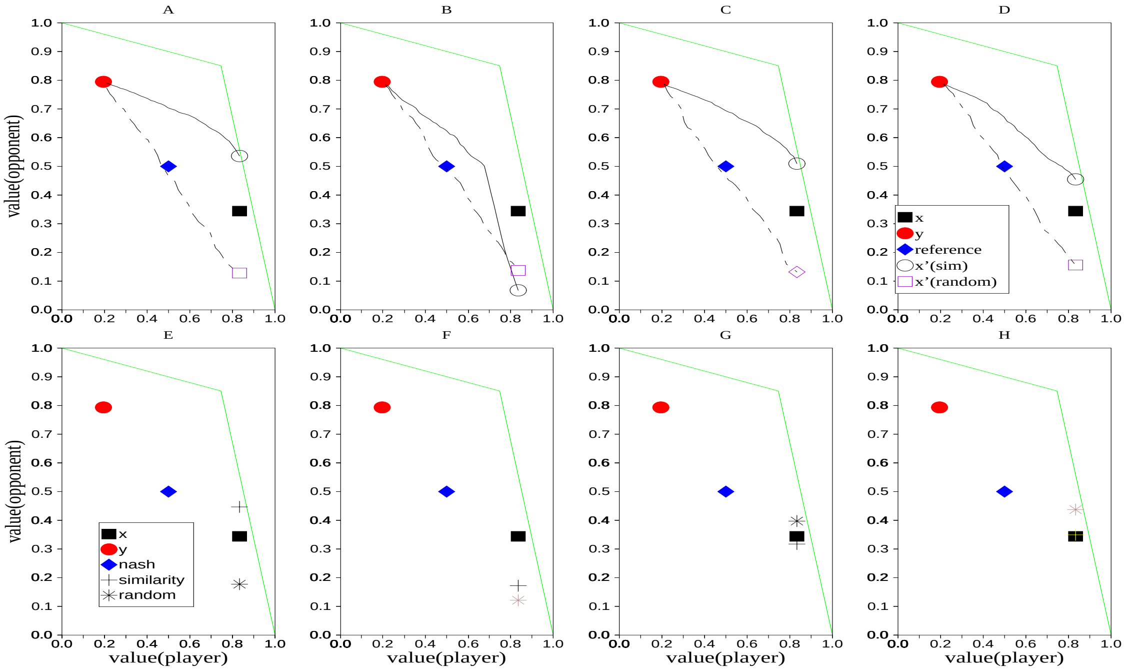

Figure 5.30: Data For 200 Children in 40 Steps (First Row), and 100 Children in 1 Step (Second Row). A) & E) Perfect Information B) & F) Imperfect Information, C) & G) Partial Information, D) & H) Uncertain Information.

for the partial weight estimation can not be made significantly different from the perfect weight estimation values because the actual values of the partial estimates are constrained both at the upper and lower limits by the perfect and uncertain weight estimation values.

Positive support about the relationship between the quality of the input and the resultant output is given in the final observation that, for all environments and variable combinations, imperfect information (figure 5.29 B and F, and figure 5.30 B and F) results in significantly poorer outcomes for the *opponent* than all the other information classes. This is only to be expected since the search is directed towards erroneous directions when the information supplied about the other agent is incorrect.

Note, in nearly all cases, the similarity based trade-off out performs the policy of randomly selecting a child for the next step towards the iso-curve. However this pattern does not hold for the cases of reaching the iso-curve in one step under partial and uncertain information environments (figure 5.30 G and H). Given an offer is generated in 1 step, this is due to chance, rather than randomness being a better strategy in this type of environment (supported by the consistently poor performance of the random selection strategy in the experiments where the number of steps to the iso-curve is set to 40, figure 5.29 C, D, G and H, and 5.30 C and D).

In summary, these results indicate that unless agents know, at least partially, the importance the other agent attaches to an issue, then the best policy for computing trade-offs is to assign uncertain weightings to all issues. These weightings can then be updated by some learning rule towards partial or perfect information models, since a) information models are private and b) erroneous predictions can result in poorer outcomes. Furthermore, engaging in trade-off negotiation, particularly with a high search factor by both parties, results in higher joint gains.

5.6 Meta-Strategy Experiments

The aim of these experiments is to empirically evaluate the influence of meta-strategies that individually use or combine a trade-off mechanism and a responsive mechanism on:

1. the dynamics of negotiation (section 5.6.2.1) and
2. the outcome (section 5.6.2.2) of negotiation

Recall from section 4.7 that a responsive mechanism implements a depth-first strategy in the negotiation state-space (figure 2.3), where the depth visited is a function of concession rate, which itself is a function of the resources left in negotiation, the time limits in negotiation and the behaviour of the other agents. Conversely, the trade-off mechanism can explore other parent nodes' siblings, as opposed to the siblings of a child node alone. A meta-strategy is then one that combines either search strategy towards an outcome (see figure 4.10). The aim of these experiments is to empirically capture the outcome and dynamic patterns of the wrapper when a combination of mechanisms are used for interactions. These patterns can then be used to form decision rules which agent designers can use to guide them in the selection of meta-strategies.

Two types of experiments are reported below. The aim of the first class of experiments is to analyze the *process* of different meta-strategy decision making (namely section 5.6.1.1). Therefore, the execution trace of the different meta-strategies are observed for a single run of an experiment. Consequently only a single outcome is observed. The aim of the second set of experiments, similar to the strategy experiments reported in section 5.4, is to analyze the effect of different meta-strategy decision making models on the final averaged joint utilities across a number of different environments 5.6.1.2. These observed final averaged utilities can then be used to deduce general statements about the meta-strategy experiments rather than their behaviour in a single run. Again this information is a useful guideline for agent designers because it can be used to assess the general behaviour of the given meta-strategy set.

5.6.1 Meta-Strategy Experimental Variables

The environment of these experiments is equivalent to the previous single-offer experiments. Briefly, the environment consists of bi-lateral negotiations between agents categorically labelled as $player$ and $opponent$, who negotiate over multiple quantitative issues $[price, quality, time, penalty]$. The interval values for these issues are perfectly overlapping (see equation 5.1). The $player$ assigns $[0.1, 0.5, 0.25, 0.15]$ and the $opponent$ assigns $[0.5, 0.1, 0.05, 0.35]$ as the importance of these issues.

In addition to the variations in the types of environments, new variables are needed that define meta-strategies. The first offer of both agents is generated using the responsive mechanism, since the trade-off mechanism requires at least one offer from the opponent. After that, an agent faces a choice of which mechanism to select. Since there can be an infinite number of meta strategies (as many as the potential sequences of choices between responsive and trade-off types of counter-proposals), the meta strategies considered in these experiments are limited to the set $\{responsive, smart, serial, random\}$. A responsive meta-strategy simply selects the responsive mechanism for generating an offer throughout negotiation. This is included to compare the trade-off mechanism against an agent that always concedes on utility. The parameters of the responsive mechanism are set to produce concession behaviours, since being responsive often involves concessions in the light of environmental needs (e.g. time, resources and behaviours). A smart strategy consists of deploying a trade-off mechanism until the agent observes a deadlock in the average closeness of offers between both agents, as measured by the similarity function. That is, the distance between the offers is not reducing. Under these circumstances, the value of the previously offered contract, $V^a(x)$, is reduced by a predetermined and arbitrary amount, here 0.05, thereby lowering the input value of θ into the trade-off mechanism. This value is chosen as a concession rate that is relatively lower than the concession rate of the responsive mechanisms. Thus, a concession in smart meta-strategy is a more "cautious" concession than its responsive counterpart. A serial strategy involves alternating between the trade-off and responsive mechanisms. Finally, the random meta-strategy randomly selects between the two mechanisms and functions as the control meta-strategy.

5.6.1.1 Process Oriented Experimental Independent Variables

The aim of the process oriented meta-strategy experiments is to investigate the dynamics, or a *single* execution trace, of different meta-strategies. Therefore, the sampling of independent variables is meaningless since the process is observed for only one execution trace. Thus the independent variables for responsive and trade-off mechanisms, as well as the associated time limits, are constant.

In these experiments the parameters of the responsive mechanism are set as follows. The tactics $[boulware, linear, conceder, titfortat]$ are set to $[0.5, 1, 5, 1]$ for both agents. These values reflect representative members of each tactic class. For example, the value of β for a $boulware$ tactic can range from values of 0 (being very tough) to 1.0 (being almost conceder). Therefore, the value of 0.5 represents an average tough tactic. There is only one member of each of $linear$ and *titfortat* tactics and the limits of the $conceder$ tactic are taken to be between 1.0 (least conceder) to 10 (the most).

The other element of the responsive mechanism, the strategy, is set as follows. Agent strategies are of type mixed2 (section 5.4). The initial value of the weighting of the tactics (Γ matrix corresponding to the initial strategy) is set to $[0.7, 0.1, 0.1, 0.1]$ for both agents. Therefore, both agents initially place more weighting on the boulware tactic. A mutual tough initial strategy is chosen because, as will be shown below, agents in subsequent iterations of negotiation modify this initial strategy by a policy that places less weight on the boulware tactic and more on the conceder tactic. Therefore, to prevent a fast approach to the interval values (large movement towards 0 along the x axis of the score for the $player$, for example, in figure 5.23), and hence quick agreements, the initial strategy is made to be tough, thereby allowing the trade-off mechanism to operate (at higher utility values—operating at θ values towards 1.0 along the x axis of the score for the $player$, for example, in figure 5.23).

The modification policy is simply slowly increasing the importance of the $conceder$ tactic as the thread of negotiation increases. Note, this policy is different to the one reported in the previous section (section 5.4) that conceded or remained firm according to the similarity between offers. The policy is that at each iteration the weighting of the $conceder$ tactic is increased by 30% and, correspondingly, the weights of the other tactics are uniformly lowered. Thus, both agents begin negotiation as tough strategist, but end up placing increasing importance on the $conceder$ tactics. Therefore, the modification policy is chosen independently of the others' offers and is dependent on the length of the thread. This policy is chosen because the overall required behaviour of the responsive mechanism is concessionary, because the combination of a concessionary mechanism and a trade-off mechanism, through a meta-strategy, can equally implement the similarity based strategy modification policy. The chosen policy will always concede because the thread of the negotiation always increases.

The parameters of the trade-off mechanism are set as follows. The exploration factor of the trade-off experiment, defined by the two independent variables number of children and number of steps to the iso-curve, are made a constant at 100 children and 40 steps respectively. The supplied similarity weights to the trade-off algorithm of each agent are set to be $[0.25, 0.25, 0.25, 0.25]$ (corresponding to uncertainty of the others' issue weightings). These values are chosen based on the previous observations in the trade-off experiments (section 5.5) that such a weight selection results in significant utility increases for the other agent (see the results shown in figure 5.29). Finally, the time limit of the both agents is set to 20 ticks of a discrete clock.

5.6.1.2 Outcome Oriented Experimental Variables

The aim of the previous experiments is to calibrate the dynamics of negotiation when agents interact with one another using either one or both of the developed responsive and trade-off mechanisms in a single type of environment. This knowledge is useful for developing an understanding of the processes involved in each of the mechanisms, but is less informative about the behaviour of a meta-strategy in different types of environments. These experiments aim to provide such an analysis by, in a similar fashion to the previous strategy experiments (section 5.4), shifting the focus of attention to the *outcome*, rather than the process, of negotiation in *types* of environments. However, once again, in order to control the number of free independent variables that can be sampled, and allow some comparison with the process-oriented experiments above, the variables of the $opponent$ are chosen to have the same values as the process-oriented experiments (section 5.6.1.1) and the variables of the $player$ are sampled.

