Interaction and Communication among Autonomous Agents in Multiagent Systems

A dissertation presented by

Nicoletta Fornara

Supervised by

Prof. Marco Colombetti

Dr. Luca Maria Gambardella

Submitted to

School of Communication Sciences

University of Lugano

for the degree of

Ph.D. in Communication Sciences

June 2003

Board:

Thesis advisor:	Marco Colombetti
Co-advisor:	Luca Maria Gambardella
Reviewer:	Carles Sierra, Gianpaolo Cugola

The work described in this thesis was carried out at Università della Svizzera italiana, Lugano, Switzerland, and at IDSIA (Istituto Dalle Molle di Studi sull'Intelligenza Artificiale), Lugano, Switzerland.

Abstract

The main goal of this doctoral thesis is to investigate a fundamental topic of research within the Multiagent Systems paradigm: the problem of defining *open interaction frameworks* in order to enable agent communicative interactions in open, heterogeneous and dynamic systems. That is to realize interaction systems where multiple agents can enter and leave dynamically, where no assumptions are made on the internal structure of the interacting agents, and that are defined using a method that enable agents designer to develop a single artificial agent that can interact with different systems designed by different organizations. Such topic of research has received much attention in the past few years. In particular the need to realize applications where artificial agents can interact, negotiate, exchange information, resources, and services, for instance in electronic commerce or information retrieval applications, has become more and more important thanks to the advent of Internet.

I started my studies on multiagent interaction systems and on their use to realize electronic commerce applications by developing a trading agent that took part to an international trading on-line game: the First Trading Agent Competition (TAC). During the design and development phase of the trading agent some crucial and critical troubles related to the TAC interaction system emerged. First the problem of accurately understanding the rules that govern the different auctions present in the game, and second the problem of understanding the meaning of the numerous messages that the registered trading agents can use to interact with the system. Another more general problem that became clear during the design phase of the trading agent is that its internal structure would have been strongly determined by the peculiar interface of the relevant interaction system, in fact the agent has to use a set of pre-defined methods to interact with the TAC server, consequently without any changes in its algorithms, it would not be able to take part to any other competition, even with slightly different rules or to communicate with any other interaction system present on Web. Furthermore the trading agent would not have been able to exploit opportunities, to handle unexpected situations, or to reason about the rules of the various auctions, since it is not able to understand the meaning o the exchanged messages. The presence of all those problems bears out the need to find a standard common accepted way to define open interaction systems.

The most important component of every interaction framework, as is remarked also by philosophical studies on human communication presented in Speech Acts Theory, is the *institution of language*. Following this approach I start to investigate the problem of defining a standard and common accepted semantics for Agent Communication Languages (ACL). Such a problem has received much attention in recent years but the solutions proposed so far are at best partial, and are considered as unsatisfactory by a large number of specialists. In particular, most current proposals are unable to support verifiable compliance to standards and to make agents responsible for their communicative actions. Furthermore

such proposals make the strong assumption that every interacting agent may be modelled as a Belief-Desire-Intention (BDI) agent.

What is required is an approach focused on externally observable events as opposed to the unobservable internal states of agents. Focusing on external events means to take into account the "social framework" within which agents interact. Following Speech Act Theory approach to human communication that views language use as a form of action, I propose an operational specification for the definition of a standard Agent Communication Language based on the notion of *social commitment*. In such a proposal the meaning of basic communicative acts is defined as the effect that sending the message has on the social relationship between the sender and the receiver of the message described through operation on an unambiguous, objective, and public "object": the commitment. The adoption of the notion of commitment is crucial to stabilize the interaction among agents, to create an expectation on other agents behavior, to enable agents to reason about their and other agents actions. Moreover given that this approach is inspired by speech act studies makes it possible to treat human and artificial agents communication in a uniform way, a crucial aspect to obtain successful mixed interactions.

The proposed Agent Communication Language is verifiable, that is, it is possible to determine if an agent is behaving in accordance to its communicative actions; the semantics is public, that is, any third part agent witness of the messages flow has to be able to draw similar inferences from the interaction, and objective in order that everybody attributes the same meaning to the exchanged messages. The proposed semantic is independent of the agent's internal structure, flexible and extensible to let agent cope with various and new situations, simple to be correctly used by agent designers, yet enough expressive.

A complete operational specification of an interaction framework able to support interactions among artificial agents using the proposed commitment-based Agent Communication Language is presented. In particular some sample applications of how to use the proposed framework to formalize interaction protocols available in interaction systems are reported. A list of *soundness conditions* to test if a protocol or a general interaction is sound is proposed. Such conditions express constrains on the content of the state of the interaction system at various stage of the conversation with regard to the meaning of the exchanged messages. The conversation protocols analyzed are the protocol of proposals and the protocol of offer that are widely used in electronic commerce applications.

To complete this research work a more complex interaction protocol, the English auction protocol, a protocol actually used in electronic commerce systems and adopted in the TAC game has been successfully formalized with the proposed framework. These positive results lead us to be optimist on the possibility of adopting the proposed framework to formalize many protocols that are actually used in the interaction systems operating on the web.

Dedicato a tutti coloro
che mi sono stati vicini in questi anni.

Acknowledgments

I would like to thank the Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), the Università della Svizzera italiana (USI), the Scuola Universitaria Professionale della Svizzera Italiana (SUPSI), and the European PLATFORM project (Computer-controlled freight platforms for a time-tabled rail transport system" 4th Framework Programme DG VII European Commission with the support of the Swiss OFES Ufficio Federale della Formazione Professionale e della Tecnologia project n. 97.0315) for their support of my research activity.

I would like to thank all IDSIA researchers for their suggestions, encouragement, and collaboration. In particular I would like to mention Luca that encouraged me to start my research activity and engaged me to work in PLATFORM project, Monica for her contagious cheerfulness, and Ivo, Giovanni, Andrea, Doug, and Braham for having read parts of this thesis and of my papers.

I would like to thank my thesis advisor Marco who has hardly worked with me in this years teaching me many important things, among them the most important one is the love for teaching. I would like to thank all USI Ph.D. students who shared with me these years of hard work and all professors, assistants, and secretaries at USI who have collaborated with me in these years.

Last but not least I would like to express my gratitude to my parents, Sergio and Graziella, to my husband Andrea, to my sister Patrizia and her husband Fabio, to my nephew Daniele and my little niece Elisa, to my grandmother Caterina, to Andrea's family, to Erika a special friend who gave me hospitality during this years, and to all friends and relatives who made these years full of love and of happiness.

Contents

Chapter 1

Introduction

The research topic of this thesis concern artificial agents and their interaction and communication within multiple agent systems. Given that studies about artificial agents are quite recent, in Artificial Intelligence research literature there is not yet a generally accepted definition of *intelligent agent* mainly because some of its distinguishing attributes may result less or more important on the basis of the different domains of application of agents. Taking inspiration from research literature on artificial agents I think that it is possible to describe an *agent* as a computer program that operates continuously in a specific environment for performing a specific task and having the following crucial characteristics: *autonomy*, *rationality*, and *interoperability*. In particular regarding to interoperability it is important to remark that like the capability of communicating with other agents using a language has had a crucial role in the evolution of human creatures in the same way it will play a crucial role in the evolution of the capabilities of artificial agents. Such a social ability is much more complex than the ability to exchange binary information but it is the capability to exchange messages in an expressive *Agent Communication Language* (ACL).

Multiagent Systems (MAS) and more recently *Agent Societies* are research areas within *Distributed Artificial Intelligence* (DAI) that study systems consisting of multiple interacting agents. There are many reasons and benefits for studying and developing systems with multiple agents. A first reason is to study distributed approaches to certain type of problems. Another reason is that there are important advantages in developing systems composed by multiple self-interested autonomous agents acting as "individuals", which very often represent real world parties, rather than as "parts" of a whole system. The opportunity for artificial agents of efficiently retrieving, filtering and exchanging information, as well as exchanging knowledge, services, products and expertise with other agents, or even with humans, enables them to solve problems that cannot be solved alone. Furthermore the need to realize this type of applications, where artificial agents can interact and negotiate, has become more and more important in the last few years thanks to the advent of Internet. In particular electronic commerce applications like for instance on-line auctions or automated negotiations are becoming more and more popular on the Web. In fact given the intrinsic complexity of evaluating which product is the best to purchase among all possible products available on various web sites and the complexity to compare their prices, their characteristics and to monitor the prices changing especially when an agent want to buy bundle of inter-independent products, the possibility to engage intelligent software agents to better perform such tasks on our behalf is crucial for the success of such applications.

1.1 Goal of the Thesis

This thesis aims at describing and investigating two fundamental subject of research within the Multiagent Systems paradigm: the problem of design *decision making models* for trading agents that operate in electronic commerce applications and the more general problem of defining an application-independent method to formalize *open interaction frameworks* where as a specific case electronic commerce interactions can actually take place. In fact the need to realize open, heterogeneous, and dynamic interaction systems has became more and more crucial with the advent of Internet. In such systems multiple agents can enter and leave dynamically, no assumptions are made on the internal structure of the interacting agents, and are defined using a method that enable agents designer to develop a single artificial agent that can interact with different systems designed by different organizations as sketched in Figure 1.1.

First of all I aim to give a schematic description of the notion of agents and of existing studies within Multiagent Systems research. Then I will analyze in detail two fundamental problems: the problem of designing agents for automated negotiation and the problem of enable them to negotiate by means of a suitable open interaction system. In particular, with respect to this second point, I aim to study, through also the analysis of existing approaches, the problem of defining a standard semantics for Agent Communication Languages. In fact, drawing inspiration by philosophical studies on human communication [143] the *institution of language* is the fundamental component of every interaction framework.

Departing from the identification of the crucial characteristics that an Agent Communication Language for open interaction systems has to abide by I aim to propose an operational definition of an Agent Communication Language based on the notion of social commitment. Finally my purpose is to test the proposed framework to define an interaction system for electronic commerce interactions, in particular to define using the proposed framework one of the most popular interaction protocol: the English Auction protocol.

1.2 Current Open Problems and Principal Contributions

I started my studies on trading agents and multiagent interaction systems by taking part to an international on-line competition: the Trading Agent Competition (TAC) game. The most interesting aspect of that competition is the complexity of devising a successful policy to buy bundle of complementary and substitutable products in parallel auctions of different types. This part of the work leads to two main contributions: one is related to the trading agent itself, that is, about its performance, its limits and its advantages, the other,

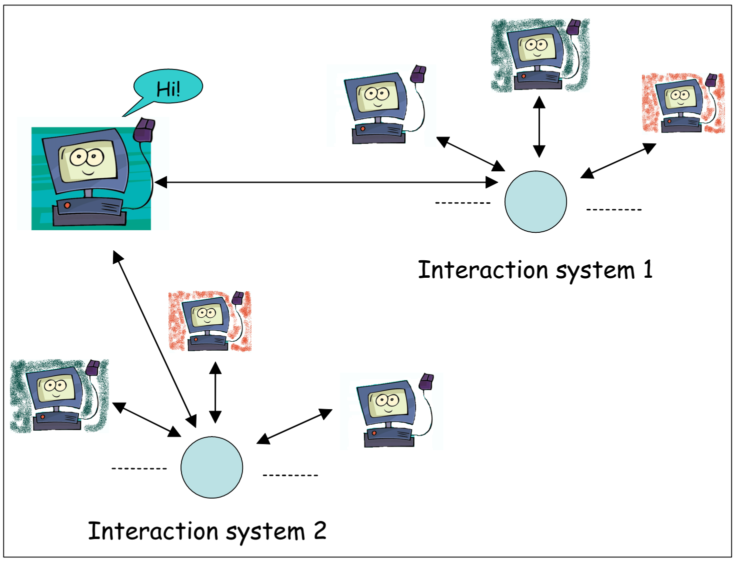

Figure 1.1: Heterogeneous artificial agents interact by means of open interaction systems.

that is more general, is related to the interaction system available for the competition.

In fact during the design and development phase of the trading agent some crucial and critical troubles related to the TAC interaction system emerged. First the problem of accurately understanding the rules that govern the different auctions present in the game, and second the problem of understanding the meaning of the numerous messages that the registered trading agents can use to interact with the system. Another more general problem that became clear during the design phase of the trading agent is that its internal structure would have been strongly determined by the peculiar interface of the relevant interaction system, in fact the agent has to use a set of pre-defined methods to interact with the TAC server, consequently without any changes in its algorithms, it would not be able to take part to any other competition, even with slightly different rules or to communicate with any other interaction system present on Web. Furthermore the agent would not have been able to exploit opportunities, to handle unexpected situations, or to reason about the rules of the various auctions, since it is not able to understand the meaning o the exchanged messages. The presence of all those problems bears out the need to find a standard common accepted way to define open interaction systems.

An open interaction systems has to enable communicative interactions among self-interested agents in open, heterogeneous, distributed, and dynamic systems. That is to realize interaction systems where multiple agents distributed around the world can enter and leave dynamically, where no assumptions are made on the internal structure of the interacting agents, and that allow to the same agent to interact with different systems designed by different organizations.

The most important component of every interaction framework, as is remarked also by philosophical studies on human communication presented in Speech Acts Theory, is the *institution of language*. Following these prospective I start to investigate the problem of defining a standard and common accepted semantics for Agent Communication Languages. That problem has received much attention in recent years but the solutions proposed so far are at best partial, and are considered as unsatisfactory by a large number of specialists. In particular, most current proposals are unable to support verifiable compliance to standards and to make agents responsible for their communicative actions. Furthermore those proposals make the strong assumption that every interacting agent may be at least modelled as a Belief-Desire-Intention (BDI) agent.

What is required is an approach focused on externally observable events as opposed to the unobservable internal states of agents. Focusing on external events means to take into account the "social framework" within which agents interact. Taking inspiration from Speech Act Theory [6, 140] an operational specification for the definition of a standard Agent Communication Language based on the notion of *social commitment* will be presented. In that proposal the meaning of basic communicative acts is defined as the effect that sending the message has on the social relationship between the sender and the receiver of the message described through operation on an unambiguous, objective, and public "object": the commitment. The adoption of the notion of commitment as an external object is crucial to stabilize the interaction among agents, to create an expectation on other agents behavior, and to enable agents to reason about their and other agents actions. Moreover given that this approach is inspired by speech acts studies it makes possible to treat human and artificial agent communication in a uniform way, a crucial aspect to obtain successful mixed interactions. Those basic communicative acts form a basic Library that can be used to express the meaning of the exchanged messages of different interaction systems.

In particular the proposed Agent Communication Language has some very important characteristics: it is verifiable, that is, it should be possible to determine if an agent is behaving in accordance to its communicative actions, the semantics is public, that is, any third part agent witness of the messages flow has to be able to draw similar inferences from the interaction, and objective in order that everybody attributes the same meaning to the exchanged messages. The proposed semantic is external with respect to the agent's internal structure, flexible and extensible to let agent to cope with various and new situations, simple to be correctly used by agent designers, yet enough expressive. Finally respect the autonomy of agents, they must have only social constrains on their behavior.

A complete formal operative specification of an interaction framework able to support interactions among artificial agents using the proposed commitment-based Agent Communication Language is presented. In particular some sample applications of how to use the proposed framework to actually formalize *interaction protocols* are reported. The protocols analyzed are widely used in electronic commerce applications and they are: the protocol of proposals and the protocol of offer. Furthermore a list of *soundness conditions* to test if a protocol or a general interaction is sound are proposed. Such conditions express constrains on the content of the state of the interaction system at various stage of the conversation with regard to the meaning of the exchanged messages.

To complete this research work an application-independent method to formalize interaction protocols is presented. It consist of three components that are: the definition of the meaning of every communicative act using the ACL proposed, the definition of preconditions for the performance of communicative acts specific to the interaction protocol analyzed, and finally the *interaction diagram* of the protocol. This method will be used to successfully formalize a more complex interaction protocol, the English Auction protocol used in the Trading Agent Competition game discussed in Chapter 4. The achievement of this result demonstrates that the proposed semantics can be successfully used to formalize interaction protocols like for example electronic auctions that are widely employed in electronic commerce applications.

1.3 Outline of the Thesis

This thesis is organized as follows. In Chapter 2 the notion of agent is introduced focusing in particular on what are its constituent characteristics, and on principal functional architectures. In Chapter 3 existing studies and main topics of research about systems with multiple interacting agents are presented, particular relevance has been given to the methods of coordination among multiple agents. Chapter 4 faces the problem of automated negotiation going into detail in the design and test of a trading agent who took part to the international Trading Agent Competition. In Chapter 5 I analyze in detail the problems related with the definition and use of a particular method of coordination among agents: Agent Communication Languages. In Chapter 6 a new proposal for an operational definition of the semantics of agent communication languages based on the notion of social commitments is presented and discussed. In Chapter 7 an application independent method for the definition of interaction protocol for open interaction systems based on the proposed ACL is presented and a concrete example of formalization of the English Auction Protocol, a protocol used in the TAC game, using the model proposed in the thesis is presented. Finally Chapter 8 concludes this thesis with comments on what have been done, and outlines interesting directions for future works.

Chapter 2

Artificial Agents

This chapter introduces the notion of an ”agent”, very important in Artificial Intelligence (AI). In fact it is possible to describe Artificial Intelligence as the subfield of Computer Science that aims to construct agents that exhibit an intelligent behaviour. This chapter is organized as follows. Section 2.1 describes what an artificial agent is, how it is possible to distinguish it from a simple program using its distinctive features and how it relates to its environment. In Section 2.2 I present a schematization of agents' functional architectures distinguishing between two broad categories: reactive agents, which react immediately to the changes in the environment's states, and deliberative agents, which reason about the expected effects of their actions on the environment. In Section 2.3 particular relevance to the Belief-Desire-Intention (BDI) model of agency, a very important model in agent communication languages studies, is given. Finally in Section 2.4 the conclusions of this chapter are given. For a more detailed introduction about artificial agents the book ”Artificial intelligence, a modern approach” by Russel and Norvig [131] is recommended.

2.1 Definition and Main Characteristics

An artificial agent is usually a computer program that was devised to obtain an entity equivalent or similar to a human being or animal that truly exists in the world. The main characteristics of human or animal beings are that they are able to live in an unpredictable environment, to act and in general to interact with that environment, with the aim to reach some goals, first of all to survive.

Recently there has been a growing interest in artificial agents in different subjects, for example in data communications and concurrent systems, in robotics, in user interface design, in electronic commerce and in information retrieval. Artificial agents are finding a wide range of applications. For example in 1998, Deep Space 1 was the first space probe to have an autonomous agent-based control system [111]. Another example is the increasing use of agents in Internet-based electronic commerce, where agents autonomously buy and sell goods on behalf of a user [115, 77]. The trading agent Nidsia, described in Chapter 4 is an example of software agent for electronic commerce [63].

As Russel and Norvig state in their book [131], there is not yet a generally accepted definition of intelligent software agent in Artificial Intelligence, mainly because some attributes are less or more important based on the different domains of application of intelligent agents. Literature mostly gives only intuitive definitions describing some important characteristics that various artificial agents must have. Below are some of these definitions proposed by the main researchers in this field, which reflect the intuitive one expressed above.

- "An agent is anything that can be viewed as *perceiving* its environment through sensors and *acting* upon that environment through effectors", Russel and Norvig [131, p.31].
- "An agent is a computer system that is *situated* in some *environment*, and that is capable of *autonomous actions* in the environment in order to meet its design *objectives*", Wooldridge [177, p.29].
- An agent is a computational entity that can be viewed as perceiving and acting upon its environment, that is *autonomous* and that operates *flexibly* and *rationally* in a variety of environmental circumstances, Weiss [172, p.1].
- "An agent is a persistent computation that can perceive its environment and reason and act both alone and with other agents. The key concepts in this definition are *interoperability* and *autonomy*", Singh [151, p.40].
- Artificial agents are computer programs operating continuously and autonomously in a specific environment in order to carry out a predefined task, Colombetti [31].

It is interesting to notice that there are some important key concepts in the above definitions: *autonomy*, *rationality*, and *interoperability*. Each one contributes to determine what an agent is and they will be discussed in more detail below. Other attributes may be considered important depending on the application of the agent, an example may be *mobility*, i.e. the ability of an agent to move around an electronic network.

An agent's behaviour is determined by its *program*, that is the mapping from the current percept and the current *state*, which represent the previous history of the system, to actions. The behaviour also depends on the agent's *physical architecture*, that is the computing devise where the program will run. The architecture makes the percepts available to the program and transfers the program's actions to the effectors. Various *functional architectures* of agent's programs will be discussed in Section 2.2.

The various possible types of *environments* are also important when investigating artificial agents. Russel and Norvig suggest the following classification of environment properties [131]. An environment may be *accessible* if the agent's sensors give it access to the complete state of the environment or at least to all the aspects that are relevant to the choice of the actions, otherwise the environment is *inaccessible*. An environment is *deterministic* if its next state is completely determined by the current state and the actions selected by the agents. If the environment is inaccessible it may appear *nondeterministic*. If the environment can change while an agent is deliberating, it is called *dynamic*, otherwise it is *static*. Finally if there are a limited number of distinct, clearly defined percepts and actions the environment is called *discrete* in the space of the states, like for example chess, otherwise it is called *continuous*.

A generic agent determined by its program and by its physical architecture, which perceives and acts on the environment is schematized in Figure 2.1.

2.1.1 Autonomy

The term *autonomy* or *autonomous* may refer to, at least, three different important aspects of artificial agents.

In its most common acceptation, the adjective autonomous is used to mean that, to some extent, agents have control over their behaviour. This means that their actions are determined by their own experience and that they are able to act without the intervention of humans or other systems. Thus an autonomous agent makes independent decisions that are under its own control and that are not driven by others. I think it is possible, metaphorically, to speak about it as "the freedom of the will" of agents with respect to their system designers. This type of autonomy is necessary in agent's theory to make possible not to tell an artificial agent *how* to do something but tell it only *what* to do.

Depending on its functional architecture (see Section 2.2) an artificial agent may exhibit different levels of autonomy with respect to its designer.

- Reactive agents are very simple agents. They consist only of a program that maps each possible perception or percept sequence in the corresponding action that the agent has to carry out. They need a *built-in knowledge*, which univocally determines their behavior. Actually these agents lack autonomy and their big limitation is that they are not flexible, in fact they operate successfully only if the assumptions about the environment, made by the designer during the project phase, will hold.

- Planning agents are more sophisticated. They have a more complex built-in knowledge about the set of actions that can be performed. This means that they know

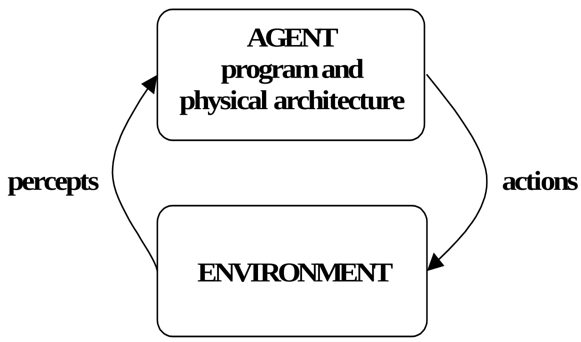

Figure 2.1: An agent interacts with its environment

the preconditions and the effects of their actions on the environment. They have also some knowledge about the mechanisms that govern the dynamic evolution of the environment. This kind of agent seems more autonomous than the previous one, but actually it can "only" choose which plan to execute among all the combinatorial combinations of their allowed actions. Even if this number can be very high, it is still a finite number and formally the agent cannot be considered truly autonomous.

- A truly autonomous artificial agent may be obtained providing it with the built-in knowledge previously described and with the powerful capability to *learn*. In this way its behavior is actually determined by its own experiences. This kind of agent may learn for example new preconditions and effects of its actions, how much is the reward of each action and so on. Examples of a very successful learning technique are Reinforcement Learning and Neural Networks. They can be used by artificial agents to build and continuously update their own model of the problem that they have to face. It is important that some of such learning techniques have proved to converge, it is the case for example of Q-learning [171] one of the simplest reinforcement learning algorithms.

However, the term autonomy has usually a slightly stronger meaning, it refers also to the quality of an agent to have its own goals and its own freedom of will with respect to other agents requests. In this case the term autonomy reflects the idea of social freedom in human beings society. This type of autonomy is very important in multiagent system applications, where the various agents are self-interested like for example in electronic commerce. But when artificial agents form a society or reflect the human one, their autonomy has to be limited because it presents some drawbacks. In fact for agents it is important to have the possibility to negotiate with others to achieve more complicated goals, but in order that the social system keeps on working, it is very important that agents honor their commitments. A *normative* system is then necessary. There are also situations where agents decide to sacrifice their autonomy to collaborate or negotiate with other agents. One example is when the common goal of the agents is to achieve a global optimum, for instance maximize the "social welfare" or to find a Pareto optimal solution (I will discuss these evaluation criteria of automated negotiation protocols in Chapter 4). Another possible situation is when in competitive applications everybody may be better off if each negotiates an agreement with the others. For example in the Prisoner's Dilemma game, described in Table 2.1, for both agents it is better to cooperate than to defect.

In literature it is possible to find also the term *design autonomy*. It refers to the autonomy of artificial agents designers to develop an agent independently from other agents' designer or super parties' directives. Design autonomy minimizes requirements on agent internal structure and on agent behaviour, thus promoting heterogeneity. Recently [151] this type of autonomy became very important thanks to the advent of truly open interaction systems like the Internet. I will discuss its importance in the definition of a commonly accepted agent communication language in Chapter 5.

Table 2.1: Prisoner's Dilemma game.

players	cooperate	defect
cooperate	3,3	0,5
defect	5,0	1,1

2.1.2 Rationality

Here I will analyze rationality as an *individual* property of intelligent agents, although it is possible to study also *collective* rationality in cooperative multiagent systems or in teamwork treating it as a property of the whole system. An intuitive description of the notion of rational agents can be the following: a system whose actions make sense from the point of view of the information possessed and on the basis of the goals or tasks for which it was designed [130]. While the following formal definition of the concept of rational agent was proposed by Russel and Norvig in their book [131]: "For each possible percept sequence, an ideal rational agent should do whatever action is expected to maximize its performance measure, on the basis of the evidence provided by the percept sequence and whatever built-in knowledge the agent has". Speaking about the term rationality, I think it is important to report that Simon, one of the fathers of Artificial Intelligence, in 1958 coined the terms "substantive rationality" and "procedural rationality" to describe the crucial difference between the question of *what* decision to make and the question of *how* to make it [150].

Determining what is rational depends on the definition of the degree of success, on what and how the agent is able to perceive and has perceived so far, and on the actions that the agent can perform. In these notions many problematic questions are hidden: first of all the problem of understanding and expressing the relationship between performance measure and goals or between performance measure and utility. Moreover the question of where the knowledge of the agents comes from related with the problem of rational *learning*. Finally the question about the capability of an agent to *predict* the effects of its actions thanks to some *planning* capabilities.

An important component of rationality is *flexibility*. As Wooldridge and Jennings state in [180], an autonomous agent is flexible if it is able to act in order to meet its design objectives in an uncertain environment. Flexibility can then be seen as the capacity to balance *pro-activeness*, that is the capacity to exhibit a goal directed behaviour and *reactivity*, that is the capacity to respond to events that occur in the environment where these events effect the agent's goal or the agent's assumptions about it.

Generally speaking it is also important to distinguish between rationality and omniscience. An omniscient agent knows the actual outcome of its actions and can act accordingly. But omniscience is impossible in reality because the environment is usually uncertain mainly because it has an inherent partially unknown dynamic or because it is populated with other agents which can continuously change it with their actions.

In Artificial intelligence there are roughly two main approaches to build the decision-making process by which artificial agents select their actions: the symbolic and the economic one.

- The *symbolic* approach focuses on a model of rational decision making as practical reasoning of the kind that humans engage everyday. In this model agents have to realize some desires that represents their goals. This choice follows the cognitive approach rooted in psychological works, which treat agents as entity with beliefs, desires and intentions (see Section 2.3). In this approach, intelligent reasoning is obtained using logic techniques, such as formal calculation, typically deduction.

- Following the *economic* approach an agent has to maximize a utility function. This view borrows from economic metaphors the idea of intelligent reasoning as the adherence to the tenets of utility theory. Combining utility theory with probability theory, that is used to manage uncertainty and partial information knowledge of the environment, von Neumann and Morgenster in 1944 [168] founded *Decision Theory*. In Decision Theory a rational agent is the one that chooses an action which maximizes expected utility, where expected utility is defined in terms of the actions available to the agent, the probability of certain outcomes, and the preferences the agent has with respect to these outcomes. In multiagent scenarios, where an agent has to interact with other agents, *game theory* is also a powerful predictive and analytic tool. Whereas an approach to solving sequential decision problems, where the agent's utility depends on a sequence of decisions, is *dynamic programming* developed by Bellman in the late 1950s [10]. The agent for electronic commerce, Nidsia, that I developed to take part in an international trading agent competition [63](see Chapter 4) is a utility based agent that chooses its action using decision theory techniques.

Following Simon's studies [150] I think it is important to point out that perfect rational agents do 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 suboptimal. In this case the term used to indicate the rationality of real system is "*bounded rationality*".

2.1.3 Interoperability

The ability to interact and in particular the capacity to communicate using a language plays a crucial role in the evolution of the human creatures. The same capability, *interoperability*, promises to play the same decisive role for artificial agents.

In fact, as I will discuss in more detail in Chapter 5, the opportunity for distributed and heterogeneous agents to exchange information, competence, services products, etc. and the ability to negotiate, cooperate, or compete with other agents and perhaps with human, makes them able to solve problems that cannot be solved alone and that intrinsically require the interaction with other entities. It is important to underline that this social ability is much more complex than simply the ability to exchange binary information.

Genesereth and Ketchpel have gone so far to equate agency with the ability of a system to exchange knowledge using an agent communication language: "software agents, i.e. software "components" that communicate with their peers by exchanging messages in an expressive *agent communication language*" [71, p.48].

Interactions among artificial agents can take place *indirectly* through the environment in which they are embedded (for example by observing one another or by carrying out actions that modify the environment) or *directly* through a shared language. The importance of using a common shared language is in its expressiveness. As I will broadly discuss in Chapter 5 about *agent communication languages* (ACLs), an ACL allows the exchange of complex information. Like for example goals, requests to carry out actions in a declarative form, commitment to perform some other actions. In general an ACL lets agents to have articulated *conversations*, that is to exchange task-oriented sequences of messages, for example to take part in a negotiation or an auction. Moreover if a certain artificial language will become a "standard" agents will become able to interact with many different applications, for example to take part in different parallel auctions present on the Internet.

2.2 Functional Architectures of Intelligent Agents

A first important distinction in agents' studies is between *robots*, which usually behave in "physical" environments, and software robots, which are called *softbots* and usually behave in "virtual" environments. An artificial environment usually may be complex, dynamic and non-deterministic like a real one; in fact it is usually a simulation of its real counterpart or has important connections with it. For example Nidsia the trading agent for electronic commerce, that will be presented in Chapter 4 is a softbot, it operates in an artificial environment constituted mainly by other trading agents but may, in principle, sell and buy real goods in place of its owner.

As is reported above, an artificial agent is essentially a permanent program that accepts percepts from the environment, makes decisions and generates actions. On the basis of an agent's decision-making process it is possible to delineate the following agents' typologies.

2.2.1 Reactive Agents

This type of agent perceives the state of the environment and decides immediately on the corresponding action such as for example those based on Brooks's Subsumption Architecture [19].

Look up agents

The simplest possible agent is a look-up agent. This name comes from its internal look-up table that is used by the agent to map every possible percept sequence into the appropriate action. Such agent uses its memory to keep track of the entire percept sequence. But this type of agent is doomed to failure because the table with all possible percept sequences where look up the actions becomes quickly bigger and bigger and then intractable. This type of agent has not autonomy because the calculation of the action is entirely built-in in the look-up table.

Reflex agents

A very basic agent that can be implemented very efficiently is the one that simply follows condition-action rules. If the agent perceives a certain state, then acts in a definite way. The action a to be performed at time $t+1$ is computed as a function of the percept s at time t:

$$a(t+1) = f(s(t)).$$

This type of agent decides what to do without reference to the percept sequences. Also human beings have many of such reactive rules, some of them learned, like for example the rules for driving, and some of them innate, such as blinking when something approaches the eye. This type of agent has no autonomy at all, because the choice of its actions is entirely built-in, so if the environment will change in an unexpected way it would be lost. A solution to this problem may be, in certain situation, to equip the agent with learning capabilities.