More specifically, the parameters of the responsive mechanism are as follows. The same update rule is used as for the process-centered experiments. However, the parameters of the tactics are now sampled for the $player$. The β parameter of the $boulware$ tactic is sampled within the interval $[0.01, 0.2]$ (more boulware than previous process-oriented experiments). The $linear$ and *titfortat* tactics can not be sampled (since these tactics can only take on a value of 1.0). A $conceder$ tactic is sampled within the interval $[20, 40]$ (more conceder than previous process-oriented experiments). Therefore, whereas the previous process-oriented experiments evaluate the average representatives of a tactic class, in these experiments more extreme tactic members are evaluated for completeness by choosing a $player$ that is more $boulware$ or $conceder$.

In turn, the parameters of the trade-off mechanism are as follows. The exploration factor, defined by the two independent variables number of children and number of steps to the iso-curve, is once again made a constant at 100 children and 40 steps respectively, for the $opponent$. However, the number of children generated at each step in the trade-off algorithm for the $player$ is now sampled between the ranges of $[100, 200]$ and the exact number of steps chosen is within the range $[40, 80]$. These values are chosen so that, on average, the $player$ is made to perform more of an elaborated search of the space of the possible outcomes. Finally, the time limit of the $opponent$ is set to 20 and sampled within the ranges of $[30, 60]$ for the $player$. Higher time limits and a greater exploration rate are chosen for the $player$ to allow the trade-off mechanism to search for better deals.

The number of environmental samplings is set to 400. This ensures that the probability of the sampled mean deviating by more than 0.01 from the true mean is less than 0.05. The experiments were written in Sicstus3.7.1 Prolog and ran on *HP* Unix parallel machines at the Center de Supercomputacio de Catalunya *CESCA* utilizing four CPUs, 9MB of memory and lasted 1954 seconds.

5.6.2 Hypotheses and Results

Finally, the expectations and observed results of the process and outcomes of meta-strategy experiments are presented in the following two subsections.

5.6.2.1 Meta-Strategy Process Hypotheses and Results

> **Hypothesis 16:** *The more the space of possible deals is explored* jointly, *the better the joint outcome. However, higher joint utilities are gained at the expense of greater communication between the agents.*

The hypothesis essentially states the expectation that a pair of smart meta-strategies should select final outcomes that have a higher joint value than other types of meta-strategies. This is expected because a smart meta-strategy is essentially a trade-off strategy that *only* concedes a small amount (0.05 in this case) when a deadlock is detected. All other experimental meta-strategies have an element of concession involved in them (since the variables of the responsive mechanism have been chosen to behave in a concessionary fashion). Thus any meta-strategy that selects a responsive mechanism in the course of negotiation (all pairs of meta-strategies except [smart,smart]) should result in joint utility execution traces that "move" south westerly, away from the pareto-optimal line. Furthermore, meta-strategies that engage more in search for higher joint utilities and less on concessions should result in higher communication loads. This latter expectation is based on the intuition that a responsive mechanism generates contracts that successively approach the point of cross over in offers faster than the trade-off mechanism. Hence it is to be expected that a meta-strategy that selects the responsive mechanism should reach acceptable deals quicker than one that is smart.

Figure 5.31 presents the data for the meta-strategy experiments investigating the process of mechanism selection. Individual offers between the $player$ and the $opponent$ are depicted as circles and squares respectively. The sequences of offers are joined by a solid line for the $player$ and a dotted line for the $opponent$. The final agreement is depicted as the offer where the circle and square meet. The communication load is simply the addition of the numbers of circles and the squares.

The observed rank ordering, in figure 5.31, across meta-strategy pairings over the summed joint utility gained for the final outcome directly supports hypothesis 16. The highest joint gain is achieved in negotiations between two $smart$ meta-strategies. In this case, the final outcome is closer to the pareto-optimal line than any other meta-strategy pairing, implying that such a pairing of meta-strategies results in outcomes that are most beneficial to both parties. The remaining rankings for $player, opponent$ pairings of meta-strategies are then [smart,serial], [serial,serial], [smart,random], [smart,responsive], [serial,responsive], [random,responsive], [random,random] with respective joint gains of $1.27, 1.18, 1.146, 1.11, 1.076, 1.06, 0.99$. In general, the higher joint utilities occur when at least one of the agents is $smart$. The $random$ meta strategists, as expected, perform worst.

Hypothesis 16 is further supported by the observation of the number of messages exchanged between agents using different meta-strategies (recall that in these experiments the number of messages exchanged between agents is simply the addition of the individual messages exchanged in figure 5.31). This indirectly measures the communication load a meta-strategy places on the agents. As predicted by hypothesis 16, the observed pattern is almost the reverse for the joint value outcomes above; with a [smart,smart] pairing incurring the highest communication cost (reaching a deal at 19 rounds (recall that the time limits allowed are 20 ticks of a discrete clock, followed by [random,random], [smart,responsive], [smart,random], [smart,serial] (14 rounds), [serial,serial] (13 rounds), and [serial,responsive] (12 rounds). This observation supports the intuition

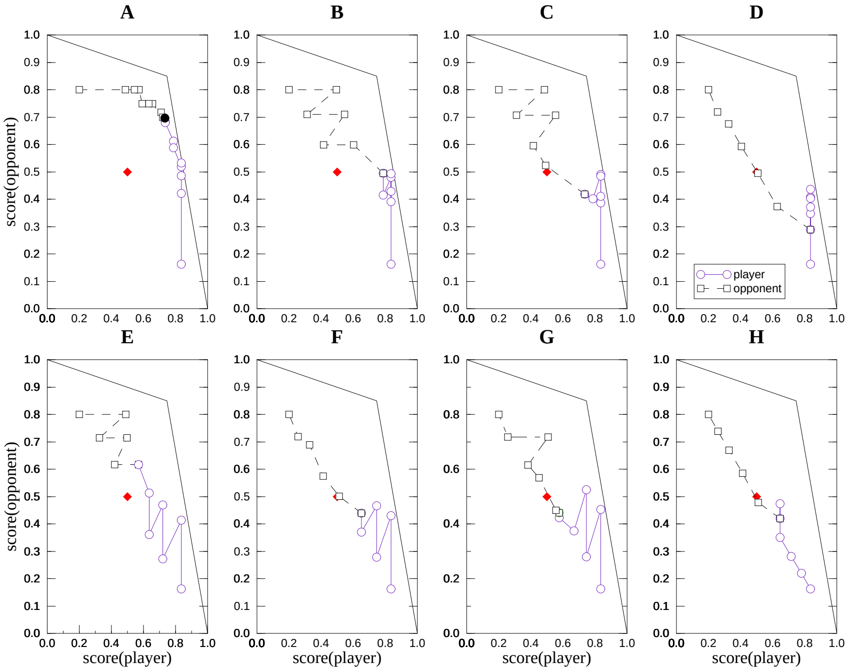

Figure 5.31: Dynamics of Negotiation Process for Meta Strategies, Pairs Denoted as Meta-Strategy of the $player$, Meta-Strategy of the $opponent$: A) smart v. smart, B) smart v.serial, C) smart v. random D) smart v. responsive, E) serial v. serial, F) serial v. responsive, G) random v. random, H) random v. responsive.

that higher joint utilities are gained through greater search, which, in turn, involves more communication between the agents.

5.6.2.2 Meta-Strategy Outcomes Hypotheses and Results

The hypothesis for these experiments is the same as the process-oriented experiments, namely:

> **Hypothesis 17:** *On average, the more the space of possible deals is explored* jointly*, the better the joint outcome. However, on average higher joint utilities are gained through greater communication between the agents.*

That is, the aim of these experiments is to show that in the long run, or on average and independently of the type of environment, better exploration of the space of possible deals should result in higher joint outcomes. The expectation of the outcome-oriented experiments is no different than the experiments that did not involve sampling the types of environments. In the average case, those meta-strategies that involve more search will result in better outcomes, but at the cost of increased communication.

Figure 5.32 supports the expectation over the joint utility part of the hypothesis. The key to the meta-strategy pairing is amended with the total summed average of the joint utility the pairing achieved. As

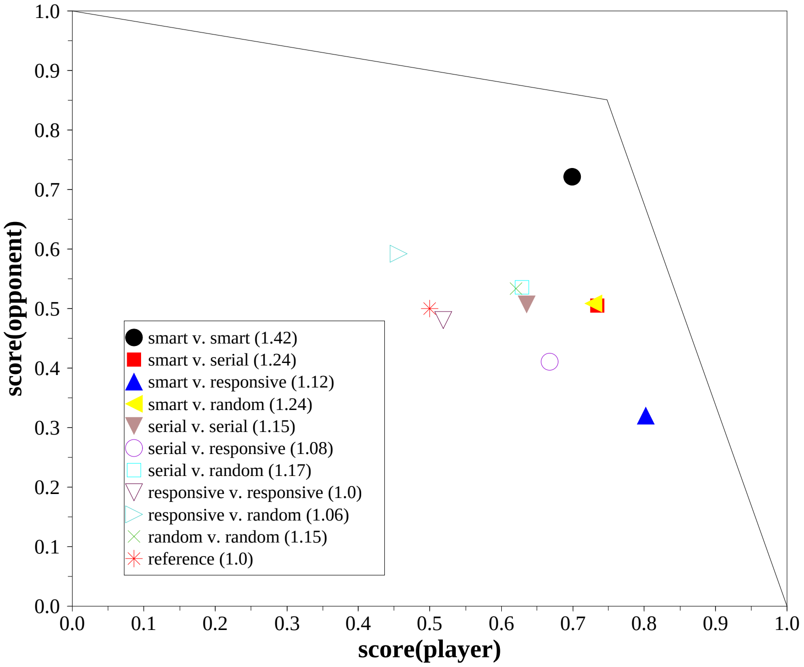

Figure 5.32: Final Average Utility Outcomes for Meta Strategies Pairings.

expected, pairings of a meta-strategy that compute counter-offers using the responsive mechanism lead to the worst joint outcomes (joint utility of 1.0, the outcome lying on the constant-sum line—see supporting data in strategic experiments, section 5.4). Only moderate joint gains above 1.0 are achieved when the meta-strategy of one of the agents is not purely a responsive one ([smart,responsive]: joint utility outcomes of 1.12, [serial,responsive]: joint utility outcomes of 1.08, [responsive,random]: joint utility outcomes of 1.06). At the other extreme, joint utility of outcomes is best maximized (outcomes lying closer to the pareto-optimal line), as expected, when agents use a smart meta-strategy. More specifically, the best outcome is achieved for a [smart,smart] meta-strategy with joint utilities of 1.42. In between these two extremes lie the outcomes that are, in the main, due to the interactions with one agent whose meta-strategy is serially switching between a trade-off and a responsive mechanism (the interval of joint utility outcomes of 1.15 to 1.24).

Once again, the meta-strategies that result in higher joint outcomes, as predicted by hypothesis 17, are achieved at the expense of higher communication costs (figure 5.33). The meta-strategy pairing [smart,smart] results in an average number of communication rounds of 19.48 (note, the proximity of this to the time limit of the *opponent*, whose deadline is fixed at 20 ticks of a clock). Conversely, interactions between two responsive meta-strategies resulted in poorer joint outcomes (figure 5.32), but at a relatively lower communication cost (10.16).

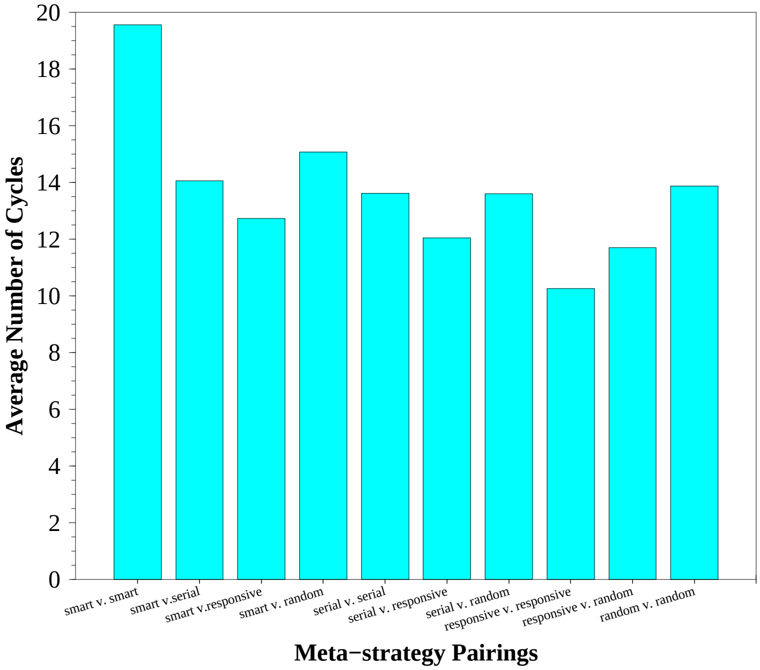

Figure 5.33: Final Joint Average Number of Cycles for Meta Strategies Pairings.

In summary, the results from the single execution trace of the trade-off algorithm (section 5.5) and the meta-strategy experiments (section 5.6) indicate that a better exploration of the space of the possible set of outcomes leads to agreements that are higher in joint gains. Furthermore, this increased search results in: i) higher joint outcomes on each iteration of the algorithm (section 5.5), across a single run in a unique environment (section 5.6.2.1) or across multiple environments (section 5.6.2.2) and ii) higher communication costs.

5.7 Summary

In this chapter, three components of the developed negotiation wrapper (the responsive, trade-off and meta-strategy mechanisms) were empirically evaluated by conducting a series of exploratory experiments. These experiments were conducted to: i) test the intuitions about the underlying causal relationships between both the model's key variables and the agent's environment and ii) provide some guidelines for how the wrapper can be "tuned" by a designer of a negotiating agent. However, manipulation experiments are needed that test more concrete causal hypothesis and result in better data models. Nonetheless, the exploratory experiments reported in this chapter help "tune" some of the parameters of the mechanisms through exploration of a subset of the space of possible variable ranges, through different combination of agent architectures and environments. The experimental agent architectures, or the choice of which decision mechanism to use for decision making, was restricted to responsive (tactics/pure, static strategic/mixed1 and dynamic strategic/mixed2), trade-off and a number of different meta-strategies that selected different combination of mechanisms. The experimental environments were in turn motivated by some of the features and requirements of the two target domains identified in section 1.4.3 such such time limits, certainty levels and number of issues.

Hypothesis one to eight summarize the expectations of outcomes and the processes of negotiation when simple tactic (pure) agent architectures interact in short or long term deadlines. In these experiments the expectation that tactics which reach reservation values more slowly will perform better was rebutted. In fact, it was found that the success of a tactic is a function of not only the composition of the population, but also appropriateness of the tactics to respond to changes in the environment. Indeed, tactics that approached the reservation values of an issue more slowly (more Boulware) did make high value deals but this benefit was reduced due to the lower number of successful deals made as well as the communication costs involved, specially when the population includes imitative tactics that “magnify” the toughness profile of the population. In fact, the best tactics were the ones that linearly approached their reservation values. Also confirmed was the expectation that such simple agents, which consider only a single environmental criteria, will result in more varied distribution of outcomes around the most equitable outcome point (hypothesis seven). In fact, simple agent architectures perform best (maximize their joint utilities) only in encounters between two pure strategies that give higher weighting to tactics that approach the reservation of an issue in a linear fashion.

Hypothesis eight to fourteen, on the other hand, summarize the expectations of outcomes and the processes of negotiation when more complex (static strategies/mixed1 and dynamic strategies/mixed2) agent architectures interacted in short or long term deadlines. The expectation in this set of experiments was that if the mixing between different tactics of both agents is more “smooth” (or the more equal the contribution of each individual tactic to the computation of a new overall concession rate), *and* if the method of computation of the new concession rate is performed intelligently according to some objective function (such as the similarity between the exchanged contracts), then the more equitable the final outcome for both parties. Indeed, variations by either party from these parameter settings results in distribution of outcomes that although maybe locally more equitable are less *jointly* equitable.

Hypothesis fifteen captured the expectations of outcomes and the processes of negotiation when an agent implemented a trade-off algorithm in long term deadlines (a more complex agent architecture than the responsive mechanism). The aim of this experiment was to evaluate whether a relationship exists between the complexity of the search of the space of possible deals and the quality of the outcome (from the perspective of the opponent) and if so whether this relationship is affected by the uncertainties involved in trade-off negotiation. Indeed, results confirmed the expectations of such a relationship where a more refined search of the possible space of contracts did resulted in selecting and offering a contract that had more value to the other agent. Furthermore, this search was directly affected by the information the algorithm had about the other's issue importance rankings where better information (less uncertainties) resulted in better contracts to be selected.

Finally, the expectation that either on a single case (hypothesis sixteen) or the average case (hypothesis seventeen) the most equitable outcomes should be reached when both agents intelligently search the space of possible contracts using both the responsive and the trade-off mechanism (the most complex agent architecture) according to some objective function. This objective function (the similarity function) implemented the meta-strategy and directed the negotiation search by selecting the trade-off mechanism when the objective function was being maximized and the responsive mechanism when the local minima of the objective function was reached. These expectations were confirmed by the observations where it was found that a pair of smart meta-strategies reached deals closer to the pareto-optimal line than combination of any other non-intelligent combination of meta-strategies.

The implications of these results for the designer of the negotiating agent is deferred to section 6.2.1.

Chapter 6

Conclusions and Future Work

The conclusions and the directions for future work, derived in the main from the identified weaknesses, are jointly presented in this final chapter. However, the work reported here is reviewed first.

6.1 Review of the Thesis

This thesis has presented a solution for the problem of coordination among two autonomous agents that need to interact with one another. The solution addresses two sets of requirements identified in the first chapter: i) the requirements of the actual problem that the coordination system should achieve (section 1.4.3) and ii) the requirements that arise in designing of a coordination system (section 1.1).

The first requirement has been how to coordinate domain problem solvers that need the services of one another in their local problem solving. This interaction problem was defined for each individual agent as the tuple $P = \langle I, C, Criteria \rangle$ (equation 2.1). I is the set of issues that describe features of a service. C describes the constraints of each of these features (such as its importance level, its reservation values and an agent's preferences over the values it can take, as well as other environmental constraints such as the time and resources available for negotiation). $Criteria$ is then defined in terms of the principle of individual rationality. The rationality principle adopted in this thesis was the maximization of some utility function. The agent interaction problem was then defined as the mutual and strategic selection of values for I that respect C and satisfy $Criteria$ for each party given the normative protocol of interaction. Furthermore, this solution has to be mutually derived without knowledge of others' sets of constraints and also with limited computational resources. For this reason, the satisfaction, rather than the optimization, of $Criteria$ is considered to be sufficient. Conflicts were then defined as when the local criteria of each agent negatively interact. The proposed solution to this constrained search has been to design a coordination framework that consists of: i) a protocol that assists the agents in the communication (or on-line) phase of their interaction problem solving and ii) a set of mechanisms that assists the agents in their deliberation (or off-line) phase of their interaction problem solving. Agents then use these two components of the coordination framework to solve their interaction problem by representing and iteratively reasoning and exchanging offers over services as issue-value pairs. The novelty of the research reported here is in the deliberation mechanisms for multi-dimensional conflicts. Multi-dimensional interactions require reasoning over a larger set of agent constraints compared to single dimension. These novel aspects were driven by the requirements outlined in section 1.4.3 where it was shown that the target problem domains of this research, and the real world in general, are multi-dimensional in nature. Likewise, each of the dimensions have constraints attached to them and agents need to reason about these constraints explicitly. For instance, some dimensions of a problem are more important than others and a search for a solution is often based on such relationships. For example, the log-rolling strategy (Pruitt 1981) searches for new solutions by violating the constraints of the least important issues and further constraining the constraints of more important issues. The multi-dimensional nature of the interaction also indirectly leads to the requirement that agents are able to combine their preferences over each of the individual dimensions. Thus, agents require a model that supports the consolidation of preferences over each issue into a single preference.

The main contribution of this thesis is the developed deliberation component. Three mechanisms were presented that, given the problem specification (the issues, their constraints and criteria), search in a distributed and autonomous fashion (important domain requirements, section 1.4.3) for individually acceptable assignment of values to each dimension of negotiation. When individual assignments are in conflict, detected by a set of evaluation functions, then agents use one or more of the decision options to resolve them. The first mechanism presented was the responsive mechanism which implements various degrees of concession (from no concession to full concession) according to the agent's current environment. This mechanism was designed to model concessionary behaviours according to how much negotiation time and resources were available (both requirements mentioned in section 1.4.3). Additionally, the mechanism models decisions based on the behavioural profile of the other agent, another important feature of the target domains. The concession mechanism is computationally simple (involving the execution of simple functions, called tactics, and the assignment and modification of importance weights to each tactic, called a strategy). Furthermore, it requires a minimal amount of information about the choices of the other(s); decisions are conditioned on the environment of the agent and minimally (through the behaviour-dependent tactics) on the choices of the other(s). Indeed, the only assumption made about the other(s) is that conflicts arise because the other agent has an opposing preference ordering over increasing domain values for all issues. This information is inferred by the roles agents play in interaction (e.g. a seller prefers higher prices to lower ones and for a buyer the reverse is true). Thus the mechanism is based on the realistic assumptions that: i) the agent is not omniscient and/or ii) super logical. Rather, an agent's knowledge about the choices of the other(s) is highly limited and its reasoning capabilities are bounded. These features of the mechanism were factored into the design process for the flexibility requirement of the wrapper (see below).

The other novel decision components of the coordination framework are the trade-off and issue-manipulation mechanisms (since they are computationally more complex than the responsive mechanism). The trade-off mechanism was developed to model cooperative reasoning over conflicts, defined as interactions where at least one of the agents is motivated by the intention to increase the social welfare function (globally rational outcomes that aim to make both agents better off), but achieves the current aspiration level over its preferences (i.e. is locally rational, satisfying the local criteria specified over each issue). This contrasts with the responsive mechanism that models more selfish reasoning, defined as interactions where agents are not interested in increasing the social welfare function, but rather only in satisfying their own preferences. The responsive and trade-off mechanisms jointly address the requirement, identified in section 1.4.3, for different types of motivations over conflict. The issue-manipulation mechanism, in turn, was developed to not only assist agents in escaping local minima in the search of the social welfare function, but also because the nature of the problem naturally involves modification of the set of negotiation issues at run time due to dynamically changing domain requirements (section 1.4.2).

Both the trade-off and issue-manipulation mechanisms are a novel way of agents individually searching the space of possible deals. However, in comparison to the responsive mechanism, such searches require more information to be supplied about the other agent and involve more deliberation about the other agent's preferences. A fuzzy similarity technology has been developed to handle these requirements. Although a formal model of the issue-manipulation mechanism was developed, its implementation by an algorithm and the analysis of the algorithm's resulting computational complexity is deferred to future work. However, a novel trade-off algorithm was developed that implements a fuzzy similarity based trade-off negotiation and its complexity was shown to be linearly proportional to the number of issues. This computational tractability is a desirable property that fits with the key assumption of this work that the agents are computationally bounded. The use of fuzzy similarity also satisfies the flexibility objective with regards to the informational requirement of the agent, because the technique is used to model the uncertainty of an agent's beliefs over the preferences of the other agents' as fuzzy relationships between values of the domain, and not the other agents' actual preferences. This means that the agents do not have to make interpersonal comparisons of preferences when making trade-offs, a task that requires full knowledge of the other agent's preferences.