Agents with an internal state

The simple reflex agent described before will work only if the correct decision can be made on the basis of the current percept. Otherwise it has to maintain some sort of internal state in order to choose the right action. The action a to be performed at time $t+1$ is computed as a function of the percept s at time t and of the current internal state $x(t)$:

$$\begin{aligned} a(t+1) &= f(x(t), s(t)) \\ x(t+1) &= g(x(t), s(t)). \end{aligned}$$

The internal state can be used to keep track of the percept sequence but also to have an internal description of the current state of the environment in the cases when it is not completely accessible. This internal description can be computed by the agent using knowledge about how the world evolves and about the effects of its own actions. It is important to distinguish between the "physical" internal state of an agent that can be perceived also by a reflex agent, for example in a robot its level of energy in the battery and the "mental" state of the agents that is used to intend the world and is like a human mental state, for instance an agent may believe that there is something inside a certain box.

2.2.2 Deliberative Agents

In many situations the action to be performed by an artificial agent has to be computed not only on the basis of the state of the environment but also on the basis of its expected effects on it, that is, the agent reasons about its actions. For example an agent that plays chess cannot decide its action only on the basis of the current position of the chess-pieces but it has also to evaluate the future effects of its moves. This type of agent needs to have a model of the dynamics of the environment and of the effects of its actions on it. Deliberative agents may appear less efficient than reactive agents because they have to reason about the action to perform but they are far more autonomous and flexible than reactive agents. One important class of deliberative agents is the Belief-Desire-Intention (BDI) model of agents treated in detail in Section 2.3.

Goal-directed agents

Goal-based agents have some sort of *goal* information, which describes situations that are desirable. This type of agent can then combine the information about the goal and about the effects of its action to choose the action to perform. This choice can be simple when goal satisfaction results immediately from a single action. But usually it will be more trickier, when the agent has to consider long sequences of actions and their effects on the environment evolution to achieve the goal. The agent has then to use some *search* or *planning* capabilities to find action sequences that do achieve the agent's goal. A detailed description of *problem solving by searching* techniques can be found in [131, p.55]. A discussion about planning and plans can be found in [131, p.337] and a detailed survey can be found in [4]. Also *learning* capabilities, are very important for deliberative agents in order that they can use their percepts not only for acting, but also for improving their ability to act in the future, a survey on this topic can be found in [131, p.525].

Utility-based agents

Utility is a function that maps a state or sequence of states onto a real number. It allows a comparison among different environment states while the notion of goal just provides a description of desirable states. A complete specification of the utility function allows rational decision in two cases where having only the notion of goal is not enough. One is when there are conflicting goals, only some of which can be achieved. Another is when there are several goals that the agent can aim for, but none of which can be achieved with certainty; utility provides a way in which the likelihood of success can be weighted up against the importance of the goals.

2.3 The Belief-Desire-Intention Model of Agency

In this section, following Wooldridge approach presented in [178], the BDI model of agency is introduced. It is the result of the combination of three distinct components: the philosophical, the logical, and the software architecture. Below I discuss the BDI model of agency with some details because of its importance in the field of Multiagent Systems.

First the idea of realizing BDI agents is so strong in AI that Shoham, in an article about Agent Oriented Programming [146], defines an agent as to be an entity that recognizes and deals with the outside world as having mental qualities such as *beliefs*, *intentions* and *desires*.

Second the *intentional stance*, that is, the strategy of interpreting the behavior of an entity (person, animal, or artifact) by treating it as if it were a rational agent that governed its choice of actions by a consideration of its beliefs and desires, is a useful abstraction tool used in computer science to describe the behavior of very complex systems. Its distinctive features can be best seen by contrasting it with another basic method of prediction the *physical stance*, which proposes to use the laws of physics and the physical constitution of the things in question to devise our prediction. The intentional stance was proposed by the philosopher Daniel Dennett [43, 44] and it was first applied to computational systems by the computer scientist John McCarthy in 1979 [109].

Third mentalistic models are good candidates for representing information about end users, for example for a personal assistant, a crucial point to enhance interactions among human beings and software agents.

Finally as I will fully discuss in Chapter 5, several researchers have proposed to use cognitive concepts as semantic basis for agent communication languages. But intentional concepts are not well suited to give basis to a public, standardized view of communication [151].

Philosophical component

The *belief-desire-intention* (BDI) model of rational agency is based on a widely respected philosophical theory of human rational actions, developed by the philosopher Michael Bratman [17] within the tradition of analytical philosophy. It is a theory of *practical reasoning*, i.e. the process of reasoning that we all go trough in our everyday lives, deciding moment by moment which action to perform next. Human practical reasoning is mainly characterized by a process for deciding *what* to achieve and by a subsequent process of deciding *how* to achieve these state of affairs. The former process is known al *deliberation* and the latter as *means-ends reasoning*.

This model focuses in particular on *future-directed intentions*, i.e. desires that has to be achieved to which human beings are committed with themselves. Intentions are important because they allow us to not waste time considering possible actions that are incompatible with our intentions. Since any software agent that we might implement must have resource bounds, this model seems attractive. It is important to note that intentions are persistent, a human being does not give up an intention without a good reason. Furthermore intentions interact with an agent's beliefs and other mental states. For example that an agent has the intention to achieve a certain state of affairs φ, implies that it believes φ is possible and it believes that given the right circumstances, φ will be achieved. Formally capturing the interaction between intention and belief is very hard.

Logical component

A complicated question in BDI systems is finding a method to axiomatize general properties of BDI agents. A formalization of some aspects of Bratman's theory, using modal logic, was made by Cohen and Levesque in 1990 [29]. In the meantime Rao and Georgeff developed the BDI logic in order to give an abstract, idealized semantics to the BDI agents they were building throughout the early 1990s at the Australian AI Institute [124, 72]. They present alternative possible-worlds formalism for BDI-architectures with three crucial elements. First, intentions are treated on a par with belief and desires. This allows defining different strategies of commitment to intentions as described above. Second, they distinguish between the choice an agent has over the actions it can perform and the possibilities of different outcomes of an action. Third they specify an interrelationship between beliefs, desires, and intentions [124, p.1].

Further recent studies proposed by Wooldridge to improve such BDI logic can be found in [178]. In this book a new logic $\mathcal{LORA}$ is presented; this logic in addition contains a temporal component that allows one to represent the dynamic of agents and their environments, and an action component that allows one to represent the actions that agents perform and their effects.

Software architecture component

Intuitively *beliefs* correspond to what an agent imagines its world state to be and these beliefs may be incomplete and incorrect. Agent's *desires* represent states of affairs that the agent would wish to be brought about. *Intentions* represent desires that it has committed to achieving.

As is mentioned in the previous section, the main components of the software architecture of a BDI agent are the deliberation phase about what intention to achieve next, and the means-end reasoning to get a plan for achieving the intention. These two processes have a time cost associated with them. Consequently, each one produces an output at time t_1 on the basis of assumptions about the world at time t_0. If the world doesn't remain static during the interval of time $t_1 - t_0$, like in realistic environments, the result of the computations may be obsolete.

The deliberation process has two distinct functional components. First the *option generation*, in which the agent generates a set of possible alternatives. It takes the agent's beliefs and current intentions and determines a set of desires. Secondly the *filtering* component that chooses one desire and commits itself to achieve it [178]. After that the means-end reasoning process creates a plan to achieve the intention. Usually it does not start from scratch [4] but its work consists of finding the correct plan in an existing plan library [73], on the basis of the pre- and post-conditions of the listed plans.

Analyzing the aspect of creating a commitment to intentions, a problem arises: how long should an agent be committed to its intentions? There are mainly three commitment strategies which are commonly discussed in the literature about artificial agents [124]: *blind commitment*, when an agent continues to maintain an intention until it believes the intention has actually been achieved; *single-commitment*, when an agent continues to maintain an intention until it believes that either the intention has been achieved, or else that is no longer possible to achieve the intention; *open-minded commitment*, when an agent maintains an intention as long as it is still believed possible. On the basis of the chosen commitment strategy the agent has to reconsider its intentions more or less often during its reasoning process.

Implementations of the BDI model

The Procedural Reasoning System (PRS) [74] was one of the first implemented systems to be based on a BDI architectures. It was implemented in LISP and has been used for a wide range of applications in problem diagnosis for example for the Space Shuttle [86], air-traffic management [103], and network management [86].

dMARS is a faster, more robust reimplementation of PRS in C++. It has been used in a variety of operational environments, like for example paint shop scheduling in car manufacturing, air combat simulation, resource exploration, malfunction handling on NASA's Space Shuttle [97].

COSY is another BDI architecture, with many similarities to the previous [82]. In addition it has given importance to both psychological and social commitments. COSY has a strong component of cooperation based on formal protocols built on top of an agent communication language. Such protocols involve commitments among the agents, and include rules through which tasks may be delegated to and adopted by different agents.

Breiter and Sadek implemented a formal theory of beliefs and intentions in their ARITIMIS system [18]. This system carries out intelligent dialogue in assisting the user in tasks such as information access. This system applies Grice's maxims [78] whereby the computer attempts to infer the user's intentions and act accordingly.

DEPNET is an interpreter for agents who can perform social reasoning [148]. Agents in this system represent knowledge about one another to determine their relative autonomy or dependence for various goals. Dependence leads to joint plans for achieving the intended goals. The underlying theory is based on dependence rather than social commitments. This tool shows how social notions can be realized in tools for simulating and analyzing multiagent systems.

2.4 Conclusions

In this chapter the notion of an agent that will be used in all the remaining parts of this dissertation has been introduced. In the following chapter systems with multiple autonomous agents will be introduced and studied and particular focus will be posed on their coordination mechanisms. Two of such coordination mechanisms will be then analyzed in detail: fixed interaction protocols in Chapter 4 and agent communication languages in Chapter 5. In Chapter 4 a deliberative agent for electronic commerce will be described and studied. In Chapter 5 a proposal for a new agent communication language based on the notion of commitments is presented and compared with current existing proposals which are based on the BDI model of agency.

Chapter 3

Multiagent Systems and Agent Societies

The software agent paradigm presented in the previous chapter is becoming more and more important thanks to its crucial contribution in the realization of innovative and complex applications consisting of multiple interacting agents. Indeed, nowadays, due to the increasing of interconnections and networking of computers, and especially the growth of Internet, the situations where an agent can operate usefully by itself are getting rare, whereas those in which agents operate and interact in environments inhabited by other agents are becoming more and more common. Departing from these observations the focus of this chapter is on systems consisting of two or more homogeneous or heterogeneous agents, which are able to interact with each other and to act in the environment.

First of all, this chapter describes the main historical and ongoing advantages of developing multiple agents applications. On the basis of the main questions posed in the investigation of systems with multiple agents, and on the consequent research topics arisen, I will try to outline a schematization of various existing studies on systems with multiple agents; identifying, in the end, mainly three areas of research: *Distributed Problem Solving*, *Multiagent Systems*, and *Agent Societies* as depicted in Figure 3.1. In this chapter I will try to describe them despite in the dedicated literature a common agreement on both the meaning and use of the various terms and on the outline of each sphere of research has not been reached yet.

Finally, various methods of *coordination* of the agents' actions will be analyzed. Coordination mechanisms are necessary so that some structured interactions may take place. In particular I will distinguish between *cooperation* that is coordination among non-antagonistic agents and *competition* that is coordination among competitive or simply self-interested agents. Among the various possible methods of interaction, particular relevance will be given to *interaction protocols* and *communication languages* and to application *dependent* and application *independent* ways of interaction necessary to realize truly open systems. Interaction protocols enable agents to have structured exchanges of messages and are largely used for automated negotiation, for example in electronic commerce applications, as I will argue in Chapter 4. Communication languages enable agents

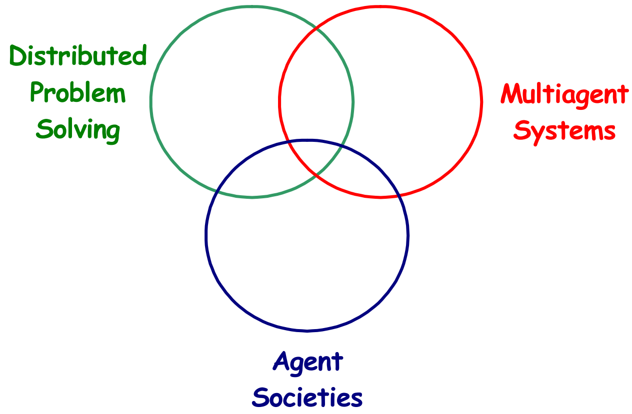

Figure 3.1: Different research approaches for studying systems with multiple agents

to exchange and understand messages and I will discuss this important topic of research and present a new proposal for the definition of the semantics of an agent communication language in Chapter 5.

3.1 Systems with Multiple Agents

Research studies concerning multiple agent systems are quite recent and in frenetic evolution. Distributed Artificial Intelligence (DAI), born in the eighties, was the first research area within Artificial Intelligence to be concerned with systems of multiple agents. DAI was defined by Weiß in 1996 as the study and design of systems consisting of several interacting entities which are logically and often spatially distributed and in some sense can be called autonomous and intelligent [173]. More recently other parallel research areas such as Multiagent Systems (MAS) and Agent Societies have been arisen.

The content of this chapter is mainly inspired by various online discussions, by several research papers, and by some of the most famous books on this subject. Important collections of articles about Distributed Artificial Intelligence were issued at the end of the eighties, and are: [83, 13, 68], whereas a more recent contribution can be found in the book of O'Hare and Jennings [116]. Noticeable works in the area of Multiagent Systems are: "Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence" edited by Gerhard Weiß (1999) [172], "Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence" by Jaques Ferber (1999) [55], "Understanding Agent Systems" by Mark d'Inverno and Michael Luck [47], and the most recent book "Introduction to Multiagent Systems" by Michael Wooldridge (2002) [179].

There are many reasons and benefits for studying and developing systems with multiple agents. At first the reason was to study distributed approaches to certain type of problems. In fact even though centralized solutions are generally more efficient, distributed computations are sometime easier to understand and to develop. This is crucial especially when the problem to be solved is itself distributed, for instance when data belong to independent organizations which want to keep their information private and safe for commercial reasons. Moreover the distributed approach ascribes important software engineering qualities to a system, like efficiency, in fact in some situations the distributed approach speeds up the problem solving thanks to the parallel use of resources, reliability and robustness which are both enhanced given that a system gets fault tolerant through redundancy. Distribution also encourages the reuse of the various components and guarantees the scalability of the entire system. Indeed since such systems are inherently modular, it is easier adding new elements to them than adding new capabilities to a monolithic system.

Another interesting aspect of the distributed perspective is the possibility of studying new approaches to solve certain type of problems. In fact, distribution can lead to computational algorithms that might not have been discovered otherwise and that often are a more natural way of representing the problem. Existing type of problems which exploit the distributed approach include: vehicle routing among independent dispatch centers, manufacturing planning, digital libraries, multiagent information gathering on the Web, routing and bandwidth allocation in multi-provider multi-consumer computer networks, electronic commerce, and various type of scheduling like scheduling among multiple companies, meeting scheduling, scheduling of patient treatments across hospitals, classroom scheduling, etc., to name just a few.

There are important advantages in developing systems composed by multiple self-interested autonomous agents acting as "individuals", which very often represent real world parties, rather than as "parts" of a whole system. The opportunity for artificial agents of efficiently retrieving, filtering and exchanging information, as well as exchanging knowledge, services, products and expertise with other agents, or even with humans, enables them to solve problems that cannot be solved alone. The capability of negotiating, cooperating or competing with other agents, and of reasoning about their goals or acting so as to influence their behavior lets artificial agents be able to complete tasks which intrinsically require an interaction with other entities, like for instance in electronic commerce or information retrieval applications.

Other reasons for studying systems with multiple agents lie in their applicability to different research fields. For example the simulation of artificial societies in biology or in social science makes it possible to validate new theories. Moreover, systems with multiple interacting agents may also be useful to investigate the various aspects of intelligence. In fact, it has been proposed that the best way to develop intelligent machines at all, might be to start by creating "social machines" [38].

In the following sections I will try to outline what is intended, within artificial intelligence, with the terms Distributed Problem Solving (DPS), Multiagent Systems, and Societies of Agents. Durfee and Rosenschein proposed three different approaches to discriminate between DPS and MAS [50]. From my point of view the best among such approaches is the one that focuses on the differences between MAS's and DPS's research agenda, and in particular on why certain systems were made, what were the questions that the designers wanted to answer, and which are the general properties that they aimed to demonstrate. I think that such an approach is crucial because it is the only one that allows to distinguish between similar systems developed with different scopes. Furthermore it is useful for the design of new applications in order to be able to identify the most relevant technology to apply on the basis of the desired properties of the target system.