When taken together, each of the mechanisms addresses a subset of the requirements identified in section 1.4.3. For example, the responsive mechanism can implement a selfish attitude in interactions, but is inappropriate for searching the solution space of possible outcomes in a more cooperative manner. However, whereas the trade-off mechanism is capable of performing such a search, it is computationally more costly than the responsive mechanisms. Given this, what is required is meta-reasoning about the various trade offs involved in the use of each mechanism for the generation of offers. This meta reasoning can then be used by an agent to address the changing requirements of the agents accordingly. Thus, the meta-strategy may select the trade-off mechanism for generating service contracts to agents that belong to the same organization, but select the responsive mechanism with a low concession rate for service negotiations with agents that are from different organizations. Thus reasoning over different features of interactions (cooperative versus selfish interactions, computationally simple v.s more complex search, long v.s short term negotiation deadlines and low or high domain resource levels, which collectively form the set of requirements enumerated in section 1.4.3) can be modeled through a temporally changing combination of mechanisms as meta-strategies. A meta level deliberation mechanism was informally presented that implements such offer generation strategies over the available mechanism choices.

The developed wrapper incorporating the responsive, trade-off and meta-strategy mechanisms was then empirically evaluated in a number of different environments. Evaluation was needed to: i) develop and test exploratory hypotheses about the causal relationship between the large number of mechanisms variables and the agent's environment, ii) assist the designer of a negotiating agent in "tuning" of the framework for given environments and iii) to validate the efficacy of the heuristic aspects of the model (for example, a meta-strategy that always involves the trade-off mechanism until a local minimum in the social welfare function is detected is a decision heuristic whose efficacy across different types of environments can not be determined a priori). For these reasons, the wrapper was empirically evaluated across a number of environments. In experiments involving interactions among two agents both using the responsive mechanism the largest variability in the results were observed if pure strategies are chosen to generate offers. The best results were obtained for strategic agents that modeled the generation of offers as a combination of tactics and modified this combination consideration in the course of negotiation. The intuitions about the trade-off mechanism, or a meta-strategy that frequently selects the trade-off mechanism, were also confirmed. The trade-off mechanism experiments found that the implementation of such strategies does indeed increase the social welfare function in more than one type of environment, but at an increase in communication costs, signifying that the search takes longer to converge on a mutually acceptable deal. Deals are made more quickly if the responsive mechanism is used, but the social welfare function is poor considering that higher joint utilities can be gained through the multi-dimensional nature of the problem.

In addition to satisfying the requirements of the target domains, the developed negotiation wrapper also addresses many of the desiderata that were identified in the design of a coordination system (section 1.1). The design requirements were introduced as the configurability requirement or the reusability and flexibility of the developed coordination framework for use across both open and closed distributed systems. The flexibility of the developed coordination framework has already been discussed above. Reusability has also been factored into the framework design by:

- making as few commitments to the domain problem solvers' architecture as possible. Interaction problem solving is separated from local domain problem solving by functionally separating the negotiation wrapper from the local domain expert. Thus the wrapper can be seen as providing social knowledge to the local asocial domain problem solver. The interface between these two modules supports low level information about the requirements of the domain problem solver (the service(s) it requires, the core and auxiliary features of the service(s), its constraints and satisfaction criteria over each of these features). The wrapper does not have control over or access to any of the operations of the domain problem solver.

- designing both cooperative and selfish decision making mechanisms into the agent's decision making architecture. In DPS systems agents are assumed to be cooperatively motivated in interaction. Conversely, in MAS agents are assumed to be selfishly motivated in interactions. Therefore, in both approaches a single agent attitude is hardwired into the decision making architecture. However, the interaction attitude of an agent ought to be a function of its environment. For example, as was seen in the target domains of this thesis, the same agent can enter two different types of interactions where one is cooperatively motivated and the other is more selfish. Therefore the agent (more correctly, the agent designer) needs to be supplied with both types of decision making facilities.

- emphasizing the notion of services. Services are, like objects in the object-oriented paradigm, a representation of the capabilities of the local domain problem solver in providing problem solving expertise. Thus, like objects, services are reusable across different problem solving episodes.

This configurability of the coordination framework has been guided by the requirement to design a library of different negotiation decision making strategies which the agent designer can then implement in their agents. The designer is free to configure his/her agent for interaction according to their prevailing objectives (such as strategies for increasing the social welfare function or for achievement of local objectives). This descriptive design approach contrasts with the prescriptive models of game theory where the most rational strategy of a game is analyzed and prescribed to the agent. In the latter case, however, it was shown that such models often make unrealistic assumptions. Therefore, the approach taken in this thesis has been to describe and empirically analyze the possible set of behaviours that can arise when more realistic assumptions are adopted. The designer of an agent is then free to choose a strategy that best suits his/her problem. This configurability claim has been procedurally demonstrated in the successful application of the coordination framework to seven different application domains, ranging from business process management to electronic commerce.

6.2 Discussion

Coordination has been identified as one of the most central problems in DAI (section 1). For this reason the research, including the work reported here, has produced a large number of proposals for coordination protocols. The coordination problem was informally introduced as a process that *consists of composing (relating, harmonizing, adjusting, integrating) some coordination objects (tasks, goals, decisions, plans) with respect to some coordination process, which solves the coordination problem by composing co-ordination objects in line with the coordination direction* (Ossowski 1999). This general view of coordination was given a more concrete interpretation through development of the negotiation wrapper. The composition process is achieved locally by each agent through implementing one, or a combination, of the proposed mechanisms. Agents then use these mechanisms, together with a communication protocol, to compose and exchange multi-issue contracts (the coordination objects) that increases either the local or global utility (the coordination direction).

6.2.1 Guidelines for the Negotiating Agent Designer

The empirical evaluation of the mechanisms also resulted in a number of findings that can be used to formulate general guidelines for agent designers wishing to use the negotiation wrapper. The aim of the experiments was the exploration of a subset of the space of possible variable ranges, through different combination of agent architectures and environments. Recall that an agent architecture is a particular instantiation of the agent that follows from the model described in chapter four. Thus given the negotiation model an agent designer can design a very simple negotiating agent where the meta-strategy selects only one mechanism. For example, the the designer may choose only the responsive mechanism for the design of his/her agent. Further simplification can be made when the designer chooses a responsive mechanism that is composed of a single tactic. These choices result in an agent that requires no meta-strategic (since the responsive mechanism is always selected) or strategic decisions (no $f()$, or pure strategy, since there exists only a single tactic). Such a simple agent is best represented by a Kasbah agent. As this example shows, an agent designers is then free to compose increasingly more complex agents by choosing different meta-strategies, tactic sets or strategic update functions. Additional complexities arise when negotiation environments are also taken in considerations.

Therefore the aim of the experiments reported in this chapter was to evaluate which architecture-environment(s) leads to (un)successful outcomes. If two agent designers are motivated by some global system evaluation criteria, such as the sum of the joint utilities (maximized by the pareto-optimal line) or the reference point, then the following guidelines can be derived from the observations in these experiments.

- An agent designer who implements a simple agent architecture (responsive and pure strategy) should expect interactions that prolong the possibilities of joint gains. This is because simple agents may fail to respond appropriately to changes in their environment. This conclusion was indirectly confirmed by the unexpected success of linear tactics.

- A more complex agent architecture (responsive and strategic) was then evaluated in a number of different environments. It was found that the outcomes, both in terms of utilities and costs, for a strategic and responsive agent is a function of:

- the composition of the responsive agent architecture—the number and types of tactics
- the initial parameters of each *individual* and *joint* architecture
- and the joint local modification of these parameters by both agents

The first guideline states the agent designer should be aware that the type and number of tactics of a responsive agent affects the outcome and process of negotiation. Thus a tactic set should be selected that adequately represents a range of desired behaviours. For example, a tactic set of mainly Boulwares will result in tough negotiator independently of strategic decisions. Conversely, an agent's behaviour will be concessionary if the domain of operations of the strategic reasoner is a tactic set with $\beta > 1$ (corresponding to tactics that quickly reach their reservation values). Therefore, to be responsive in different environments an agent requires appropriate set of tactics.

An agent designer using the developed model must also set the initial values of the strategic reasoner. The initial value of weights of the tactic set corresponds to a slightly more complex agent who reasons about a number of environmental factors by computing a new concession rate. It was shown that better social outcomes follow when both agents engage in computing a new concession rate based on a number of environmental factors. In fact better social outcomes should be expected if agent designers can jointly agree on the same set of tactics and strategy for their initial settings.

However, most equitable outcomes should be expected with even more complex agents as shown when a responsive agent interacted with another responsive agent and both compute a new concession rate, given a set of environmental factors, *according to some objective function*. An agent designer who selects a strategy similar to a fixed (mixed1) strategy for his/her agent should expect an undirected search for a solution. However, best social outcomes follow when agents engage in directed search according to some objective function (in this case the closeness between successive offers). That is intelligent adjustment (or search), rather than constant adherence to the same environmental considerations, should result in better social outcomes.

- For more complex agent architectures that involve trade-off negotiation, the task of the agent designer is transformed from specifying "tunings" that affect local problem solving to "tunings" that affect the problem solving of the other agent. That is, the problem of the agent designer using the trade-off mechanism is to represent information about the other agent (as beliefs in the AM). It was empirically shown that this uncertainty is best addressed if the designer does not attempt to guess the information of the other agent (unless completely sure), but rather assigns an uncertainty to the agent's belief about the other agent (note the similarity with the argument of strategic interactions presented in chapter two). Indeed, although not shown, better results should be expected if this uncertain belief is sequentially updated in the course of negotiation (learning implemented as Bayesian updating).

- If time and computation are resourceful or there is a need for increasing the social welfare of the outcomes, then a more complex agent architectures that involve strategically selecting between the responsive and trade-off mechanisms should be expected to perform better. In particular, best social outcomes should be expected if the search for a solution is intelligently directed by an objective function that selects the trade-off mechanism when the objective function is being maximized and the responsive mechanism when the local minima of the objective function are reached. That is, the more intelligent the meta-reasoning about which mechanism to select, the more the social welfare function is maximized.

6.2.2 Limitations of the Current Work

The contribution of this thesis has been a proposal for *a* computational model of decision making for negotiating agent that has been empirically evaluated. However, this proposal only models a subset of the issues identified in chapter two. Much more work is required to develop richer interaction protocols that adequately model a more elaborate concept of coordination that is applicable to a wider set of problems. In particular, the following limitations need to be addressed:

- development of an issue-manipulation algorithm
- the current negotiation model does not handle qualitative issues
- better models of other agents are needed
- the current bi-lateral protocol is inadequate in capturing inter-dependencies among complex activities

6.3 Future Work

The proposals for future work are derived from the limitations of the work presented above and is based on addressing some of the additional issues identified in chapter two. In particular, the future work is categorized into extensions to the:

- decision making level
- interaction protocol level
- evaluation level and
- application level

6.3.1 Extensions of Decision Making

The decision making functionality of the negotiation wrapper adequately models individual agents' decision choices over actions and strategies given the information, time and computational constraints involved. However, three future directions of research still need to be addressed: i) developing an issue-set manipulation algorithm, ii) modeling of qualitative issues and iii) a methodology for modeling other agents.

Although a formal model of how the set of negotiation issues can be manipulated, no algorithms have been developed. This is clearly an important direction of future research. Furthermore, the presented model has concentrated on resolution of quantitative issues where movements along the utility function of an issue is continuous. However, all mechanisms need to be extended to deal with the introduction of qualitative issues that have an associated non-continuous utility function. Some work has already been carried out to extend the responsive mechanism to handle non-continuous domain for qualitative issues (Matos, Sierra, & Jennings 1998). However, the trade-off or issue-manipulation mechanisms still need to be extended.