An example of different research agenda applied to the investigation of similar interaction systems can be found in studies about the Contract Net Protocol [153] (Section 4.1.1). Such protocol initially was born as a coordination protocol for cooperative agents in distributed systems, whereas Sandholm used it later [133] as interaction protocol for a system consisting of different self-interested interacting agents to study the social outcomes of such a system, a typical investigation in Multiagent Systems studies. Other examples are Sandholm's studies on various types of auctions, such as the Vickrey auction [134]. These studies focused on the advantages and limitations of the protocol when it is used with self-interested, insincere, manipulative agents. Whereas other studies on electronic commerce applications are much more focused on the decision model of trading agents [157], or on the social organization necessary to realize effectively open dynamic multiagent systems (see Section 3.1.3).

3.1.1 Distributed Problem Solving

Distributed Problem Solving (DPS) is a sub-field of Distributed Artificial Intelligence, it deals with how to get multiple agents to work together in order to solve problems that require a collective effort [49]. Following the DPS approach, a problem is first decomposed into several, not entirely independent, sub-problems, then these last are solved independently, and finally their solutions are synthesized so that to form the solution for the original problem.

Within the DPS approach some assumptions are usually made about the systems studied. First, it is assumed to have a *centralized designer*, that is a single person or a team, who imposes some fixed properties, such as goals or attitudes, to the various agents. The centralized designer in turn makes strong assumptions about the compatibility of different problem-solving entities. Usually it imposes an interaction protocol, which determines the possible actions that agents can take at different stages of the interaction. Obviously there is no clear rule for expressing to what detail a centralized designer should specify the agents' design in order to make it a DPS system [50].

A second assumption of DPS systems is that of having agents with *common goals*, even if it might happen that an agent by solving some local problems would take decisions which may be globally incoherent. It should be noticed in this regard that it is not clear at which level a goal should be common in order to make a system a DPS. As an example consider the case where agents meet in a market-place for an auction. They share the common goal of interacting however, they also have other opposing goals for the result of the auction: the auctioneer usually desires to sell a product at a price as high as possible whereas the competing bidders want to purchase it at a price as low as possible.

The third and final assumption that usually is made about agents in a DPS system is that they are supposed to be *noncompetitive*, even *benevolent*. For example, in the first applications of the Contract Net Protocol as an interaction protocol [153], agents are supposed to give an honest bid on a task, indicating how well they expect to perform it. At the same time the auctioneer is supposed to award honestly the task to the best bid. It should be pointed out that even with this assumption cooperation and coherent coordination are not ensured. There is in fact always the risk that local views of the problem can lead to non optimal allocation of the tasks with respect to the entire system.

As a general remark DPS deals mainly with the investigation of the external properties shown by a system, that is, how efficiently agents can coordinate their plans, how robust the system is to network failures, what impact message delays or losses have, and so forth [50]. These studies are crucial when a distributed approach to a problem becomes necessary. That means in those situations when the information composing the problem is geographically distributed, or when a system needs to be very robust to failure and is better to avoid a centralized control, or finally when agents can accomplish their tasks better by solving numerous small problems.

3.1.2 Multiagent Systems

Multiagent System (MAS) is a sub-field of Distributed Artificial Intelligence, a science which studies the overall properties of a system of multiple agents given that no assumptions are made on the internal properties of the various agents, which usually have their own goals and interests [50]. This change of assumptions is due to the fact that during the last few years we have witnessed the growth of standardized communication infrastructure such as Internet and within such an infrastructure agents belonging to different owners and having different goals need to interact, in order to solve very complex problems which requires services and capabilities of other agents present on the net. Moreover also experiences borrowed by social sciences, game theory, and organization and management sciences push the research, about system with multiple agents, in the direction where the only possible assumption is that interacting agents want to maximize their benefit, i.e. their payoff.

The main issue in Multiagent System's studies is on how to produce useful complex global behavior of the system by means of the local interactions of its constituent parts. There are several properties characterizing multiagent systems. First each agent has usually incomplete information or capabilities for solving a problem alone, second, data are decentralized, third computations are asynchronous, fourth there is no a system global control or centralized design [161], and finally fifth they are usually open systems in the sense that new agents may enter or exit the system.

In order to interact in such an open environment agents have to overcome essentially two problems: they must be able to interact and to find each other, since they might appear and disappear at any time [92]. Further problems related with the possibility to simultaneously interacting with different types of systems will be discussed in the next section about Agent Societies.

MAS researchers develop interaction protocols, communication languages and agent architectures which can be used for the development of multiagent systems. Agents usually are self-interested and every one choose the best strategy for itself, which cannot be explicitly imposed from outside. Therefore the protocols need to be designed using a non-cooperative strategic perspective. This approach is required in designing robust non-manipulable multiagent systems where the agent may represent different real world parties and may be constructed by separate designers [135].

MAS researchers propose also mechanisms for advertising, finding, fusing, using, presenting, managing and updating agent services and information. One example is the proposal to introduce the notion of *middle agent* [40]. Middle agents are entities to which other agents advertise their capabilities and which are neither clients nor providers in the interaction under consideration. For example an agent may assist service requesters to find service provider agents or an agent may receive and hold requests for other agents [28, 113].

MAS technologies are nowadays used in many application domains like electronic commerce, digital libraries, information retrieval, user interface, robotics (such as Robocup), education and training, virtual environments, ubiquitous computers, social simulation (for instance [102, 65, 127]), etc. For an historical survey on multiagent systems aspects and applications, organized along the most important characteristics of the involved agents, that is the degree of heterogeneity and the degree of communication see [159]. As I will discuss in the next section, an important future prospective is to propose multiagent systems as a software engineering paradigm for the development of computer systems. But before its use will become a reality it is necessary to reach an agreement among designers on systematic methodologies for specifying and structuring applications as MAS.

3.1.3 Agent Societies

In the past few years a new issue of research regarding studies about *agent societies* and *agents organizations* is becoming more and more investigated and important in the Multiagent Systems community till becoming a wide area of research. Even if this is a young field of research, below I will try to outline a schema of the research works about Agent Societies and to sketch out some personal ideas about how to tackle some of the discussed problems.

The advent of Internet, probably the most complex, distributed, open and unpredictable environment that application designers have to face today, has as consequence the emergence of new situations and new possible applications for multiagent systems. Using Internet, for instance, different organizations that have never worked together in the past can decide to become partners in a transaction. Otherwise a trading agent may decide to interact with different open marketplaces with widely different regulations, in search for quality products and services at the best price. Such kind of applications are very important nevertheless, most agents' architecture and interaction systems proposed up to date are "closed", in the sense that agents cannot easily interact with different systems.

There are many reasons, listed below, why the realization of such *open* and *dynamic* systems is difficult [42, 41, 54, 33].

- First of all such systems must be able of coping with heterogeneous agents. To this end a standard way of interaction, that is, a common accepted *agent communication language*, is necessary, in order that the same agent without any changes in its design has to be able to interact with different systems.

- The members of a closed system rely for their interaction on a *social context*, consisting of a number of agreements on policies, protocols, duties, authorizations and meaning of the exchanged messages, usually discussed during the design time and incorporated in the agents. In order for heterogeneous agents to become able to dynamically interact, the social context of the interaction system has to be explicitly expressed. Moreover such social context is not static and may evolve over time as consequence of the agents' actions.

- Another very important aspect is the limited degree of trust among the agents. Agent cannot always be expected to follow the rules defined in the system for many reasons: presence of bugs, agents are self-interested, agents compete with each other, they have bound-rationality, etc. Given that there is a lack of control over the actions of independently developed agents, some mechanisms to detect violations of the accepted agreements, contracts, and rules are needed. Mainly, a method to detect violations of the commitments taken by agents during conversations and a way to test compliance of the participating agents to conversation protocols are needed.

- In some applications may be essential to go behind the simple detection of violations. In such situation a mechanism for *social control* to discourage agents from violating agreements and which may resolve violations or award incentives have to be introduced.

- Another relevant and problematic aspect of modern applications is that interacting agents represent their human counterpart, that is they act on behalf of their owners and very often their behavior in electronic space has legal consequences in the real world. In this case a clear and new concept of ownership of the agents has to be realized.

Many topics of research, consisting of looking for proposals to resolve the mentioned problems, come out from the previous analysis.

As regards the problem of defining a standard way of interaction among agents, in Chapter 5 a proposal for the definition of the semantic of an application independent agent communication language [61] will be presented. In such a proposal departing from the observation that communication is intrinsically social I define the meaning of various communicative acts using the notion of *social commitment*. A social commitment is a pledge of one agent to another agent that a certain state of affair holds or to undertake a specific course of action. Using the proposed agent communication language is possible to test the compliance of the various agents to an adopted conversation protocol and to detect violations of commitments.

In the remaining part of this section I will try to outline the existing approaches directed to resolve two of the mentioned problems: how to define and make explicit the social context of various systems and how to manage the legal aspects of agents' actions. The problem of effectively design, implement and manage systems with multiple interacting agents can be seen from two different prospectives: a macro-level prospective referring to the infrastructure of the system or from a micro-level prospective referring to the internal architecture of the agents. Below my focus will be on the macro-level aspects, which are considered by the multiagent community as highly delicate [101].

A system with multiple interacting agents can be considered a *computational organization* or an *artificial society* and then investigations in this field are typically inspired by works of organizational theorists, economists, and sociologists. Furthermore such systems has been studied by software engineering researchers [126], and recently by researchers in agent community [8, 182] (see [84] for a survey).

In the organizational approach the identification and definition of the groups, roles and relationships among roles is considered essential [138]. A group is simply an aggregation of agents and the set of roles available in a system is local to a certain group. Defining a computer system through a set of roles presents many advantages because they are an abstract description of the system, like for example in an auction the role of auctioneer and the role of bidder. While the concrete realization of the system is obtained through the association (instantiation) of the various types of roles with actual individual [182]. Moreover the instantiation is not necessarily static, throughout the system's lifetime, agents may change their role or take more than one role.

One of the first attempts to include organizational concepts in multiagent systems can be found in the work of Gasser in 1987 [67]. Other more recent proposals are listed below. The AALAADIN system [56] is based on a meta-model for artificial organizations design and can be used to build multiagent systems with different forms of organizations. The Gaia methodology [183] oriented to the development of closed systems with multiple cooperative agents and its extensions to manage the development of open system proposed in [182, 190]. An interesting example of formal specification of electronic institutions proposed by Esteva et al. can be found in [54], while another proposal that uses the notion of roles can be found in [96].

The roles available in a system and how these roles relate to one another can be defined in different ways. For example in [182] a role is defined using three attributes: responsibilities, permissions and protocols. Responsibilities determine what state of affairs an agent has to bring about and which ones have to avoid. Permissions are the "rights" associated with a role, for example the amount of resources that an agent can use or the possible operations that it can perform. An agent needs permissions in order to realize its responsibilities. Finally a number of protocols define the ways that an agent having a certain role can use to interact with the others. Similarly in [54] a role is defined as a finite set of dialogic actions and is characterized by some rights, duties and opportunities. In their formalization of the notion of roles the authors borrow and adapt the formalization proposed in [137]. While Jones and Sergot in [95] proposed a more subtle distinctions of the notion of institutionalized power from the concepts of permissions and practical possibility.

A different approach can be found in [41] where a Contractual Agent Societies is proposed. In this work the notion of roles is not explicitly mentioned, whereas there is the proposal to describe a social system and its social order as a set of *social contracts*, continuously negotiated, over the rights and duties of the participants. This approach relies on the assumption that is possible to develop agents capable of negotiating contracts and adapt their behavior to those contacts, a very difficult task. This viewpoint is influenced by work of organizational theorists and economists and by the observation that many organizations seem to be stable but are in fact the product of continual negotiation [160]. Finally some other works study how to introduce in a computational system off-line social laws [147] which guarantee the successful coexistence of multiple agents.

From my point of view a role can be defined using the rights and the duties associated with it. Duties are the set of *obligations* and rights are the set of *permissions* and levels of *authority* associated with a role. Usually rights are necessary to an agent with a certain role to abide by the duties. The notion of social commitment, crucial for the definition of the semantics of an agent communication language (see Chapter 5), may be crucial also for the definition of roles. The idea is to define the obligations associated with a role as a set of social commitments which specify the course of action that agents with a certain role have to undertake. While the rights associated with a role may be defined specifying inside the preconditions of all the possible actions the set of roles authorized to perform that action.

As regards the detection of violations of commitments, every agent of the system has to be able to detect the violations correlated with itself, and use such notion to updated its own level of trust or reputation of the various agents. While the creation of an appropriate system of *social control* goes behind the simple detection of violation but can be used to discourage agents from violating agreements and to propose alternatives to resolve violations. It has to individualize the agent deemed responsible for the breach of a contract and apply the corresponding sanctions. Studies about this issue of research can be found in the documentation of the European project ALFEBIITE [2]. They studied how trust may be created, how to manage legal consequences of agent's actions, and how some interactions between individuals and/or organizations are empowered by third party institutions.

3.2 Coordination of Systems with Multiple Agents

In a system of multiple agents that are performing some activities in a shared environment an important aspect is the method of *coordination* adopted for the harmonious and coherent functioning of parts to obtain effective results. It is an important issue of research that has been widely investigated in several areas of research: computer science, operations research, game theory, social sciences, economic theory, organization theory, linguistics, and psychology.

From an agent's prospective participating to a social situation involves the acceptance of some constrains on its behavior, in the sense that it must make its contribution to the system, and implies some advantages, in the sense that its participation provides resources and opportunities which would otherwise be unavailable [75]. In a system with multiple interacting agents there are at least three main reasons why their actions may need to be coordinate [89]:

- To take into account the dependencies between agents actions. Coordination is very important when goals of different agents are interdependent. For example when the results of the actions of one agent are required for the performance of another agent's actions. When the actions of one agent have to be delayed or brought forward to avoid conflicts with another agent's actions. When two or more agents have to perform their actions in parallel.
- When there is the need to meet some global constrains. For instance when some overall system's performance is required: spend a limited amount of money, give an answer within a fixed amount of time, maximize the social welfare (the sum of all agents' utilities), etc.
- Because an agent may need the competence, resources, or information of other agents to accomplish its tasks.

The coordination of systems with multiple agents is a very general problem and the great diversity in all existing studies makes it complicated to organize a complete and schematize treatment of this matter. A first interesting distinction that can be made is between coordination methods with an internal prospective and coordination methods with an external prospective. That is between methods that may be explicitly dealt with by the internal structure of the agents and methods that result from limitations on the set of possible agents' actions, obtained for example with fixed interaction protocols or with social structures.

A crucial distinction, which follows from the historical evolution of coordination methods, is between methods for cooperative agents and methods for competitive or even conflict agents.

Another important distinction can be made among coordination methods for "closed" systems and coordination methods for truly "open" systems. The main characteristic of open system is that various heterogeneous agents might enter and leave it dynamically and autonomously.

3.2.1 Cooperation

As discussed in Section 3.1.1 about Distributed Problem Solving, following a distributed approach to resolve a problem means decompose it in smaller sub-tasks, which might present some interdependencies, and then distribute such tasks to different autonomous components (agents) on the basis of their capacities, of the resources required to accomplish the task, and on the temporal and spatial dependencies among the tasks.

The mechanisms developed to distribute the tasks among cooperative agents govern the exchange of series of messages among the various entities to maintain global coherence of the system without an explicit global control. Such mechanisms can be viewed as methods to coordinate the work of the agents from an external prospective. In this case the main assumption made in the system is that the various autonomous agents of the systems are benevolent (cooperative) towards others components, in the sense that they work together toward achieving some common goals. Coordination among benevolent, non-antagonistic agents is called *cooperation*.

Among the various implementation-oriented models for realizing cooperative systems [154, 48] the best known and most widely applied is the Contract Net Protocol [153, 39]. Other also well-known mechanisms are based on the metaphor of the "blackboard" that is used by the partners to interact, cooperate, and give their contribution to the development of a solution for the given problem.