There are two choices of approach that address the current weaknesses in modeling of other agent. On the one hand, mechanisms can be developed within the negotiation wrapper itself that assist the agent in modeling the other(s), given the current single shot, sequential alternating protocol of interaction. Alternatively, the current decision mechanisms could be supplied with an alternative interaction protocol that allows the agents to learn and develop a model of one another. Which of these approaches to handling uncertainty of the others' is best is seen as an empirical question that needs to be tested for given environments and types of problems.

If the first approach is adopted, then one proposal for modeling others is to develop other types of utility functions that model an agent's attitude towards risks (risk taker, neutral or aversive (Binmore 1992)). Although not directly modeling other agents' decisions, a utility function that takes into account an agent's attitude towards uncertain events, given a sure event, does indirectly model other(s) by modeling the expected utility an agent will gain given the uncertainty of others' choices. Although this approach has weaknesses, identified in chapter two, it is a reasonable choice of an extension because: i) the modifications to the proposed model to handle this addition are minimal, requiring the design of utility functions that model agent's preferences in risky environments and ii) to be a Bayesian agent, or to compute the expected utility of a deal, requires supplying agents with an a priori probability distribution of the likely outcomes. Recall that the source of these a priori distributions has been a criticism leveled against the Bayesian approach. However, similarity measures, modeling the problem domain and not an agent, can be used as the a priori distribution in such cases.

6.3.2 Extensions of the Protocol

In some situations, however, the initial a priori distribution may simply be wrong. The solution to this problem is closely related to the second approach proposed above to better model the other agent—that is the current decision mechanisms can be supplied with an alternative interaction protocol that supports learning. Then if interactions are repeated, a Bayesian agent can update the similarity induced a priori distributions given the evidence it gains from interaction.

The currently proposed set of mechanisms can also be appended by other mechanisms to better handle the uncertainty of others' actions, even if the first choice is not adopted; i.e. the decision mechanisms of the wrapper are kept without any alterations. In particular, what is needed is to append to the current set of decision mechanisms learning algorithm(s) that assist the agent in better "tuning" each of the wrapper's decision mechanism parameters. For instance, learning algorithms can be used in the responsive mechanism to modify not only parameters of the individual tactics (e.g. β or δ of the time-dependent and behavior-dependent tactics respectively), but also the agent's strategy ($f()$ that modifies the Γ matrix, section 4.4.3). Likewise, learning algorithms can be useful in approximating values for the weights the other agent(s) place on each of the issues ($\{W\}$). Such knowledge is useful for the operation of all of the mechanisms.[1] For instance, as was empirically shown in the trade-off experiments (section 5.5.3) better approximations of others' weights results in an increase in the social welfare function. A better knowledge of other agents' attached importance to each issue is highly relevant information in making trade-offs. This information can also be usefully utilized in the issue-set manipulation mechanism for making decisions about which issues to add or remove. Finally, learning algorithms can be applied at the meta-strategy level to condition the selection of the most appropriate mechanism to the history of previous interactions. For example, the trade-off mechanism may have resulted in higher success frequencies than other mechanisms in the course of previous interactions between two given agents. More sophisticated learning can involve an agent learning which mechanism to select from the relationship between the features of the current interaction with those of previous interactions with other agent(s). The extension of the current model with such Case-Based reasoning learning algorithms (Kolodner 1993) is natural because the developed similarity technology can be used to model such relationships between the present and the past cases.

However, as noted in section 2.1.4, the replacement of a single-shot with a repeated interaction protocol has a number of significant consequences on the agent architecture. Although agents can benefit from learning in a repeated interaction protocol, additional mechanisms must also be designed to support reasoning in such environments. Repeated interactions have been extensively studied in game theory (Axelrod 1984) due to their role in resolving multiple equilibria problems through the development of systems of conventions. Thus, if a game has multiple equilibria and if agents interact repeatedly, then they can decide on a single equilibria as a convention (driving on the left or right is an example of such a convention). An example of how agents' reasoning changes in a repeated game was briefly introduced in section 2.2.5.

[1] This knowledge, again although possibly incorrect, can nonetheless be revised and updated in subsequent interactions given the outcome of the past interactions.

There, it was shown that the dynamics of negotiation altered in repeated games. In particular, the stability of Mrs Shee's strategy of action (up) depended on her observation of Mr. Hee's strategy choice. Thus, an agent's current choice is dependent on the future choices of others. Agents must therefore reason about this type of action contingency given the reputation of other(s) and how much the agent can trust them from their commitment history.

Another extension to the protocol is also necessary not only when the frequency of interactions is considered, but also when the size of the agent society is considered an important factor to model (section 2.1.1). The size of the society becomes important when the types of problems considered are not just restricted to the resolution of conflicting preferences between only two parties, but, rather, extends to a number of agents performing distributed problem solving in a group. As it stands, the proposed coordination framework is inappropriate for the latter types of problems. In order to solve this type of problem, the coordination framework needs to be modified so that multiple agents can exchange not offers over services, but plans, goals or other meta-attitudes such as intentions (Dennett 1987). The evaluatory components of the decision mechanisms can then be used to evaluate plans, goals or intentions from a local perspective. However, plan, goal or intention generation mechanisms would need to be designed to generate offers over plan, goal or intention alternatives. New mechanisms are therefore needed because the input into the current set of mechanisms needs to be changed from an issue (together with its associated reservation values, weights and preferences) to a higher level structures such as plans, goals or intentions which are composed differently and exhibit different properties to issues. Therefore, other reasoning mechanisms are required that generate offers over higher level representations.

Multi-lateral negotiations also open up the possibility of extending the wrapper to model coalitions where a collection of agents form a group to perform or achieve a common objective (Kahan & Rapoport 1984, Sandholm & Lesser 1997, Shehory & Kraus 1995). For example, buyers in a market economy often form large coalitions to reduce sellers' prices. The problem then is how to modify or adapt the current wrapper so that agents can reason about coalitions. One such solution may be to allow agents to form a group using some coalition forming algorithm (coalition formation has been extensively studied in game theory (Kahan & Rapoport 1984, Binmore 1985, Sandholm & Lesser 1997, Shehory & Kraus 1995)). Then the reasoning about the interactions between the agent representing the coalition and the other agents (one-to-many interactions) can be directed by the wrapper decision mechanisms. However, the suggestion here is to increase the social welfare function of the coalition by supplying within the wrapper adaptive algorithms that assist the representative agent to dynamically modify the reservation values of each of the issues given multiple offers from a number of other agents. The suggestion is that the wrapper can be used to reason not about how to form a coalition, but how to behave on behalf of the coalition. Note also that this functionality can be applied in normal one-to-many service negotiations. Work is currently underway to investigate multi-lateral protocols and negotiation decision strategies for design of an exchange system where N number of sellers engage in parallel negotiations with a single buyer for the procurement of a service. Decision functions are currently being developed that generate offers based on simultaneous consideration of many threads of negotiation.

6.3.3 Extensions to the Evaluation Work

The penultimate proposal for future work is to further evaluate the developed coordination framework. Although the wrapper has been empirically tested in a number of environments, this evaluation has nonetheless been carried out within a limited environment (e.g. interactions are only amongst agents that adopt the same wrapper architecture). Thus, the observed results are only valid for two agents that utilize a wrapper architecture. Although control measures were included and the results were compared to optimal solutions, it would be interesting to perform comparative evaluation of the performance of an agent utilizing an agent architecture derived from the proposed negotiation model and one that utilizes some other architecture. This comparative study can then be used to benchmark the performance standard of different architectures with respect to the optimal solutions. One possibility of performing such a comparative evaluation is the submission of the architecture (or its output as a strategy) to market competitions such as the Trading Agent Competition held at ICMAS 2000 (TAC 2000) where trading agents bid to buy and sell goods, in order to maximize a given objective based on the goods bought and sold and the prices of the exchanges. In such cases, the coordination framework can then be used as a "laboratory" to test which of the possible set of strategies are likely to perform the best in the competition.

6.3.4 Extensions to Other Application Domains

Finally, another line of future work is to extend the application of the coordination framework to other types of problems. The configurability requirement has been one of the central design concerns of the framework. As was shown in the first chapter, its application to seven different domains has procedurally demonstrated the configurability claim. However, further evaluation of this claim is required. Specifically, better metrics are required that test the applicability of the framework to different domains. Indeed, such an evaluation is intended to be carried out in a future application of the framework at The Center for Coordination Sciences at MIT. The aim of this project is to use the developed coordination framework for system recovery in cases when exceptions occur, such as the failure of a single agent, corrupted or invalid information within the system or erroneous execution schedules. In such cases agents can enter negotiation to either prevent predicted future failures or recover from failures that have occurred (Dellarocas & Klein 2000). Because exceptions can occur across many different types of domains then domain problem solvers require social interactions to recover from such failures. Thus the configurability of the framework (as well as the benefits of negotiating agent technology in comparison to traditional methods) can be evaluated more objectively.

Bibliography

Adler, M., Davis, A., Weihmayer, R., and Worrest, W. 1989. Conflict resolution strategies for non-hierarchical distributed agents. In Gasser, L., and Huhns, M., eds., *Distributed Artificial Intelligence*, volume II. Pitman Publishing: London and Kaufmann; San Mateo, CA. 139–16.

Admati, A., and Perry, M. 1987. Strategic delay in bargaining. *Review of Economical Studies* 54:345–364.

Aho, A., Hopcroft, J., and Ullman, J. 1985. *Data Structures and Algorithms*. Reading, Massachusetts, USA: Addison-Wesley Series in Computer Science and Information Processing.

Amazon. `http://www.amazon.com`.

Arrow, K. 1950. A difficulty in the concept of social welfare. *Journal of Political Economy* 58:328–346.

AuctionBot. http://auction.eecs.umich.edu.

Aumann, R. 1976. Agreeing to disagree. *Annals of Statistics* 4:1236–9.

Axelrod, R. 1984. *The Evolution of Cooperation*. New York, USA.: Basic Books, Inc., Publishers.

Bannock, G., Baxter, R., and Davis, E. 1992. *Dictionary of Economics*. London, UK.: Penguin, 5 edition.

Barbuceanu, M., and Fox, M. 1997. Integrating communication action, conversations and decision theory to coordinate agents. In *Proceedings of the First International Conference on Autonomous Agents, Marina del Rey (Agents-97),Marina Del Rey*, 47–58.

Barbuceanu, M., and Lo, W. 2000. A multi-attribute utility theoretic negotiation architecture for electronic commerce. In *Proceedings of the Fourth International Conference on Autonomous Agents, Barcelona, Spain (Agents-2000)*, 239–247.

Binmore, K., and Dasgupta, P. 1986. *Economic organizations as games*. Oxford, UK: Basil Blackwell Ltd.

Binmore, K., and Dasgupta, P. 1989. *The Economics of Bargaining*. Oxford, UK: Basil Blackwell.

Binmore, K., and Vulkan, N. 1997. Applying game theory to automated negotiation. *DIMACS Workshop on Economics, Game Theory and the Internet, April 18–19, Rutgers University*.

Binmore, K., Shaked, A., and Sutton, J. 1984. An outside option experiment. *Quarterly Journal of Economics* 104:753–770.

Binmore, K. 1985. Bargaining and coalitions. In A.E.Roth., ed., *Game-Theoretic Models of Bargaining*. Cambridge University Press.

Binmore, K. 1990. *Essays on the foundations of game theory*. Oxford, UK: Basil Blackwell.

Binmore, K. 1992. *Fun and Games: A Text on Game Theory*. Lexington, Massachusetts: D.C. Heath and Company.

Bond, A., and Gasser, L. 1988. *Readings in Distributed Artificial Intelligence*. San Mateo, California: Morgan Kaufmann Publishers, Inc..

Bratman, M. 1990. What is intention? In P.R. Cohen, J. M., and Pollack, M., eds., *Intention in Communication*. MIT Press. 15–31.

Brenner, W., Zarnekow, R., and Wittig, H. 1998. *Intelligent Software Agents: Foundations and Applications*. Berlin: Springer-Verlag.

Brooks, R. 1991. Intelligence without representation. *Artificial Intelligence* 47(1):139–159.

Bui, H., Kieronska, D., and Venkatesh, S. 1996. Negotiating agents that learn about others' preferences. In *Papers from the AAAI 96 Spring Symposium on Adaptation, Co-evolution, and Learning in Multiagent Systems, March 25–27, Stanford University, California*, 16–21.

Burmeister, B., Haddadi, A., and Matylis, G. 1997. Applications of multi-agent systems in traffic and transportation. *IEEE Transactions on Software Engineering* 144(1):51–60.

Busuoic, M., and Griffits, D. 1994. Cooperating intelligent agents for service management in communications networks. In Deen, S., ed., *Proceeding of the 1993 Workshop on Cooperating Knowledge Based Systems (CKBS-93)*, 213–226. DAKE Center, University of Keel,UK.

Carley, K., and Gasser, L. 1999. *Computational Organization Theory*. Cambridge, Massachusetts: The MIT Press. 299–330.

CASBA. 2000. `http://www.casba-market.org/`.

Castlefranchi, C., and Conte, R. 1997. Limits of strategic rationality for agents and MA Systems. In Boman, M., and de Velde, W. V., eds., *Multi-Agent Rationality: Proceedings of the 8th European Workshop on Modeling Autonomous Agents in Multi-Agent World, MAAMAW'97*, number 1237 in Lecture Notes in Artificial Intelligence. Springer-Verlag.

Castlefranchi, C. 1995. Guarantees for autonomy in cognitive agent architecture. In Wooldridge, M., and Jennings, N., eds., *Intelligent Agents: Theories, Architectures, and Languages*, LNAI,890, 56–70. Springer-Verlag.

Chavez, A., and Maes, P. 1996. Kasbah: An agent marketplace for buying and selling goods. In *Proceedings of the First International Conference on Practical Applications of Intelligent Agents and Multi-Agent Technology (PAAM'96),London, UK.*, 75–90.

Chen, L., and Sycara, K. 1998. Webmate. a personal agent for browsing and searching. In *Proceeding of the Second International Conference on Autonomous Agents (Agents-98), Minneapolis/St Paul, MN*, 132–139.

Clearwater, S., Costanza, R., Dixon, M., and Schroede, B. 1996. Saving energy using market-based control. In S.H.Clearwater., ed., *Market Based-Control*. World Scientific:Singapore. 253–273.

Coad, P., and Yourdon, E. 1991. *Object-Oriented Design*. Englewood Cliffs: Yourdon Press.

Cohen, P. R., and Levesque, H. 1990. Intention is choice with commitment. *Artificial Intelligence* 42:213–261.

Cohen, P., and Perrault, C. 1979. Elements of a plan-based theory of speech acts. *Cognitive Science* 3:177–212.

Cohen, P. 1995. *Empirical Methods for Artificial Intelligence*. Cambridge, Massachusetts: MIT Press.

Coleman, J. 1973. *The Mathematics of Collective Action*. Chicago, IL: Aldine.

Corfman, K. P., and Gupta, S. 1993. Mathematical models of group choice and negotiations. *Handbooks in Operation Research and Management Sciences* 5:83–142.

Corkill, D., and Lesser, V. 1983. The use of meta-level control for coordination in distributed problem solving network. In *Proceeding of the 8th International Joint Conference on Artificial Intelligence*, Karlsruhe, Germany, 748–756.

Coulouris, G., Dollimore, J., and Kindberg, T. 1994. *Distributed Systems*. Harlow, England: Addison Wesley Publishing Company, second edition.

Cramton, P. 1991. Dynamic bargaining with transaction costs. *Management Science* 37(10):1219–1233.

Cranor, L. F., and Resnick, P. 2000. Protocols for automated negotiations with buyer anonymity and seller reputations. *Netnomics* 2(1):1–23.

Cross, J. 1969. *The Economics of Bargaining*. New York: Basic Books.

Curry, D., Menasco, M., and van Ark, J. 1991. Multi-attribute dyadic choice. Models and tests. *Journal of Marketing Research* 28(3):259–267.

DAIS. 1984. A platform to build on, icl corporate systems publications, `http://www.icl.com/`.

Damme, E. V. 1986. The nash bargaining solution is optimal. *Journal of Economic Theory* 38:78–100.

Debreu, G. 1959. *Theory of Value: An Axiomatic Analysis of Economic Equilibrium*. New York: Wiley.

Decker, K., and Li, J. 1998. Coordinating hospital patient scheduling. In *Proceedings of the International Conference on MultiAgent Systems (ICMAS 98), Paris, France.*, 104–111.

Decker, K. 1995. *Environment Centered Analysis and Design of Coordination Mechanisms*. Ph.D. Dissertation, Department of Computer Science, University of Massachusetts, Amherst.

Dellarocas, C., and Klein, M. 2000. An experimental evaluation of domain-independent fault handling services in open multi-agent systems. In *Proceedings of International Conference on Multi-Agent Systems (ICMAS-2000), July 2000, Boston, MA*.

Dennett, D. 1987. *The Intentional Stance*. Cambridge, MA.: Bradford Books/MIT Press.

Doorenbos, R., Etzioni, O., and Weld, D. 1997. A scalable comparison-shopping agent for the world wide web. In *Proceeding of the First International Conference on Autonomous Agents (Agents-97), Marina del Rey, CA*, 39–48.

Durfee, E., and Lesser, V. 1989. Negotiating task decomposition and allocation using partial global planning. In Gasser, L., and Huhns, M., eds., *Distributed Artificial Intelligence, Volume 2*, 229–243. Morgan Kaufmann Publishers, Inc.

Durfee, E. H., and Rosenschein, J. S. 1994. Distributive problem solving and multi–agent systems: Comparisons and examples. In *Proceedings of the 13th International Workshop on Distributed Artificial Intelligence,Seatle, WA*, 94–104.

Durfee, E., Lesser, V., and Corkill, D. 1988. Cooperation through communication in a distributed problem solving network. In Bond, A., and Gasser, L., eds., *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann Publishers, Inc. 29–58.

Durfee, E. 1998. Practical negotiation strategies (extended abstract). In *Proceedings of the International Workshop on MultiAgent Systems (IWMAS 98), 12–15 October, MIT. Massachusetts*.

eBay. `http://www.ebay.com`.

Ephrati, E., and Rosenschein, J. S. 1994. Multi-agent planning as search for a consensus that maximises social welfare. In Castelfranchi, C., and Werner, E., eds., *Artificial Social Systems — Selected Papers from the Fourth European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW-92 (LNAI Volume 830)*, 207–226. Springer-Verlag: Heidelberg, Germany.

Etzioni, O. 1996. Moving up the information food chain: Deploying softbots on the world wide web. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference*, 1322–1326. Menlo Park: AAAI Press / MIT Press.

FACTS. 1998. `http://www.labs.bt.com/profsoc/facts/`.

Faratin, P., Sierra, C., Jennings, N., and Buckle, P. 1999a. Designing flexible automated negotiators: Concessions, trade-offs and issue changes. Technical Report RP-99-03, Institut d'Investigacio en Intel.ligencia Artificial Technical Report.

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

Faratin, P., Jennings, N. R., Buckle, P., and Sierra, C. 2000. Automated negotiation for provisioning virtual private networks using fipa-compliant agents. In *Proceedings of the Fifth International Conference on The Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM-2000), Manchester, UK*, 185–202.

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

Faratin, P., Sierra, C., and Jennings, N. R. 2000. Using similarity criteria to make negotiation trade-offs. In *Proceedings of the International Conference on Multiagent Systems (ICMAS-2000), Boston, MA.*, 119–126.

Faris, R. 1953. Development of small group research movement. In Wilson, M., ed., *Group Relations at the Crossroads*. New York: Harper and Brothers. 155–184.

Finin, T., and Fritzson, R. 1994. KQML–A language and protocol for knowledge and information exchange. In *Proceeding of the International Distributed Artificial Intelligence Workshop, Seattle*, 126–136.

FIPA-OS. 2000. `http://www.nortelnetworks.com/products/announcements/fipa/index.html`.

FIPA97. 1997. `http://www.fipa.org/spec/FIPA97.html`.

Fischer, K., and Ury, W. 1981. *Getting to Yes: Negotiating Agreement without giving in*. Boston, MA: Houghton Mifflin.

Fischer, K., Mủller, J., and Pischel, M. 1996. Cooperative transportation scheduling: An application domain for DAI. *Applied Artificial Intelligence* 10(1):1–34.

Fishburn, P. 1981. Subjective expected utility. *Theory and Decision* 13:139–99.

FishMarket. `http://www.iiia.csic.es/projects/fishmarket`.

Fudenberg, D., and Tirole, J. 1983. Sequential bargaining with incomplete information. *Review of Economical Studies* 50:221–247.

Gasser, L. 1991. Social conceptions of knowledge and action. *Artificial Intelligence* 47:107–138.

Genesereth, M., Ginsberg, M., and Rosenchein, J. 1986. Cooperation without communication. In *Proceedings of the National Conference on Artificial Intelligence, Philadelphia, Pennsylvania*, 51–57.

Georgeff, M. 1983. Communication and interaction in multi-agent planning. In *Proceedings of the National Conference on Artificial Intelligence*, 125–129.

Gibbons, R. 1992. *A Primer in Game Theory*. New York: Harvester Wheatsheaf.

Grand, S., and Cliff, D. 1998. Creatures: Entertainment Software Agents with Artifical Life. *Journal of Autonomous Agents and Multi-Agent Systems* 1(1):39–57.

Griffeth, N., and Velthuijsen, H. 1994. The negotiating agents approach to run-time feature interaction resolution. In L.G.Bouma, and H.Velthuijsen., eds., *Feature Interactions in telecommunications Systems*. IOS Press. 217–235.

Gruber, T. 1994. A translation approach to potable ontology specification. *Knowledge Acquisition* 5:199–220.

Guha, R., and Lenat, D. 1990. Cyc: A mid term report. *Artificial Intelligence Magazine* 11(3):32–59.

Guillbaud, G. 1966. Theories of general interest, and the logical problem of aggregation. In Lazarsfeld, P., and Henry, N., eds., *Readings in Mathematical Social Sciences*. Chicago: Science Research Associates Inc. 262–307.

Gupta, S., and Livine, Z. 1988. Resolving a conflict situation with a reference outcome: An axiomatic model. *Management Science* 34(11):1303–1314.

Gupta, S. 1989. Modeling integrative multiple issue bargaining. *Management Sciences* 35(7):788–806.

Guth, W., Schmittberger, K., and Schwarze, B. 1982. An experimental analysis of ultimatum bargaining. *Journal of Economic Behaviour and Organization* 3:367–88.

Guttman, R., and Maes, P. 1998. Agent-mediated integrative negotiation for retail electronic commerce. In Noriega, P., and Sierra, C., eds., *Agent Mediated Electronic Commerce: First International Workshop on Agent Mediated Electronic Trading, AMET-98, Minneapolis, MN, USA*, number 1571 in LNAI. Springer-Verlag. 70–91.