In a different way, following an internal prospective agents can coordinate themselves with the others reasoning about their local actions and reasoning about the future actions of others to try to ensure the community acts in a coherent manner [89]. In particular Jennings proposed a coordination method based on the following hypothesis: "all coordination mechanisms can ultimately be reduced to (joint) commitments and their associated (social) conventions" [88, p.5]. Commitments in this case are viewed as pledges, that an agent takes with itself, to undertake a specific course of actions. While conventions constrain the conditions under which commitments should be revised by the agent. Those two concepts embody the notion of intentions (discussed in Section 2.3). Commitments are useful to overcome agents' bounded rationality, necessary for agents to plan and coordinate their future actions, to reason and predict other agents' actions, and to avoid resource conflicts or consider global constrains.

A coordinated behavior among very simple components can be obtained also without communication but from observations of a shared environment. An example of this type of coordination can be found in colony of ants where trace of pheromone is used to exchange information about the environment. These kinds of systems very often are able to resolve heuristically very complex combinatorial problems [12].

Recently in works on cooperative problem solving in which a group of autonomous agents choose to work together to achieve a common goal, benevolence among the agents is not any more assumed. Agents are autonomous and are free to decide to cooperate with others. These models usually include the stage of recognition of potential partners, the team formation, the negotiation of a joint plan and the execution of the negotiated roles. An example of formalization of this model can be found in [181].

3.2.2 Competition

There are also many other important reasons, rather than creating one large efficient system, why software components that are networked need to interact. For instance when agents represent different interests in electronic commerce applications. In these systems the benevolent assumption no longer holds, the various agents are *self-interested*, in the sense that they act to further their own interest and their own goals possibly in conflict with the goals of other participants. Viewing agents as selfish may be important also to make the system more tolerant to partial failure of its components. Coordination among competitive or simply self-interested agents is called *competition*.

In competitive systems a frequent form of interaction is *negotiation*, a mechanism used by the agents to interact with the scope to determine a contract or less formally an agreement.

Negotiation among multiple agents can consist of structured exchanges of messages governed by a specific *interaction protocol* which represent a fixed frame that has to be followed by agents to communicate. Automated negotiation topics will be widely treated in Chapter 4, where the trading agent Nidsia will be described. Example of interaction protocols devised for competitive agents are: Contract Net Protocol modified to overcame the problems that arises with self-interested agents, which can break their contracts [133], Market Mechanisms [174] effective for coordinating the activities of many agents with minimal direct communication among them, all different type of auctions, voting, bargaining etc.

A research topic in negotiation theory is to devise interaction protocols that guarantee some performance for the overall system with the assumption that the interacting agents have their own goals and follow a best strategy form themselves. Researchers in Game and Economic Theory proposed relevant formal theories about the circumstances under which cooperation can occur in a society of self-interested agents [7] and how negotiation protocols can be designed to ensure that truth telling is the optimal strategy [129]. However these models present some drawbacks: are based on the restrictive assumption that the agents involved in the interaction always act in order to maximize their expected payoff; assumes that is possible to order agents' preferences with respect to possible outcomes, for example through an utility function: a hard task when agents act on behalf of a human counterpart; finally these models often assume perfect computational rationality.

In other situations agents may use a *communication language* to interact and negotiate. In these cases agents have to be able to understand the meaning of the exchanged messages and to reason on the behavior of other agents. The definition of a standard Agent Communication Languages (ACL) includes the definition of the meaning of a set of communicative acts. Almost all the efforts present in literature to define an ACL are based on Speech Act Theory [6, 139]. So far, two very famous ACLs are KQLM [57] and FIPA ACL proposed by the Foundation for Intelligent Physical Agents (FIPA) [60]. Those languages use mental states of the interacting agents to define the meaning of the communicative acts, an approach that is adequate in cooperative multiagent systems. While in Chapter 5 a new proposal for facing the crucial problem of defining an agent communication language for open competitive multiagent systems, based on the notion of social commitments, will be presented, discussed and compared with other existing approaches.

3.2.3 Coordination in Open Systems

Recently in large, complex, and open multiagent systems the distinction between cooperative agents interested in optimizing the overall system performance and competitive agents that are trying to maximize their own performance is blurring [101]. In the sense that in complex societies self-interested agents behave in more cooperative ways in order to acquire information, services or products from other agents. While benevolent agents will behave in more selfish ways, given the costs of understanding the more global ramification of their actions and interactions. Moreover in open system heterogeneous agents may enter and leave dynamically. For these reasons in recent and future proposals of coordination methods for open and dynamic systems of multiple heterogeneous agents no assumptions can be made about the internal structure and behavior (benevolent, self-interested, utility based, etc) of the agents.

Moreover in order that the same agent could be able to interact with different systems *standard* and *objective* methods of interaction and coordination among agents have to be defined. Examples of those methods of interaction are agent communication languages, for instance the one proposed in Chapter 5, and *standard conversation protocols* which are interaction protocols defined using a standard set of communicative acts. A communication language can be used only by complex agents able to understand the meaning of the various communicative acts. While standard conversation protocols might be used also by simple reflex agents designed to follow the protocol.

Usually to obtain effective coordination among agents in these open multiagent systems an organizational context has to be defined as discussed in Section 3.1.3. Such social context is essential for the performance of certain types of communicative acts, for example orders and declarations, or to support various institution, for instance the notion of property [33].

3.3 Conclusions

In this chapter I have described the main reasons and advantages of developing applications where multiple autonomous and heterogeneous agents may interact. Following an historical prospective, the research focus of Distributed Problem Solving was first presented. Then I presented Multi Agent Systems and Agent Societies research topics arisen thanks to the advent of Internet and to the necessity of different organizations scattered around the world to interact. In particular the problem of studying efficient and effective methods of coordination among various autonomous agents and the problem of realizing truly open interaction systems. In Chapter 4 various interaction protocols and in particular auction mechanisms will be studied through the experience of developing a trading agent for an international electronic commerce competition. Given the limits of adopting a fixed interaction protocol in Chapter 5 standard agent communication languages will be studied and a new one based on the notion of commitments will be proposed. In order to realize open interaction systems, besides the definition of a commonly accepted agent communication language, a social framework for the definition of the social context of agents interactions has to be defined ad discussed in Section 3.1.3.

Chapter 4

Automated Negotiation

As discussed in Chapter 2 a distinctive characteristic of artificial agents is *autonomy*. Agents have a high degree of self-determination with respect to other agents, in that they autonomously decide when, what, and under what conditions their actions should be performed.

In most cases, as discussed in Chapter 3, such autonomous agents need to interact with other agents either to cooperate or to compete. For example they may need some information, resources, services or products from other agents, to achieve their own goals. While in other situations multiple autonomous agents may desire to synchronize their actions to perform very complex tasks that they cannot carry out alone. Possible methods of interaction used by autonomous agents are: fixed interaction protocols (discussed in this chapter) that enable agents to have structured exchanges of messages and conversations, free exchanges of messages that agents are able to understand and that are defined in standard agent communication languages (as discussed in Chapter 5).

In future societies of agents, as discussed in Section 3.1.3, the notion of autonomy of artificial agents will be better defined by the limits imposed by an institution to the available actions that every agent may perform on the base of its role in the society [33]. In any case usually agents do not give orders to other, but they have to try to persuade their interlocutors to act in a certain way.

A possible pattern of behavior used by agents to persuade someone else is *negotiation*: the process by which two or more agents communicate to try to reach a common mutually acceptable agreement on some matter. Negotiation theory is important for many different contexts and has been studied from different point of view: Artificial Intelligence, Game and Economic Theory, Social Psychology, Linguistics. Following the classification presented in [90], automated negotiation research usually deals with three main topics:

- *Negotiation Protocols*. The design of the set of rules that govern the interaction among agents. A negotiation protocol defines the possible *roles* of the participants, for example in the English auction there is an auctioneer and there are many bidders, while in the continuous double auctions there are many sellers and many buyers. For each roles the negotiation protocol define the valid actions in particular states, for instance which kind of messages can be sent or have to be sent by whom, to whom, at what stage of the interaction. The negotiation protocol may also define the events that cause some changes in the state of the negotiation, for example the negotiation duration or in an auction the rule to assign the negotiated product to the winner. The main research topic in this area is to study the overall properties of the system given that a certain negotiation protocol is adopted by the participants [129]. Some of the most famous negotiation protocols and of their desirable properties will be discussed in Section 4.1.

- *Negotiated issues and aspects.* The definition of the issues over which an agreement must be reached. In many cases, for example in auctions, only one property of the object is negotiated, usually the price. Generally may be possible that numerous properties of the object are negotiated, for example quality, date of delivery, penalties and so on. Another issue concerns the type of operation that participants can perform on the agreement under negotiation. Very often the structure of the agreement is fixed and participants can only accept or reject it. In other situations the negotiation may be much more effective if the participants are able to change the characteristics of the agreement. For example participant can answer to a proposal with critics or counter-proposals. Participants may also be allowed to extend the structure of the negotiated object, for instance a travel agent may add some entertainments to a travel package in order to clinch the deal. Instead of negotiating only the issues of the object it is possible to include other aspects. For example participants might be allowed to justify their negotiation stance or try to persuade another to change its negotiation stance in order to convince the others to change their mind. The negotiation where agents provide arguments to support their stance is called *argumentation-based negotiation* [98, 91].

- *Agents' Decision Making Models.* This topic, discussed in Section 4.3, concerns the design and development of the decision making mechanism of participants on the basis of the restrictions imposed by the negotiation protocol, on the nature of the negotiation object and in order to achieve their goals or to maximize some sort of utility. Regarding to the utility notion, usually agents negotiate on behalf of their owner. An important area of research concerns the problem of representing in artificial agents the desire, preferences, interest, and prejudices of their human being counterpart.

This chapter is organized as follows. In Sections 4.1 various interaction protocols, in particular auctions will be presented. The focus is on auctions because they represent the most used protocol in electronic commerce (presented Section 4.2), one of the most important applications of automated negotiation studies. Subsequently in Section 4.3 the decision making model of a trading agent for electronic commerce that I developed will be described and discussed. In particular a bidding policy for buying products that present various combinatorial correlations and that are sold parallel auctions with different rules will be studied.

4.1 Negotiation Protocols

Interaction protocols define for each stage of the interaction the set of possible actions (messages) that participating agents can perform or have to perform, usually to progress towards deals. Interaction protocols can be used to define the roles played by various agents in interacting systems, in particular to define the agents' duties [33]. By restricting the set of possible actions that an agent can perform they allow to very simple agents, for example reactive agents, to take part to complex interactions, while the usage of an agent communication language requires more complex, usually deliberative, agents. Moreover sometime they are necessary to reduce to the minimum the number of messages exchanged among agents and thus avoiding the congestion of the entire system. As discussed in Chapter 3, some protocols are devised to operate on group of self-interested agents that may have conflicting goals while some other protocols are devised for group of agents that have similar goals or common problems and the objective of the protocol is to maintain globally coherent performance of the agents without explicit global control.

Negotiation protocols can be designed in order that the overall interacting system shows certain desirable properties. Below are listed some of those possible properties [135, p.202].

Evaluation Methods of Negotiation Protocols

- *Maximizing social welfare.* A protocol maximizes social welfare when maximize the sum of all agents' utilities. To be applied this criteria require that the agents' levels of welfare may be comparable, this is not obvious in open systems made by heterogeneous agents.

- *Pareto efficiency.* A negotiation outcome is Pareto efficient if there is no other solution where one agent's utility is higher without making another agent's utility lower. If a certain outcome maximizes social welfare then it is also Pareto efficient. Otherwise it is possible that a solution is Pareto efficient but it is not social welfare maximizing. In fact it may happen that in another solution an agent's utility increases much more than another agent's utility decreases.

- *Individual rationality.* An protocol is individual rational if for the agent is better taking part to the negotiation than not participating to it.

- *Stability* A protocol is stable when an agent is motivated to behave in a certain way. Reminding that a strategy is a mapping from state history to action, a protocol may have *dominant strategies* when an agent is motivated to behave in a particular way without taking into account other agents behavior. While often the agent's best strategy depends on other agents behavior. The strategies of a set of agents are in *Nash equilibrium* if no agents is motivated to change its strategies given that the others do not change their ones. Nash equilibrium has some problems: in some games it does not exist and some other games have multiple Nash equilibriums and it is not obvious which one the agent should actually play. Furthermore it is a weak properties because group of agents can deviate in a coordinated manner.

- *Computational efficiency.* A protocol is efficient if it leads to solutions that satisfy some evaluation criteria with the lowest computational work.
- *Distribution and communication efficiency.* A protocol should be preferred if has not a single point of failure and minimizes the amount of communication.

The remaining part of this section presents different interaction protocols. In particular I will focus first on the Contract Net Protocol a very important protocol widely used in real applications. Secondly on auction mechanisms broadly applied in electronic commerce as discussed in the following parts of this chapter.

4.1.1 Contract Net Protocol

The Contract Net Protocol has been proposed by R. G. Smith in 1980 as a "high level protocol for communication among nodes in a distributed problem solver" [153, p.1]. As discussed in Section3.1.1 Contract Net Protocol has been initially used to solve the so called *connection problem*, that is to allocate tasks to an appropriate agent when a distributed solving approach to a problem is followed. It is usually applied to problems that do not have a well-defined algorithm for their solutions and when the presence of numerous tasks and various nodes can lead to a combinatorial explosion of the number of possible solutions.

The collection of all nodes of a net, able to solve tasks is called contract net, while a contract is the deal between two nodes for the execution of a task. Every node of the net can take on of two roles: the *manager* who desires that a task will be executed or solved and is responsible for processing the results of the task's execution; the *contractor* the agent that executes the task.

The protocol prescribes the following steps in the interaction: a manager announces a task, then various contractors evaluate it and eventually submit bids to execute that task. The manager evaluates the bids and awards the task to the most appropriate node. A contractor may then further partition the task and subcontract it, becoming the manager for those subcontracts creating a hierarchical control structure. After executing the task various contractors report the results to the manager who synthesizes the results.

The messages used in the protocol are:

1. *Task announcement.* It is used by a manager of a task to announce it to all potential contractors. This message includes slots for: *addressee*, the name of the potential contractors; *eligibility specification*, the conditions that every potential contractor must meet; *task abstraction*, a brief description of the task used by contractors to rank tasks from several task announcements; *bid specification*, an explain to potential contractors of what information must be provided with the bid, it will be used by the manager to compare bids from different potential contractor; *expiration time*,

the deadline for receiving bids. A common language is needed in order that every node is able to understand such information.

2. *Bidding*. The potential contractors process the announcement, evaluate unexpired task announcements and decide if offer a bid.
3. *Awarding the task*. The manager receives all bids, processes them and awards the task to one of the contractors. The award message includes the data necessary for the execution of the task.
4. *Reporting results*. The information message is used for general communication between the manager and the contractor during the processing of the contract. The report message is used by the contractor to inform the manager about the state of the execution. The result description message contains the results of the execution.
5. *Termination*. The manager can terminate a contract with the termination message and the contractor terminates the execution of the corresponding task.

The advantage of this protocol is that it is very simple and it allows tasks distribution without any negotiation. The Contract Net Protocol may be suboptimal in the allocation of tasks, in fact a task could be awarded to a contractor with limited capacity if a better qualified contractor is busy at award time or ranks the announced task below some other tasks under consideration. The manager can resolve this problem asking an immediate response to contractors, for example an answer like eligible but busy, ineligible, uninterested. Another limitation of this protocol is that the manager is under no obligation to inform potential contractor that an award is already be made. The Contract Net Protocol is similar to a first-price sealed-bid auction (see below) where the manager is not obliged to award the task to the highest bid.

The adoption of the Contract Net Protocol in the interaction of cooperative multiple agents leads to good performance for the overall system. While the Contract Net Protocol presents some limits when applied to the interaction of competitive agents. Sandholm in [136, 133, 135] proposes some improvements to the protocols to overcome those limits. In particular he introduced a formal description of the contractors' decision mechanism based on the marginal cost calculation. Moreover he introduced the notion of *contingency contracts*, where the obligations of the contract are made contingent on future events, and of *level commitment contracts* where different penalties are used to choose among different levels of commitment. It was shown that the leveled commitment protocol outperforms the full commitment protocol.

4.1.2 Auctions

Auction mechanisms are particular interaction protocols that fix the rules of the interaction among one or more sellers and one or more buyers and fix the rules for determining resource allocation and prices on the basis of bids from the market participants. In auctions the seller, also called the auctioneer, want to sell an item at the highest possible value while buyers, also called bidders, want to buy the item at the lowest possible price. Therefore participants to an auction are self-interested agents that want to maximize their payoff, but usually an auction is not a zero-sum game, that is games in which one player's losses are the opponent's gains. An auction's outcome is usually the definition of a contract between a seller and a buyer for the exchange of a certain product. Usually auctions are used when the quantity of products available is limited and there are at least two participants that compete to get the products.

Auction mechanisms are also used to sub-contract tasks to various agents, in this case the seller want to obtain the lowest possible price while the buyers the highest possible payment but the mechanisms for this setting is analogous to the mechanisms for the previous setting.

Auction mechanisms are nowadays widely used in electronic commerce web sites for buying and selling items among different users. Examples of existing marketplaces that include various auctions where sellers and buyers can exchange commodities are: eBay [51] the most famous auction site, Yahoo! auctions [187], MSN auctions [110], uBid [164], to mention only a few.

Auction mechanisms present some advantages:

- Using auctions to sell an item is more flexible than setting a fixed price, especially when it is not easy to fix a value for the item. For instance when the products do not have a fixed or determined market value and the seller is unsure of the price he can get.

- Reaching a deal following the rules of auctions is less time-consuming than negotiating a price. In fact in auctions the competing bids are offered almost simultaneously.

- The participants have a common knowledge about the rules of the negotiation process and usually every participant has to commit to such rules before entering the auction site.

- The number of exchanged messages is minimized and usually is easier to reach an agreement.

There are different reasons for bidding in an auction. If a bidder desires the item for itself it makes its own *private valuation* of the product. In other cases a buyer's scope can be to acquire an item for resale. In this case the buyer has to compute what is called the *common value* of the item. Finally there are cases when the value of an item depends partially on its private value and partially on other's value, it is called the *correlated value* of the item. This may happen when the buyer decides after the purchase if keeping the item or sell it. The bidding behaviour of buyers changes upon which motivation is driving them.

An important limitation of auction mechanisms is that only the price and in some cases the quantity can be negotiated, even if other aspects may be important like delivery schedule, method of payment, etc.

On the basis of different value of certain parameters is possible to have many different types of auction mechanisms. The most famous types of one-side auction, that is with one auctioneer and many bidders, are: English, First-Price sealed bid, Dutch, and Vickrey. While the most important type of double auction, that is with many sellers and many buyers is the Double Auction.

- In the *English* (fist-price open-cry) auction there is one auctioneer and many bidders. Each bidder can bid for the sold item at a price that has to be higher than the current best bid. When no bidder is willing to raise anymore the price the auction ends and the item is awarded to the highest bidder at the price of his bid.
- In the *Dutch*(descending) auction the auctioneer starts the price of the item at a very high value, then the price is progressively lowered until a bidder takes the item at the current price. This type of auction is typically used to sell fish and flowers because is very efficient in terms of the time necessary to reach a deal.
- In the *First-Price sealed-bid* auction the price of a bid is hidden from other bidders, in fact each bidder submits one bid without knowing the other's bids. The highest bidder wins the item and pays the amount of his bid.
- Also in the *Vickrey* (second-price sealed bid) auction the price of a bid is hidden from other bidders, but in this auction the highest bidder wins the item at the price of the second highest bid [167].
- The *Double Auction* [64, 185] is one of the most famous auction where multiple buyers and multiple sellers submit their bids. Those bids are ranked and the lowest sell offer is matched with the highest demand bid until there are sell offers lower than demand bids. The price of the exchange is a price between the sell and buy offers matched.

An important problem in this field is the lack of common names for the same type of auction. An attempt to standardize various interaction protocols has been done by the Foundation for Intelligent Physical Agents (FIPA), an international organization that is dedicated to promote standards in the field of agents [60].

There is not a general rule to decide which auction type is best. It depends from which perspective the question is posed. For example the seller's point of view is different from the buyer's point of view, in some cases the time needed to reach an agreement may be crucial, while in other cases it is important to decrease the incentives to cheat. In any case in every auction type bidders can be penalized by the "winner curse" that is the winner of an auction can pay for the item more than it is worth. An interesting analysis of various auction protocols with their evaluation using the criteria reported above can be found in [135].

There are different types of auction and on the basis of my experience in this field, in particular with the Trading Agent Competition discussed below, I propose to use the following attributes to identify the principal characteristics of every type of auction. This is a classification specific for auctions a more general classification for negotiation in electronic commerce is presented in [104]. This classification is used in Appendix A to present a schematic description of the three auction types used in the Trading Agent Competition.

A Classification Scheme for Auction Protocols

Before presenting my proposal, some terms have to be introduced. In a generic auction the auctioneer applies the rules in accepting bids, reveals status information through *price quotes*, and determines the resulting *clearing price*. An auction's clearing price represents the exchange terms agreed upon by the bidder and the auctioneer as a result of the auction. The rules of the auction specify exactly how the clearing price is determined. Between the start of an auction and its final clearing, the auction may reveal intermediate status information based on bids received through a designated price quote. The standard price quote consists of two elements.

- *Bid quote*, the maximum price at which a new sell bid would be matched by the existing bids. Any existing sell bid less than the bid quote would be matched in the current state.

- *Ask quote*, the minimum price at which a new buy bid would be matched by the existing bids. Any existing buy bid greater than the ask quote would be matched in the current state.

At the time of the price quote, there may exist sets of buy and sell bids that already match. In interpreting a price quote, it is important to remember that bids may have been added, modified, or withdrawn since the price quote has been computed.

The proposed classification scheme includes the following parameters:

1. Number of buyers and number of sellers.

2. Types of objects. This parameter explains if the object of the negotiation is a discrete or continuous set of products.

3. Rules for bids acceptance. For example the rule may be: new bids must dominate previous bids or the reservation price, or new bids must beat current price quote. Another rule may be that a null bid means withdraw, or that there must be a minimum bid increments. Other rules may be about bid expiration conditions or about whether a bid may be subdivided.

4. Type of bids. For example buyers can offer for one unit of the good or specify both a price and the number of units of each good. Less common way of bidding is placing combinatorial bids in exclusive-or, for example when the buyer wants this quantity with this price or that quantity with that price but not both. This bid type is only valid in an auction with multiple products. Finally different types of bid may include some decommitment rules.

5. Allocation policies. They concern the clearing policy, for example in some auctions the good can be awarded to the highest bid or some other important factors can be taken into account, like delivery requirement or terms of payment. Criteria for resolving multiple bids tie at the same price, possible criteria is awarding to the earliest bid or to the one with the larger quantity. Criteria for determining the prices, for example different prices for different bidders or the same price for all bidders. The frequency of clears, like continuous match, scheduled match, random match, inactivity match, activity match. From the frequency of clears parameter determines if agent's deliberation time is more or less bounded.

6. Auction closes rules. These rules determine whether a clear should be the final clear. For example an auction can close at a fixed scheduled time or at random time or after a period of inactivity.

7. Information made available to buyers. Quote information can be provided or not, for example this is a list of different possible strategies: the best bid can be shown when placed; all bids can be shown when placed; winning bids are shown at clearing; all bids are shown at clearing; only a price quote value is shown computed on the value of the current bids already received. Other information may be the frequency at which the price quote is communicated, the number of price quotes, the transaction history on prices, quantities or the identities of the agents.

8. Interaction format. For example open cry, in this case buyers must be able to response to each bid in a short time, or sealed bid.

4.2 Negotiation in Electronic Commerce

One of the most important and concrete applications of automated negotiation is *Electronic Commerce*. The term "commerce" refers to all stages of the interaction between buyers and sellers to exchange commodities on large scale, involving: finding products or potential buyers, negotiating and purchasing, and delivery. In the last years thanks to the constant growth of the Internet a lot of companies start to run part of their business-to-business and customer-to-business transactions in electronic format.

Agent-based technologies may play a crucial role in Electronic Commerce in both the first stage of commerce with the use of agents that act as intermediary in the search of products or merchants and in the step forward with agents that may negotiate deals and purchase products. Negotiation in electronic commerce is a difficult task and the help of artificial agents in making decisions can be crucial. In fact there is a high richness of information, the conditions for deliberation are constantly changing and highly uncertain, and deliberations are time-bounded.

Other issues of research investigated in Electronic Commerce that are very important in order that it could become a reality are the definition of standards and the definition of security features. Regarding to standard, important components are the definition of standard agent communication languages, the definition of a social framework where interactions can take place, and of standard ontologies for categorizing objects and define their relations in order to give them a meaningful semantic for artificial agents. Regarding to security, especially for commercial tools used by customers to participate in electronic commerce, important issues are the definition of methods for authentication, access control, encryption, signatures, auditing and traceability, public key infrastructure and certification. Finally also legal issues about the privacy of personal data that are processed and exchanged by negotiating systems are crucial.

In real applications, up to know, agents are used as products and merchant brokers and as customer assistant in closed markets, that is, in market place where artificial agents are all designed by the owner of the market. This happens for two main reasons. The first one is that the strategies of artificial trading agents are not still enough reliable and secure to use them as fully negotiator by the users. Mainly because it is hard to perfectly forecast their behavior in a constantly changing environment. Moreover artificial agents with programmed strategies and a limited flexibility may be overcome by human participants who are more flexible and innovative or by other software agents that know their negotiation strategy. For example if a buyer knows that a seller want to sell a good above a certain threshold value it will offer exactly that value that is the worst possible for the seller. The second reason is the absence of open systems where interacting agents made by different designers can interact and negotiate. While in close systems buyers will be reluctant to delegate to an agent, that is not completely under his control, to sign up contracts.

Examples of implemented Negotiation Support Systems (NSS), that is software systems devised to help human negotiators to make better decisions are INSPIRE and INSS [87], both developed at Carleton University. They help human users to negotiate a solution, help them during the specification and assessment of their preferences, the system contains a scoring function to aid users in the construction of offers, a graphical displays of the negotiation progress, and a facility for computing non-dominated compromises.

It is possible to envision that initially trading agents will be used to explore the set of possible agreements and pass this information to the customer for the final decision. In future, agent-mediated electronic commerce applications with fully automated negotiation will be exploited when the application domains will have some of the following characteristics [149].

- The time between proposal and counter proposal is very short and there is no time to wait the intervention of the user between trading rounds.
- Interactions are repeated with either high communication or the domain is limited. In this case agent can effectively learn about the user behavior or about other participants behavior.
- The value of each trade is relative small. In this case it is possible to monitor the sequences of transactions and stop it if some problems arise without significant losses. With the term "relative" is intended to stress the differences between the value of a transaction for a company or for a private customer. Consequently most probably

agent will be firstly used in business-to-business transactions.
- The trading process is repeated over long time to justify investments in software, hardware, training, and to let the agent learn the customer preferences.
- The negotiated object is relatively easy to specify. It is difficult to negotiate over complex object, mainly for problems related to the semantics of the communication.

The simplest interaction protocols used up to know in real electronic market places are *auctions* as discussed in Section 4.1.2. The application of artificial agents in taking part to electronic auctions is very interesting in fact for human being is very difficult or even impossible to evaluate so quickly all the possible advantage and disadvantages of each possible bid, and to evaluate how to bid on correlate products. Automated shopping through on line auctions is considered plausible in the near future [52].

Environments for testing trading agents

Different research groups are looking for the development of electronic market places where various trading agents may negotiate and their performance may be tested and compared with the other agents. Examples are Kasbah [24] a MIT Media Lab project ended in 1999. Kasbah was devised as a marketplace where different users can sell and buy products buy creating buying and selling agents that negotiate on their behalf. A key feature of Kasbah is that it is open in the sense that new types of agents using different selling strategies may be added to the system.

Fishmarket FM [58, 128] developed at the Artificial Intelligence Research Institute (IIIA, [85]) of the Spanish Scientific Research Council. Fishmarket is an electronic auction house, based on the traditional fish market, where agents of arbitrary complexity may trade. The auction type is downward (Dutch) bidding protocol and trading agents can be of arbitrary complexity, even human agents, it is enough that these agents limit their messages to the fixed protocol that the market system understands. Such agents can interact with the market through standardized Java agent-interface applets buying and selling various types of products.

The Auctionbot server [186] developed at Artificial Intelligence Laboratory at the University of Michigan for research purposes, now officially retired and used as market place platform for the Trading Agent Competition series [162] (see Section 4.3.1 for more details). AuctionBot was a flexible, scalable, and robust auction server for the interaction of both human and software agents through different type of auctions.

4.3 Agent's Decision Making Model for Simultaneous Auctions

Given the presence of so many auctions marketplace on the web an interesting research focus is on studying how to create software agents that monitor and participate in these auctions. Designing software agents for bidding in an electronic auction for a single good, especially for certain widely studied types, like the Vickrey auction, is a straightforward task. Much more complex is studying and designing *bidding policies* for trading agents whose task is to buy *complementary* and *substitutable* products in *parallel* auctions.

Usually the goal of trading agents is to maximize their utility: a value that is function of the bought and sold commodities and of the prices of the exchanges. Choosing a bidding policy means deciding in *which* auction to bid, *how much* to bid and *when* to bid on the basis of the products already owned and on the information available about the running auctions. Some products could be complementary, meaning that obtaining one good without another makes that good worthless. Some other products could be substitutable, meaning that obtaining a certain bundle of products can lower the value of obtaining another, or render it worthless.

This kind of problem is very common, in fact it happens very often that people want to buy some products but it results a complicate task in presence of multiple sellers that offer various resources of interest and when the products present some correlations.

Nowadays existing studies on bidding strategies are few and usually specific to a certain problem. Moreover a coherent repository, which describes which negotiation technique is better for a given problem or domain, does not exist. An interesting approach may be to organize competitions in order to be able to compare different approaches to common problems. There are numerous examples of such competitions where different heterogeneous agents compete and interact directly, this means that the success of an agent's strategy depends on the strategies of the other competitors. Examples are the various editions of the robot soccer world cup (RoboCup) [156, 155] from 1997, with the scope of encourage research with a focus on developing cooperation between autonomous agents in dynamic multiagent environment. The Fishmarket trading agent tournaments [132] started up in 1997 where trading agents developed by different designers compete in downward bidding auctions. The Axelrod's iterated prisoner's dilemma tournament from the late 1970's [7]. The Santa Fe double auction tournament in the late 1980's [132] and the recent RoShamBo programming competition [11]. The three series of the Trading Agent Competition (TAC) [162] described in next section.

In the following sections I will present and discuss the bidding policy for simultaneous auctions used by the trading agent Nidsia in the First International Trading Agent Competition (TAC) [175]. The content of these sections is mainly based on the paper "An autonomous bidding agent for simultaneous auctions" by Fornara and Gambardella [63].

4.3.1 Trading Agent Competition

The first International Trading Agent Competition (TAC) held in Boston on July 2000 was organized by researchers at the University of Michigan and North Carolina State University. A detailed description of the game written by TAC team can be found in [175]. TAC-00 was followed by TAC-01 held in Tampa, Florida on October 2001 and TAC-02 that will be organized and run by the Swedish Institute of Computer Science (SICS) in 2002 [162].

TAC-00 included 22 initial teams that competed again each other in preliminary rounds, the 12 best ones (included the trading agent described in this section) from 6 countries were invited to the semi final in Boston. For a complete description of the 12 final competitors see [157].

The game

TAC market place runs at the University of Michigan and the various bidding agents are connected to the server over the Internet using TCPI/IP socket to send bids and to update their knowledge of the market state. During the competition, the agents take part in series of games. During each game eight trading agents compete bidding in various auctions that start at the same time, run in parallel and can last till the end of the game or finish in advance. Each competitor is a travel agent with the goal of assembling a travel package for each one of its eight customers. These customers express their preferences for various aspects of the trip. The objective of the travel agent is to maximize the total satisfaction of its customers.

A travel package consists of a round-trip flight, a hotel reservation in a good or bad hotel, and tickets to three types of entertainment events (Boston Red Sox Baseball, Boston Symphony or Boston Theater). A travel package is *feasible* if it contains rooms in the same hotel for every night between the arrival and departure dates. The soonest that any customer can leave is one day after arrival. An entertainment ticket is useful and can be inserted in a feasible package if none of the tickets are for events on the same day, all of the tickets coincide with nights the customer is in town, and all assigned tickets are of different type. From this brief description it is clear that flight tickets and the corresponding hotel rooms are complementary products and that night rooms in the good or bad hotel and entertainment tickets on the same day are substitutable products. A concise description of the TAC game is depicted in Figure 4.1.