Harsanyi, J. 1955. Cardinal welfare, individualistic ethics, and interpersonal comparison of utility. *Journal of Political Economy* 63:309–321.

Harsanyi, J. 1956. Approaches to the bargaining problem before and after the theory of games. *Econometrica* 24:144–157.

Harsanyi, J. 1967-1968. Games with incomplete information played by bayesian players, parts i, ii, iii. *Management Science* 14.

Hayes-Roth, B., Hewett, M., Washington, R., Hewett, R., and Seiver, A. 1989. Distributed intelligence within an individual. In Gasser, L., and Huhns, M., eds., *Distributed Artificial Intelligence*, volume II. Pitman Publishing: London and Kaufmann; San Mateo, CA. 385–412.

Hayes-Roth, B. 1995. An architecture for adaptive intelligent systems. *Artificial Intelligence: Special Issue on Agents and Interactivity* 37:329–365.

Hewitt, C., and de Jong, P. 1984. Open systems. In Brodie, M., Mylopoulos, J., and Schmidt, J., eds., *On Conceptual Modeling — Perspectives from Artificial Intelligence, databases and Programming*. Springer Verlag. 147–164.

Holt, A. 1988. Diplans — a new language for the study of co-ordination. *ACM transactions on Office Information Systems* 109–125.

Huhns, M., and Singh, M. 1997. Ontologies for agents. *IEEE Internet Computing* 1(6):81–83.

Huhns, M., and Singh, M. 1998. Managing heterogeneous transaction workflows with cooperating agents. In Jennings, N., and Wooldridge, M., eds., *Agent Technology: Foundations, Applications and Markets*. Springer-Verlag. 219–239.

Huhns, M., and Stephens, L. 1999. Multiagent systems and societies of agents. In Weiss, G., ed., *Multiagent Systems*. Cambridge, Massachusetts: The MIT Press. 79–121.

i2. `http://www.i2.com`.

Ingrand, F., Georgeff, M., and Rao, A. 1992. An architecture for real-time reasoning and system control. *IEEE Expert* 7(6):34–44.

Ito, A., and Yano, H. 1995. The emergence of cooperation in a society of autonomous agents - the prisoner's dilemma game under the disclosure of contract histories. In Lesser, V., ed., *Proceedings of the First International Conference on Multi-Agent Systems*, 201–208. San Francisco, USA: AAAI Press/The MIT Press.

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

Jennings, N. R., Faratin, P., Johnson, M. J., O'Brien, P., and Wiegand, M. E. 1996b. Using intelligent agents to manage business processes. In *Proceedings of First International Conference on The Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM96), London, UK.*, 345–360.

Jennings, N. R., Faratin, P., Norman, T. J., OBrien, P., Wiegand, M. E., Voudouris, C., Alty, J. L., Miah, T., and Mamdani, E. H. 1996c. Adept: Managing business processes using intelligent agents. *Proceeding of the British Computer Society Expert Systems 96 Conference (ISIP Track), Cambridge, UK* 5–23.

Jennings, N., Corera, J., Laresgoiti, I., Mamdani, E., Perriolat, F., Skarek, P., and Varga, L. 1996d. Using archon to develop real-world dai applications for electricity transportation management and particle acceleration control. *IEEE Expert* 11(6):60–88.

Jennings, N. R., Faratin, P., Norman, T. J., O'Brien, P., and Odgers, B. 2000a. Autonomous agents for business process management. *Int. Journal of Applied Artificial Intelligence.* 14(2):145–189.

Jennings, N. R., Faratin, P., Norman, T. J., OBrien, P., Odgers, B., and Alty, J. L. 2000b. Implementing a business process management system using adept: A real-world case study. *International Journal of Applied Artificial Intelligence* 14(5):421–463.

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

Jennings, N. R. 1993. Commitments and conventions; The foundations of coordination in Multi-Agent systems. *Knowledge Engineering Review* 8(3):223–250.

Jennings, N. 1996. Coordination techniques for distributed artificial intelligence. In O'Hare, G. M. P., and Jennings, N. R., eds., *Foundations of Distributed Artificial Intelligence*, Sixth-Generation Computer Technology Series. New York: John Wiley. 187—211.

Jiang, Y., Kautz, H., and Selman, B. 1995. Solving problems with Hard and Soft constraints using a stochastic algorithms for maxsat. In *First International Joint Workshop on Artificial Intelligence and Operations Research, Timberland, Oregon*.

Johnson, E., and Payne, J. 1985. Effort and accuracy in choice. *Management Science* 31(4):395–414.

Kahan, J., and Rapoport, A. 1984. *Theories of Coalition Formation*. Lawrence Erlbaum Associates Publishers.

Kalai, E., and Smorodinsky, M. 1975. Other solutions to nash's bargaining problem. *Econometrica* 43:513–518.

Kautz, H., Selman, B., and Shah, M. 1997. The hidden webs. *AI Magazine* 18(2):27–35.

Keeney, R. L., and Raiffa, H. 1976. *Decisions with Multiple Objectives*. New York: John Wiley and Sons.

Kiss, G. 1992. Variable coupling of agents to their environment: Combining situated and symbolic automata. In Wener, E., and Demazeau, Y., eds., *Distributed Artificial Intelligence*, 231–248.

Knuth, D., and Moore, R. 1975. An analysis of alpha-beta pruning. *Artificial Intelligence* 6(4):293–326.

Kolodner, J. 1993. *Case-based Reasoning*. San Mateo: Morgan Kaufmann.

Kraus, S., and Lehmann, D. 1995. Designing and building negotiation automated agent. *Computational Intelligence 11* 11(1):132–171.

Kraus, S., and Wilkenfeld, J. 1993. A strategic negotiations model with application to an international crisis. *IEEE Transactions on Systems Man and Cybernetics* 23(1):313–323.

Kraus, S., and Wilkenfeld, J. 1995. Multiagent negotiation under time constraints. *Artificial Intelligence Journal* 75(2):297–345.

Kraus, S., Nirkhe, N., and Sycara, K. 1998. Reaching agreements through argumentation: a logical model and implementation. *Artificial Intelligence Journal* 104(1–2):1–69.

Kraus, S., Wilkenfeld, J., and Zlotkin, G. 1995. Multiagent negotiation under time constraints. *Artificial Intelligence Journal* 75(2):297–345.

Kraus, S. 1997a. Beliefs, time and incomplete information in multiple encounter negotiations among autonomous agents. *Annals of Mathematics and Artificial Intelligence* 20(1–4):111–159.

Kraus, S. 1997b. Negotiation and cooperation in multi-agent environments. *Artificial Intelligence* 94(1–2):79–97.

Kraus, S. 2000. *Strategic Negotiation in Multi-Agent Environments*. Cambridge, USA: MIT Press.

Kreps, D., and Wilson, R. 1982. Sequential equilibria. *Econometrica* 50:863–894.

Kreps, D. 1990. *A Course in Microeconomic Theory*. Princeton University Press.

Krulwich, B. 1996. The bargainfinder agent: Comparison price shopping on the internet. In Williams, J., ed., *Bots, and other Internet Beasties*, volume II. Macmillan Computer Publishing: Indianapolis. 257–263.

Laughlin, P., and Earley, P. 1982. Social combination models persuasive arguments theory, social comparison theory, and choice shift. *Journal of Personality and Social Psychology* 42:273–280.

Laughlin, P. 1980. Social combination processes of cooperative problem solving groups on verbal intellectual tasks. In Fishbein, M., ed., *Progress in Social Psychology*. Hilsdale, N.J.: Earlbaum.

Lemel, G. 1995. A strategic model for negotiation among autonomous agents. Master's thesis, Department of Mathematics and Computer Science, Bar-Ilan University, Ramat Gan (written largely in Hebrew).

Lenat, D. 1975. Beings: Knowledge as interacting experts. *IJCAI* 4:126–133.

Lesser, V. R. 1998. Reflections on the Nature of Multi-Agent Coordination and Its Implications for an Agent Architecture. *Journal of Autonomous Agents and Multi-Agent Systems* 1(1):89–111.

Liberman, H. 1995. Letizia: An agent that assists web browsing. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95), Montréal,Québec,Canada*, 924–929.

Ljungberg, M., and Lucas, A. 1992. The oasis air-traffic management system. In *Proceedings of the Second Pacific Rim International Conference on Artificial Intelligence (PRICAI '92)*.

Luce, R. D., and Raiffa, H. 1957. *Games and Decisions*. New York: Wiley.

Luenberger, D. 1973. *Introduction to linear and nonlinear programming*. Addison-Wesley.

MacCrimmon, K., and Messick, D. 1976. A framework for social motives. *Behavioural Sciences* 21:86–100.

Maes, P. 1995. Artificial life meets entertainment: Life like autonomous agents. *Communication of ACM* 38(11):108–114.

Malone, T., and Crowston, K. 1990. What is co-ordination theory and how can it help design co-operative work systems. *Proceeding of the International Conference on Computer Supported Co-operative Work (CSCW–90)* 357–370.

Malone, T., Fikes, R., Grant, K., and Howard, M. 1988. Enterprise: A market-like task scheduler for distributed computing environments. In Huberman, B., ed., *The Ecology of Computation*. Amsterdam: North-Holland. 177–205.

Malone, T. 1990. Organizing information processing systems: Parallels between human organizations and computer systems. In *Cognition, Computation and Cooperation*. Norwood, NJ.: Ablex. 56–83.

Marr, D. 1982. *Vision : a computational investigation into the human representation and processing of visual information*. San Francisco: W.H. Freeman.

Marschak, J., and Radner, R. 1972. *Economic Theory of Teams*. New Haven: Yale University Press.

Marsland, A., and Schaeffer, J. 1990. *Computers, Chess and Cognition*. Berlin: Springer-Verlag.

Matos, N., Sierra, C., and Jennings, N. R. 1998. Determining successful negotiation strategies: an evolutionary approach. In *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS-98), Paris, France*, 182–189.

Mertens, J. F., and Zamir, S. 1985. Formulation of Bayesian analysis for games of incomplete information. *International Journal of Game Theory* 14(1):1–29.

Milgrom, P., and Roberts, J. 1982. Predation, reputation, and entry deterrence. *Journal of Economic Theory* 27:280–312.

Mitten, L. 1970. Branch and bound methods: General formulations and properties. *Operation Research* 18:23–34.

Mowbray, T., and Zahavi, R. 1995. *The essential CORBA system integration using distributed objects*. John Wiley and Object Management Group.

Mullen, T., and Wellman, M. 1995. A simple computational market for network information services. In *Proceedings of the International Conference on MultiAgent Systems (ICMAS 95), San Francisco, California*, 283–289. AAAI/MIT Press.

Nash, J. 1950. The bargaining problem. *Econometrica* 18:155–162.

Nash, J. 1951. Non-cooperative games. *Annals of Mathematics* 54:286–95.

Neches, R., Fikes, R., Finin, T., Gruber, R., Patil, R., Senator, T., and Swartout, W. 1991. Enabling technology for knowledge sharing. *AI Magazine* 12(3):36–56.

Neumann, J. V., and Morgernstern, O. 1944. *The Theory of Games and Economic Behaviour*. Princeton, N.J.: Princeton University Press.

Newton, I. 1972. Principia mathematica. In Koyr, A., and Cohen, I., eds., *Isaac Newton's Philosophiae Naturalis Principia Mathematica*. Cambridge, Mass.: Harvard University Press,.

Norman, T. J., Jennings, N. R., Faratin, P., and Mamdani, E. H. 1996. Designing and implementing a multi-agent architecture for business process management. In Mueller, J. P., Wooldridge, M. J., and Jennings, N. R., eds., *Intelligent Agents III*, volume 1193 of *Lecture Notes in Artificial Intelligence*. Springer Verlag. 161–275.