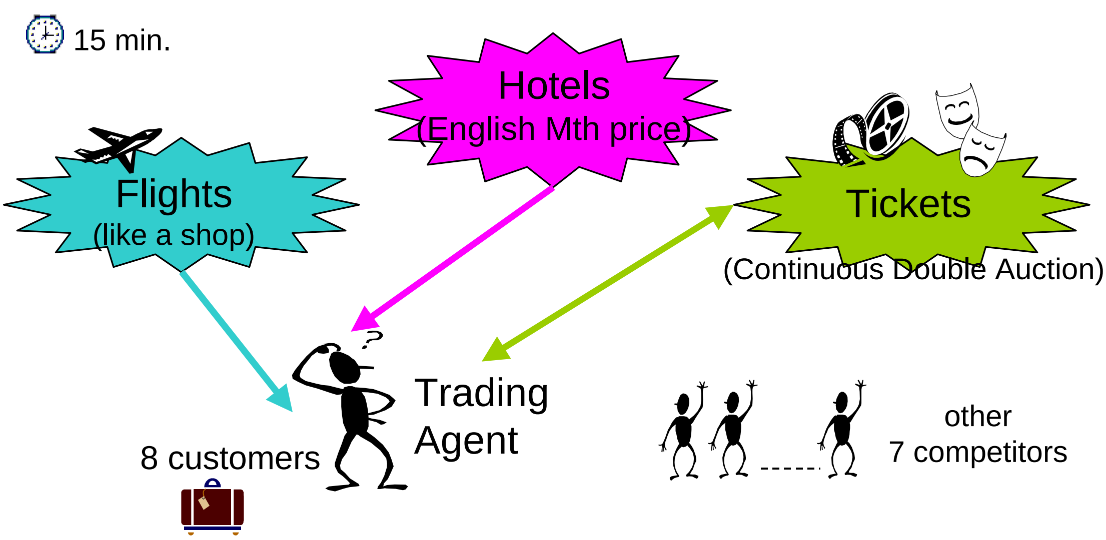

Figure 4.1: The game of the Trading Agent Competition 2000

The utility function

At the start of each game the trading agents query the auction server their customers preferences on travel packages and their initial endowment of entertainment tickets. The preferences of the clients are expressed by a preferred arrival date AP; a preferred departure date DP; a reservation value for upgrading to the better hotel HU and a reservation value for each type of entertainment event BRS, SY, PH. In the travel packages pairs of arrival/departure days are equally likely, hotel reservation values are chosen for each customer uniformly in the range \$50 to \$150 and entertainment reservation values are chosen uniformly in the range \$0 to \$200. A travel package is specified by an actual arrival date AD, an actual departure date DD, a grand hotel indicator GH in {0,1}, and a ticket indicator for each event type BT, ST, PT, each in {0,1}. The utility for each customer is:

$$Utility = 1000 - travel\_penalty + hotel\_bonus + fun\_bonus; \tag{4.1}$$

where

$$travel\_penalty = 100 * (|AP - AD| + |DP - DD|); \tag{4.2}$$

$$hotel\_bonus = GH * HU; \tag{4.3}$$

$$fun\_bonus = BT * BRS + ST * SY + PT * PH; \tag{4.4}$$

The final score, for the trading agent, is the sum of the utility for each customer minus the expenses and minus the negative entertainment balances. In fact if an agent tries to sell tickets it does not have, it is assessed a penalty of 200 for each ticket. If the travel package for a certain client is not feasible the agent receives zero utility for that client.

Auction rules

Each game instance in TAC last 15 minutes. There are three different auction types: one for flights, one for hotel rooms and one for tickets. There is a separate auction for each good and each one of the 5 possible day of the travel package, up to a total of 28 parallel auctions. A detailed description of each auction type structured following the schema proposed in Section 4.1.2 can be found in Appendix A. Some important aspects of the three kinds of auctions are listed below.

Flights (8 auctions). Available flights are infinite. The price follows a random walk and does not depend on other bidders, but rather only on the auctioneer. If a bid is higher than the asking price, the bidder always gets the flight. Further, since the price increase and decrease with equal probability, the expected change in price is 0.

Hotel Rooms (8 auctions). The auction for rooms is a standard English ascending auction Mth price, with M=16 [185]. In this auction all bids are raked and the 16 highest bids win the products at the 16th highest price. The final clear price depends on other buyer bids, so in a very competitive scenario prices could be very high. In this case, a bid higher than the asking price is not guaranteed to get a hotel room. Rather, depends on the behavior of other buyers. Further more, it is crucial to note that we do not know the outcome until after the auction is closed.

Entertainment Tickets (12 auctions). Entertainment tickets are sold using a continuous double auction [64] (see also Section 4.1.2). In this case the final clear price depends on the other buyer and on the other seller bids. If a bid beats the ask-quote and comes in before other buyer bids, success is certain. If a bid is less than the bid quote and comes in before other seller bids, success is certain.

Comments on the TAC game

TAC game represents an interesting benchmark and a challenging market for studying the behavior and performance of trading agents that have to compete with competitors developed by multiple suppliers. Like in a real market place, limited quantity of products are sold in parallel auctions and usually each auction of a different vendor, has its own rules and characteristics. Further, like in the real life, the utility of getting some products is related to conquering some other resources, so the trading agent's goal is to buy bundles of products. Still, in a real market place a buyer has to choose among various vendors where to buy substitutable products. Keeping on with the analogies, in electronic commerce, like in the TAC game, it is not easy to know the name or important properties of the others competitors. It is not possible to repeat the same game, with the same customers' preference and the same contestants. Furthermore each auction game is too short to extract reliable statistical information about the behavior of competitors. This makes hard to introduce learning techniques in a trading agent for TAC.

On the other way, there are also some unrealistic rules. The number of available flights is unlimited, so the corresponding auction is very similar to a shop with an expected change in price equal to zero. Another unrealistic characteristic is the absence of a penalty in the utility function, when the trading agent does not buy a feasible travel package for one of its customer. Finally in real electronic marketplaces it is more likely that various auctions start at different instant of time.

A limitation of this marketplace is that it is closed. In fact agents designers have to register their selves to the competition in order that their software agents can take part to the race. Real marketplaces in future have to be open in order to let artificial agents to interact with different markets.

4.3.2 Description of the Trading Agent Nidsia

The objective of this work is to study a new devised bidding policy for parallel auctions for several valuable bundles of products with possible overlap. The proposed approach is not restricted to this game.

Before to propose my solution to the problem, I had a look at the state of the art in this field. I found studies on auction protocols like for instance the works present in [135] which focus on using a bidding policy for a team of competing agents, and then study what social outcomes follow for the multiagent system at whole or which auction is better in which situation from the seller's point of view. I found one study on bidding strategies for sequential auctions the paper by Boutilier et al. [15], but I didn't find existing studies on bidding strategies for parallel auctions for resources that are complementary and substitutable.

In this game a trading agent knows the current ask and bid price (see Section 4.1.2) of the open auctions and it does not know nothing about other competitors: their name, preferences, bids or endowment. The two main problems that it has to tackle in playing in the TAC game are:

- The *acquisition problem*. At each time during a game the trading agent has to resolve the acquisition problem, that is, it has to choose its bidding action: where, how much and when to bid in the auctions. The objective of this research work is to study a *bidding policy*, that is a mapping from states into bidding actions, with the goal to maximize the agent's utility. This kind of problem is interesting because the trading agent does not know the independent value of the individual products, it do knows only the value of complete bundle of products that forms a feasible package. Moreover, the agent does not know the final assignment of the auctions, that is the effect of its actions, therefore the problem can be considered as a decision problem under uncertainty. These considerations lead us to construct an agent bidding policy, in which bids for any resource are conditioned by the possible outcome of the other bids. The proposed approach is inspired by value iteration [123], a standard stochastic dynamic programming algorithm, and by Boutilier, Goldszmidt and Sabata's paper on bidding policies in sequential auctions [15].

- The *allocation problem*. Given the set of products the trading agent already holds, it has to decide how to optimally allocate them to its eight customers. The solution of this problem is relevant during the bidding phase and at the end of the game. It can be proved that the problem of finding the allocation that guarantees the maximum utility for the agent, can be formalized as an integer linear programming problem; specifically, as a weighted set packing problem, that is NP-complete. Although only few actual instances of the problem lead to a significantly longer solution times (in only 1% of 1866 problem instances, taken before the finals, the optimization took 6 or more seconds) [158]. Two of the competitors, RoxyBot [76] and ATTac-2000 [158], implemented an optimal strategy algorithm for allocation, while the other TAC participants, like us, used some form of greedy strategy [77].

The allocation and acquisition problems are strictly connected; in fact it is necessary to use the allocation module to evaluate the goodness of every bidding action. It is possible to simplify the approach to these two problems facing them in a distributed manner. That is, do not try to compute one collective bidding policy that satisfies all the customers' requests at once. But try to compute a bidding policy for each customer independently, then the overall bidding policy is the sum of each isolated approach. The disadvantage of this strategy is in the loose of the centralized view of the problem. The agent looses the chance to exchange products among its own customers, during the game. The distributed approach scales better with the number of customers respect to the centralized approach and is simple to implement, for these reasons I decided to use it to compute the bidding policy of the trading agent, while I decide to greedy allocate the products at the end of the game in a centralized manner.

Nidsia Bidding Policy

Nidsia is the name given to the trading agent that implements the bidding policy that I devised. The bidding policy in the same for the entire game, while what really changes is the estimation of the probability to obtain the desired products. Some other competitors [157] instead decide to split their bidding strategy in two parts: one strategy for the first part of the game, with the only goal of keeping open the hotel auctions and another strategy for the final part of the game, in which all competitors start to bid seriously to get the products.

Nidsia runs the following inner bidding loop:

```
REPEAT
   Get current market prices and holdings
   For each customer
      Decide on what products to bid
UNTIL game over
Allocate products to clients
```

First it gets current market prices and holdings. Then it makes decision on what products to bid, it focuses on auctions for travel products (flights and hotels) that are crucial to create a feasible package. Therefore by the nature of the TAC auction mechanisms, its decisions are primarily concerned with hotel auctions. In fact the supply of flights is unlimited and their price are predictable, while the supply of hotel rooms is limited and their prices are unpredictable. In computing its bidding action Nidsia forecasts the behavior of the system for one step (as explained better below), assuming that the next state would be the final one. This sequence of operations is repeated until the end of the game when the awarded products are allocated to its customers.

The model of the problem

In my model the trading agent's state s_t is a bit vector that describes the agent's current holdings for a certain customer at time t.

A bidding policy is a function from states to actions, where an action a is a vector of bids, one per auction, and each bid is a price-quantity pair. There are 28 auctions for every game, each auction at time t has a given ask quote $q_{t,i}$. The number of possible bids (and therefore actions) is numerable infinite, if one considers all possible discrete values of price and quantity. To reduce the space of actions to a manageable size, I only consider bids in which the quantity is 1 and the price for auction i is the ask-quote $q_{t,i}$ at time t plus a fixed increment δ. With this simplification, an action a is a bit vector, where $a_i = 1$ if a bid is submitted at price $q_{t,i} + \delta$ and $a_i = 0$ if no bid is submitted.

The time needed to reach a new state, s_{t+1}, is not known, because it depends on the time needed by the auctioneer to compute another price quote or to terminate the auction and assign products to bidders. This asynchronism among the auctions makes impossible to forecast the behavior of the system during an entire game. Because of this, in computing the bidding policy the agent assumes, at every computation of the bidding policy, that the following state, s_{t+1}, will be the final one. Under these simplifying assumptions, Nidsia computes an optimal bidding policy.

One of the main problem in defining a bidding policy for a trading agent is to estimate the distribution of the probability of getting a good given a certain bid in the corresponding auction. One possible solution in real market place, adopted for example in [145] is to extract the distributions from the historical data. The drawback of this approach is that usually it is rare that winning bids in the future are similar to winning bids in the past. Another approach is to use a reasonable distribution of probability on the basis of designer's experience and I decide to follow this way.

The probability of obtaining item i is assumed to be near 0 at the beginning of the game and near 1 at the end of the game. Specifically, for the purposes of TAC game, these probabilities were given by the following equation for a straight line: $F(t) = mt + b$, that satisfies the condition: $F(1) = 0.1$ and $F(15) = 1$, that is $m = 0.9/14$, $b = 0.1 - m$ (15 seconds is the duration of a TAC game). For each good i if the agent has bid for it, the probability that the bit i of s_{t+1} is equal to 1 is $F(t)$ while the probability that the same bit is equal to 0 is $1 - F(t)$ (Formula 4.5).

$$p\ (s_{t+1,i} = k|s_{t,i}, a_{t,i}) = a_{t,i}\ [k\ F_i(t) + (1-k)\ (1 - F_i(t))] \tag{4.5}$$

The overall probability to reach the state s_{t+1} from the state s_t doing the action a, is equal to the product of the various probabilities for each auction (Formula 4.6). At this level the previous bids that are still active are taken in consideration. I assume that the probability distributions among the various auctions are independent.

$$p\ (s_{t+1}|s_t, a_t) = \prod_i\ p\ (s_{t+1,i}|s_{t,i}, a_{t,i}) \tag{4.6}$$

The utility $V(s_t)$ to be at state s_t is taken to be the reward $r(s_t)$ for being in state s_t less the cost $c(s_t)$ of obtaining the items held in that state (Formula 4.7). The cost $c(s_t) = \sum s_{t,i}c_{t,i}(h)$, where $c_{t,i}(h) = 0$ if Nidsia owns item i at time t, and $c_{t,i}(h) = q_{t,i}$ otherwise, for h representing hotel rooms. The reward $r(s_t)$ is taken to be the maximum possible value obtainable among all feasible packages that include the hotels indicated by bit vector s_t.

$$V(s_t) = r(s_t) - c(s_t) \tag{4.7}$$

The expected utility $E[U(s_t, a)]$ of taking action a in state s_t is the sum over all possible states s_{t+1} of the probability $P(s_{t+1}|s_t, a)$ of reaching state s_{t+1} times the utility $V(s_{t+1})$

of state s_{t+1} (Formula 4.8).

$$E[U(s_t, a_t)] = \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t)V(s_{t+1}) \tag{4.8}$$

For each customer and given current holdings, Nidsia computes the expected utility of each of 256 possible actions, corresponding to whether or not each of the 8 possible hotel rooms is included in the action or not. Then the trading agent bids according to the action that maximizes the expected utility.

$$a_t = \arg\max_j E[U(s_t, a_j)] \tag{4.9}$$

In some cases the expected utility of the best action could be near the expected utility of some other actions, the action that bids for the little number of products, the one with less risk, was chosen.

Allocation Strategy

As mention above Nidsia allocates its products to customers according to a fixed heuristic, rather than computing optimal allocations (using e.g. integer linear programming). One minute before the end of the game, Nidsia bids on flights that coincide with the hotel room auctions that it expects to win. Also at this time, the initial endowment of entertainment tickets is greedily allocated to customers. Unused tickets are auctioned off, and useful tickets currently on sale are purchased. At the end of the game, Nidsia checks that its customers have all the necessary products to complete their travel, and it heuristically tries to allocate any unused products so as to satisfy as many customers as possible.

4.3.3 Evaluation Phase, Experiments and Results

I decided to carry out some experiment to study Nidsia behavior in different situations. In particular I wanted to manipulate some factors that where not under our control in the actual competition, like for example the composition of the group of competitors.

Experiment 1. In the first experiment Nidsia competes against seven dummy agents. Dummy agents are provided by the TAC server. The behavior of a dummy agent is not random, but is based on a simple and fixed strategy. This scenario is the less competitive one. The results are that Nidsia wins almost always (see Table 4.1). The average utility of Nidsia is $2'111$, while the average utility of the best dummy agent is 792. The difference between the mean utilities is statistically significant. In 2-tailed t-test p, the probability that the two samples belong to the same population, is very close to zero, $p = 5.02\ 10^{-14}$.