Oliver, J. 1994. Finding decision rules with genetic algorithms. *AI Expert* 9(3):32–39.

OMG. 1996. The common object request broker: Architecture and specification, revision 2. OMG Technical Document formal/97-02-25, `http://www.omg.org/corba/corbiiop.htm`.

orbix. 2000. `http://www.usa-iona.com/`.

Osborne, M., and Rubinstein, A. 1990. *Bargaining and Markets*. San Diego, California: Academic Press Inc.

Ossowski, S. 1999. *Co-ordination in artificial agent societies*. Berlin.: Springer Verlag.

Parsons, S., Sierra, C., and Jennings, N. R. 1998. Agents that reason and negotiate by arguing. *Journal of Logic and Computation* 8(3):261–292.

Parunak, H. V. D. 1987. Manufacturing experience with the contract net. In Huhns, M., ed., *Distributed Artificial Intelligence*. Pitman Publishing: London and Morgan Kaufmann: San Mateo, CA. 285–310.

Parunak, H. V. D. 1999. Industrial and practical applications of dai. In Weiss, G., ed., *Multiagent Systems*. Cambridge, Massachusetts: The MIT Press. 377–425.

Payne, J. 1976. Task complexity and contingent processing in decision making: An information search and protocol analysis. *Organization Behaviour and Human Performance* 16:366–386.

Pedrycz, W., and Comide, F. 1998. *An Introduction to Fuzzy Sets: Analysis and Design*. Cambridge, Massachusetts: MIT Press.

Perry, M. 1986. An example of price formation in bilateral situations: A bargaining model with incomplete information. *Econometrica* 50:97–109.

Pruitt, D. 1981. *Negotiation Behavior*. New York: Academic Press.

Raiffa, H. 1982. *The Art and Science of Negotiation*. Cambridge, USA: Harvard University Press.

Rao, A., and Georgeff, M. 1990. Intelligent real-time network management. In *Proceedings of the Tenth International Conference on AI, Expert Systems and Natural Language,Avignon, France*.

Rao, A., and Georgeff, M. 1991. Modeling rational agents within a BDI architecture. In Allen, J., Fikes, R., and Sandwall, E., eds., *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, 473–484. San Mateo, CA: Morgan Kaufmann.

Rapoport, A. 1964. *Strategy and Conscience*. New York: Harper and Row.

Rasmusen, E. 1989. *Games and Information*. Oxford, UK: basil Blackwell.

Rodríguez, J., Noriega, P., Sierra, C., and Padget, J. 1997. FM96.5 a java-based electronic auction house. In *In Second International Conference on The Practical Application of Intelligent Agents and Multi-Agent Technology: (PAAM-97), London, UK*, 207–224.

Rosenchein, J. S., and Genesereth, M. 1985. Deals among rational agents. In *Proceedings of the ninth International Conference on Artificial Intelligence, Los Angles, California*, 91–99.

Rosenschein, J. S., and Zlotkin, G. 1994. *Rules of Encounter*. Cambridge, USA: The MIT Press.

Roth, A. 1985. Toward a focal point theory of bargaining. In Roth, A., ed., *Game-Theoretic Models of Bargaining*. Cambridge University Press, NY. 259–268.

Roth, A. 1995. Bargaining experiments. In Kagel, J., and Roth, A., eds., *Handbook of Experimental Economics*. Princeton University Press. 253–348.

Rubinstein, A. 1982. Perfect equilibrium in a bargaining model. *Econometrica* 50:97–109.

Rubinstein, A. 1985a. A bargaining model with incomplete information about time preferences. *Econometrica* 53:1151–1172.

Rubinstein, A. 1985b. Toward a focal point theory of bargaining. In Roth, A., ed., *Game-Theoretic Models of Bargaining*. Cambridge University Press, NY. 99–114.

Russell, S., and Norvig, P. 1995. *Artificial Intelligence: A modern approach*. Upper Saddle River, New Jersey: Prentice Hall.

Russell, S., and Wefald, E. 1991. *Do the Right Thing*. The MIT Press.

Sandholm, T., and Lesser, V. 1997. Coalitions among computationally bounded agents. *Artificial Intelligence* 94(1):99–137.

Sandholm, T. 1993. An implementation of the contract net protocol based on marginal cost calculations. In *Proceedings of the National Conference on Artificial Intelligence*, 256–262.

Sandholm, T. 1996. *Negotiation Among Self-Interested Computationally Limited Agents*. Ph.D. Dissertation, Department of Computer Science, University of Massachusetts Amherst.

Sandholm, T. 1999. Distributed rational decision making. In Weiss, G., ed., *Multiagent Systems*. Cambridge, Massachusetts: The MIT Press. 201–259.

Sandholm, T. 2000. `http://www.cs.wustl.edu/ sandholm`.

Sathi, A., and Fox, M. 1989. Constraint-directed negotiation of resource reallocation. In Gasser, L., and Huhns, M., eds., *Distributed Artificial Intelligence Volume II*, 163–195. San Mateo, California: Morgan Kaufmann.

Schank, R., and Abelson, R. 1977. *Scripts, Plans, Goals, and Understanding*. Hillsdale, NJ.: Lawrence Earlbaum Associates.

Schelling, T. 1960. *The Strategy of Conflict*. Cambridge, MA: Harvard University Press, 2nd edition.

Schwuttke, U., and Quan, A. 1993. Enhancing performance of cooperating agents in real-time diagnostic systems. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93), Chambéry, France*, 332–337.

Scott, W. 1987. *Organizations: Rational, Natural, and Open Systems*. Englewood Cliffs, NJ.: Prentice Hall, Inc.

Searle, J. 1990. Collective intentions and actions. In Cohen, P., Morgan, J., and Pollack, M., eds., *Intentions in Communication*. Cambridge, MA.: MIT Press. 401–416.

Selman, B., Levesque, H., and Mitchell, D. 1992. A new method for solving hard satisfiability problems. In *Proceedings of American Association of Artificial Intelligence (AAAI-92), San Jose, California*, 440–446.

Sen, S. 1994. The role of commitment in cooperative negotiation. *International Journal on Intelligent Cooperative Information Systems* 3(1):67–81.

Shehory, O., and Kraus, S. 1995. Task allocation via coalition formation among autonomous agents. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95), Montréal,Québec,Canada*, 655–661.

Shuffle. 2000. `http://www.ist-shuffle.org/`.

Sierra, C., Faratin, P., and Jennings, N. 1997. A service-oriented negotiation model between autonomous agents. In Boman, M., and de Velde, W. V., eds., *Proceedings of 8th European Workshop on Modeling*

Autonomous Agents in Multi-Agent World, number 1237 in Lecture Notes in Artificial Intelligence, 17–35. Springer-Verlag.

Sierra, C., Faratin, P., and Jennings, N. 1999. Deliberative automated negotiators using fuzzy similarities. In *Proceedings of EUSFLAT-ESTYLF Joint Conference on Fuzzy Logic, Palma de Mallorca, Spain*, 155–158.

Simon, H. A. 1982. *Models of Bounded Rationality, Volume 2*. Cambridge, Massachusetts.: MIT Press.

Simon, H. A. 1996. *The Sciences of the Artificial*. Cambridge, Massachusetts.: MIT Press.

Singh, M. 1994. *Multiagent Systems: A theoretical framework for intentions, know-how and communications*. New York: Springer.

Smith, A. 1776. *The Wealth of Nations*.

Smith, R. 1980. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers* C-29(12):1104–1113.

Smith, J. M. 1982. *Evolution and the Theory of Games*. Cambridge: Cambridge University Press.

Sobel, J., and Takahashi, I. 1983. A multi-stage model of bargaining. *Review of Economical Studies* 50:411–426.

Sutton, J. 1986. Non-cooperative bargaining theory: An introduction. *Review of Economical Studies* 53:709–724.

Svenson, O. 1979. Process descriptions of decision making. *Organization Behaviour and Human Performance* 23:86–112.

Sycara, K., Decker, K., Pannu, A., Willamson, M., and Zeng, D. 1996. Distributed artificial agents. *IEEE Expert* 11(6).

Sycara, K. 1987. *Resolving Adversarial Conflicts: An Approach Integrating Case-Based reasoning and Analytical Methods*. Ph.D. Dissertation, School of Information and Computer Science, Georgia Institute of Technology.

Sycara, K. 1989. Multi-agent compromise via negotiation. In Gasser, L., and Huhns, M., eds., *Distributed Artificial Intelligence Volume II*, 119–139. San Mateo, California: Morgan Kaufmann.

TAC. 2000. `http://tac.eecs.umich.edu/`.

Topkis, D. 1988. *Supermodularity and Complementarity*. Princeton, NJ: Princeton University Press.

Tsvetovatyy, M., Gini, M., Mobasher, B., and Wieckowski, Z. 1997. Magma: An agent-based virtual marketplace for electronic commerce. *Applied Artificial Intelligence* 11(6):501–524.

Tversky, A. 1969. Intransitivity of preferences. *Psychological Review* 76:31–48.

Valverde, L. 1985. On the structure of F-indistinguishability. *Fuzzy Sets and Systems* 17:313–328.

Varian, H. 1992. *Microeconomic analysis*. New York: W.W. Norton.

Vincent, D. 1989. Bargaining with common values. *Journal of Economic Theory* 48:47–62.

Vogel, A. 1996. *Java programming with CORBA, OMG*. New York: John Wiley.

Vulkan, N., and Jennings, N. 1998. Efficient mechanisms for the supply of services in multi-agent environments. In *Proceedings of the International Conference on MultiAgent Systems (ICMAS-98), Paris, France*.

Vulkan, N., and Jennings, N. R. 2000. Efficient mechanisms for the supply of services in multi-agent environments. *Int Journal of Decision Support Systems* 28(1–2):5–19.

Walton, D. N., and Krabbe, E. C. 1995. *Commitments in Dialogue: basic Concepts of Interpersonal Reasoning*. New York.: State University of New York Press.

Wang, H., and Wang, C. 1997. Intelligent agents in the nuclear industry. *IEEE Computer* 30(11):28–34.

Werner, E. 1989. Cooperating agents: A unified theory of communication and social structure. In L.Gasser, and Huhns, M., eds., *Distributed Artificial Intelligence, Volume 2*, 3–36. Morgan Kaufmann Publishers, Inc.

Wilson, R. 1969. An axiomatic model of logrolling. *American Economic Review* 3:331–341.

Wooldridge, M., and Jennings, N. 1995. Intelligent agents: Theory and Practice. *The Knowledge Engineering Review* 10(2):115–152.

Yang, H., and Zhang, C. 1995. Definitions and application of a comprehensive framework for distributed problem solving. In Zhang, C., and Lukose, D., eds., *Distributed Artificial Intelligence — Architecture and Modeling*, number 1087 in LNAI, 1–15. Springer Verlag.

Young, O. R. 1975. *Bargaining: Formal Theories of Negotiation*. Urbana: University of Illinois Press.

Zadeh, L. A. 1971. Similarity relations and fuzzy orderings. *Information Sciences* 3:177–200.

Zeng, D., and Sycara, K. 1997. How an an agent learn to negotiate. In Mueller, J., Wooldridge, M., and Jennings, N., eds., *Intelligent Agents III. Agent Theories, Architectures, and Languages*, number 1193 in LNAI, 233–244. Springer Verlag.

Zlotkin, G., and Rosenchein, J. 1992. A domain theory for task oriented negotiation. *Proceeding of the American Association of Artificial Intelligence, San Jose, CA* 148–153.

Zlotkin, G., and Rosenchein, J. 1996. Mechanisms for automated negotiation in state oriented domains. *Journal of Artificial Intelligence Research* 5:163–238.