Experiment 2. In the second experiment Nidsia competes against three instances of itself and against four dummy agents. This second scenario is more competitive than the previous one. The result is that the 4 Nidsia agents are placed almost

game n°	7845	7848	7856	7859	7863	7867	7868	7869	7870
1° Nidsia	3577	1673	3047	2420	2297	2226	2491	1757	2287
2° dummy	85	526.5	1993	360	1887	1910	356	589	1034
3° dummy	-1047	289.5	1957.5	-728	1526	1886	-807	-190.5	716.5
4° dummy	-2649.5	175.5	-150.5	-798	-1979	1820	-1269	-264.5	520.5
5° dummy	-3192	-3278	-2365	-1021	-2001	1394	-1979.5	-618.5	-1937
6° dummy	-3296.5	-3816	-3326.5	-1074.5	-2113	-1353.5	-3003	-2633	-2591.5
7° dummy	-3752	-5875.5	-3844.5	-2357	-4377	-3365	-4532.5	-2912.5	-3086
8° dummy	-4325	-5908	-4542	-4323.5	-4769	-3454.5	-5012	-3240	-5715.5
game n°	7884	7926	7936	7937	7941	7947		7847	**average**
1° Nidsia	1513	1140	2521	3504	2405	2252	**4° Nidsia**	-1330	2,111
2° dummy	1405	-1009	-792	1869	947	1309.5	1° dummy	207.5	792
3° dummy	618	-3909	-2079	1520	-2996	1307	2° dummy	-1042	-186
4° dummy	241	-4277	-4642	903	-3133	1002.5	3° dummy	-1174.5	-980
5° dummy	-2187	-4330	-4679	-397	-3526	-598	5° dummy	-1402.5	-2,007
6° dummy	-3233	-4819	-4956.5	-1381	-3776	-2625.5	6° dummy	-1511	-2,844
7° dummy	-3794	-5322	-5897.5	-4325	-3957	-4183	7° dummy	-4276	-4,116
8° dummy	-4084	-6089	-6317	-6339	-4907	-6960.5	8° dummy	-4442.5	-5,027

Table 4.1: Results of the games in Experiment 1

always in the first places in the race (see Table 4.2). The average utility of the first Nidsia agents is 2656. It is possible to notice that in these games there are less negative scores because the four Nidsia agents exploit and share the available resources better. In this experiment, like in the previous one, the difference between the mean utilities, of Nidsia and dummy sample, is statistically significant. 2-tailed t-test gives an extremely small $p = 6.71\ 10^{-15}$.

Experiment 3. In the third experiment I want to compare the performance of Nidsia agent with the performance of Nidsia2, a similar agent that computes a different probability of success for each auction. The idea is to exploit the information, about the current game, that is possible to extract from the on going of prices in the market. In Nidsia2, if an auction is very competitive, meaning that its price-quote rises very quickly, the probability of success is lower than in a less competitive auction. It results that Nidsia2 is a risk adverse agent. Precisely, for each item i the probability of obtaining it, is given by the following equation for an exponential function:

$$F'(t,i) = e^{-((\Delta price/\Delta time)/\Delta rate_{max})} F(t,i) \tag{4.10}$$

where

$$\Delta rate_{max} = 100 \tag{4.11}$$

The probability of failing to obtain an item at time t is $1 - F(t,i)$. In Figure 4.2 three different kinds of price trends are shown and in Figure 4.3 are shown the corresponding probability trends. In this experiment Nidsia competes against Nidsia2 and against 6 dummy agents. The two Nidsia agents are placed first and second in almost all the test games, but Nidsia2 does not always perform better than

game n°		7838		7846	7858	7864	7871	7873	7874	7880	7898	
Nidsia	**2°**	-122	**1°**	4095	2366	1986	3135	2640	2900	2802	2258	
Nidsia	**3°**	-183	**2°**	2676	2365	1316	2884	1606	2610	2247	1580	
Nidsia	**6°**	-3830	**3°**	2309	2204	-273	1843	1470	2022	1912	930	
Nidsia	**8°**	-6096	**4°**	2169	963	-1977	1445	1126	1258	1876	628	
dummy	1°	1488	5°	2010	-2567	-2545.5	555	-1248	-2542	-2288	-2976	
dummy	4°	-535	6°	664	-3521.5	-2688	-1425	-2655	-3517	-2473	-3799	
dummy	5°	-1479.5	7°	-182	-3730	-3086.5	-1791	-3255	-4231.5	-4124	-4829	
dummy	7°	-4446.5	8°	-1237	-5112.5	-3703	-2398	-3853	-4890.5	-6297	-6066	
game n°		7881	7899		7882		7883		7887		7890	**average**
Nidsia	**1°**	2453	3440	**1°**	3532	**1°**	3471	**1°**	2896	**1°**	1982	2656
Nidsia	**2°**	1739	1925	**2°**	2498	**2°**	3136	**3°**	1993	**2°**	1734	2008
Nidsia	**3°**	1541	221	**3°**	1536	**6°**	768	**5°**	1764	**4°**	1453	1058
Nidsia	**7°**	-1866	-2536	**5°**	-21	**7°**	754	**6°**	1120	**5°**	1232	5
dummy	4°	-459	104	4°	999	3°	2903	2°	2174	3°	1615	-185
dummy	5°	-603	-77	6°	-2398.5	4°	1713	4°	1822.5	6°	-420.5	-1328
dummy	6°	-1524	-501.5	7°	-2614.5	5°	1448	7°	959	7°	-2441.5	-2092
dummy	8°	-2881	-5648.5	8°	-3738	8°	-3196	8°	-2933.5	8°	-3767	-4011

Table 4.2: Results of the games in Experiment 2

Nidsia. A possible explanation is that the information that is possible to exploit during a single game, is not enough to have a more successful agent. The difference between the mean utilities of Nidsia and Nidsia2 sample, is not significant (2-tailed t-test, $p = 0.46$).

The TAC competition showed that Nidsia was able to produce a satisfactory performance, taking also into account that its bidding policy was thought to be adapt also to eventual changes in the rules of the game. Nidsia was eleventh in the semi-final TAC game competition of the twelve participants and was not admitted to the final phase of the eight best agents.

Comments on Nidsia behavior

The resulting behavior of Nidsia during one of the games is the following: during the initial part of the game, the agent bids in every auction, because the prices and the probability of success are very low. So it bids for substitutable products to be sure to get at least one of them. Then after some minutes, the agent starts to react in accordance with the way of the prices are changing and it begins to concentrate its bids on the more convenient products. In general Nidsia prefers short travel packages, because they have a much higher probability to be obtained entirely. Further it does not consider the utility of the entertainment tickets when it computes its bidding policy, so is not interested to remain longer to get the fun bonus.

During the entire game the probability distribution of success in the auctions is a crucial point. If the probability of success is underestimated, like in the first minutes of

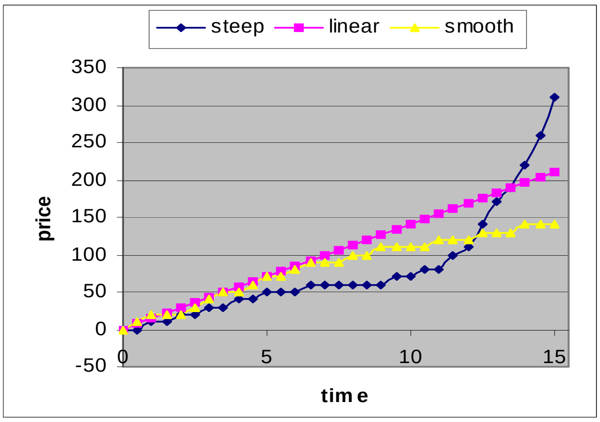

Figure 4.2: Different price trends

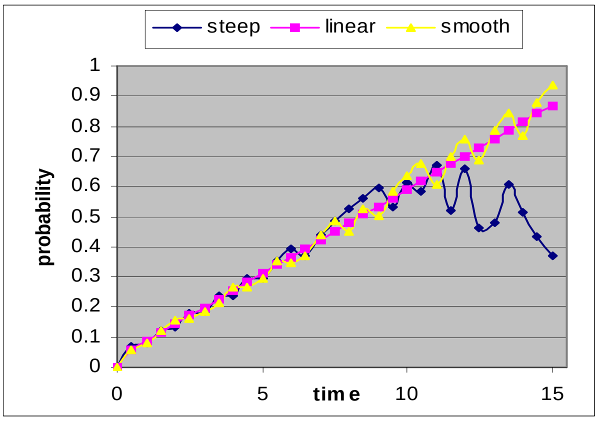

Figure 4.3: Different probability trends corresponding to the different price trends showed in Figure 4.2

the game, the agent bids in almost all auctions, to avoid to be left without some important products but it risks to buy much more products than it really needs. On the opposite side, if the probability of success is overestimated the agent is quite sure to get some products. It bids exactly on the products to form the best feasible packages and it does not take into account the risk of being left without necessary products.

A drawback of Nidsia's bidding policy is that when it decides (two minutes before the end of the game) on which flights to bid, relies on hypothesis that all its active bids will be successful. But if it overestimates the probability of success, especially in some high competitive game, it gets stuck in a situation where it has the flights, that usually are very expensive, but it does not have the necessary rooms to compose a feasible package. This is an important difference between Nidsia and other competitor's approach. In fact Nidsia also in the final phase bids for hotel rooms the value of the ask quote plus a certain fixed value δ, while some other competitors bid their marginal cost, that is the utility of the package that will be formed purchasing the hotel room (given that they have already bought flight tickets). Such agents usually made very high bids (1000) always winning the hotel rooms by paying less (M^{th} price auction). In the final competition a lot of trading agents, from observing the preliminary rounds, start to bid very high values with the consequence of many negative scores, because there was more than m high bids. In this situation only adaptive agents able to reduce the likelihood of bidding in highly demanded auctions would perform well. For example ATTac agent, the first one classified at the competition, has a mechanism to predict closing prices of the hotel auctions on the basis of closing prices of previous game and on the basis of the previous behavior of actual competitors. Even if such information was not available via the server's API ATTac obtains the name of other participants from TAC web page.

With regard to the allocation module, at the end of the game, Nidsia uses a heuristic strategy, although during the official competition it gets 95% of the optimal utility [157]. As concern the distributed approach to tackle the acquisition problem during the game. It is possible to notice that it is not very relevant, because in many games of the TAC competition, few products are bought during the first and middle part of the game, so there are few products to be redistributed among the customers.

Finally it is possible to notice that TAC is not a zero-sum game, but there is high competition for hotel room resources. In some games it is possible to notice that trading agents compete strongly on a special good, so the price rises and the utility average decreases. While in some other games there is less competition and the utility average is higher.

4.4 Conclusions

The problem of devising and studying bidding policies for buying various complementary and substitutable products sold in parallel auctions with different rules, is very important and crucial in order to be competitive in nowadays electronic market places. Bidding in different auctions is not an easy task for human being both because it takes a lot of time monitoring many different web sites where the prices are continuously changing in unpredictable way and because it is very complex taking into account numerous existing correlations among various commodities.

In particular my approach investigates how to apply the idea of maximizing the expected utility of each possible action to this kind of problems. It results that the crucial problem is how to estimate the probability to win a certain good given a certain bid value.

The bidding policy described in this chapter was not tailored to the particular auction mechanisms of TAC, but rather is more general in its applicability to combinatorial auctions of substitutable and complementary products. As a result, the implementation of the algorithm required some strong simplifying assumptions. Nonetheless, Nidsia's overall performance illustrates the promise of this general method even if more evidence that Nidsia performs well in other games is needed.

The most important characteristic of trading agents is adaptability as demonstrated by ATTac trading agent [158], the first-places finisher in TAC competition. In fact the environment is very rich of information, it is constantly changing, and highly uncertain. ATTac is flexible in order to be able to successfully cope with a wide variety of scenarios. It has two bidding policies: a passive one for the initial part of the game and an active one for the final part. It learns when to switch from passive to active mode, it learn when to use integer linear programming solution for allocation problem, it adapts its strategy to account for skyrocketing prices in the final phase of the game. Adaptability resulted very important during the competition because many designers changed their agent bidding policy for the final phase of the competition.

Recently the problem of designing bidding strategies is become much investigated. For example in [184] (2001) the design of flexible trading agents able to encompass with multiple types of auctions is presented. In [20] (2002) another decision procedures to bid across multiple simultaneous auctions that operate different protocols is proposed.

TAC game like any other currently existing market place is closed, in the sense that only agents designed for that server can interact with it. In particular TAC designer defined a finite set of messages (set of methods that form the application program interface API of the server) that every agent has to use to exchange information with the auctioneer: getting information on auctions and on personal endowments, sending bids, and getting auction results. Given this limitation a trading agent cannot actually bid in different market places to get a certain product at best price. Moreover the same agent, for example Nidsia, would become useless if the TAC server is changed as it happened during the evolution from TAC-00 to TAC-01. A very important solution to this problem consists on defining a standard way of interaction among agents, a standard semantic for agent communication languages, as discussed in Chapter 5.

Chapter 5

Agent Communication

As discussed in Chapter 3 the possibility for artificial agents to interact with each other is a crucial point for their successful application to real problems. They need to exchange information, knowledge and services with other agents, or perhaps with human, in order to be able to solve problems that cannot be solved alone. To perform such tasks agents need to be able to negotiate and to have articulated conversations, that is, complex tasks that go beyond the simple ability to exchange binary information.

In Chapter 4 application dependent interaction protocols were presented. The adoption of such protocols as way of interaction, like for example in the TAC game, presents some drawbacks from both the system and the agent point of view. It makes impossible to realize open dynamic interaction systems because there is not a standard way to define the social context of the interaction system, and there is not a standard way to define the meaning of the allowed messages in the system. Moreover it is impossible to implement agents able to interact with different systems because there is not a standard way of interaction adopted by all systems' designers. A solution to these problems can be found by defining a commonly accepted semantics of Agent Communication Languages (ACLs) with some fundamental properties that will be discussed in the following sections.

The topic of this chapter regards human and agent *communication*. In Section 5.1 mainly philosophical studies about human communication and Speech Acts Theory will be introduced. Such studies will be the basis for all subsequently studies made about agents communication. In Section 5.2 the main components of an Agent Communication Language will be analyzed. Furthermore principal problems and existing proposals for the definition of a standard semantics of Agent Communication Languages will be discussed. This chapter can be viewed as an introduction to the next one, where a new proposal for the definition of the semantics of Agent Communication Languages based on the notion of *social commitment* will be presented.

5.1 Human Communication

In this section I will try to synthesize existing ideas and studies about the meaning of the word "meaning", about how a language works, and about human linguistic communication.

In reaching this objective I was mainly inspired by Searle's book "*Mind, Language and Society*" [144] and by recent psychological studies on cognition. My subsequent purpose is then to use such schema, such formalization of human communication, to study artificial agents language and communication. Specifically to use such ideas to propose a new operational specification of the semantics of a commitment-based Agent Communication Language.

A formal study of a language has three aspects. The *syntax* that deals with how the symbols are structured; the *semantics* that deals with what the symbols denote and *pragmatics* that deals with how such symbols are interpreted and used. *Meaning* is a combination of semantics and pragmatics. That is a combination of the meaning of words as part of the specific language used and of the meaning that depends on the speaker mental states and on the environment in which he/she exists. The second one, the *speaker meaning*, is the type of meaning that I want to investigate. In order to understand which is the mechanism that we use to attribute a meaning to an acoustic event, we have to analyze how the human *mind* works.

Mental States

The essence of our minds is *consciousness* that is, the quality or state of being aware. Consciousness has a large number of forms and varieties that are called *mental states*. According to J. R. Searle's approach, mental states have some crucial characteristics.

The primary evolutionary role of the mind is to relate us to the world and to other people. *Intentionality* is the feature of the mind by which mental states are directed at, or are about, or refer to, or aim at objects and states of affairs other than themselves. It is the biological means used by the mind to understand, to intend and represent to itself the world. Note that the objects do not need actual to exist to be represented by our mental states. For example child can believe that Santa Claus will come at Christmas even if he does not exist.

To better understand the structure of mental states is necessary to make a distinction between the *content* of the state and the *type* of state that it is. For example we can hope that it will snow or we can be afraid that it will snow. Then we have to distinguish different types, which express how we feel the content. The content of a mental state is a proposition and thus has conditions of satisfaction.

The *conditions of satisfaction* of a mental state are its most important characteristic. An intentional state is *satisfied* if the world is the way it is represented by the intentional state as being. They are more general than truth conditions, because there are mental states like desires and intentions that cannot be simply true or false but can be satisfied, fulfilled, etc. If we want to know exactly what a person's intentional state is, we must ask ourselves under *what conditions* exactly would it be satisfied or not satisfied. This distinction between true conditions and other kind of conditions, for different type of mental states, leads us to realize that mental states have another feature: the *direction of fit*.

Different types of mental states relate their propositional content to the real world with different obligations of fitting. Some types of mental states, like for example beliefs or hypothesis are said to be true depending on whether the world is really as they represent it. For this reason these types of mental states have a *mind-to-world* direction of fit. Other type of mental states like for example desires and intentions are said to be satisfied depending on whether the world really becomes as it is represented. So to speak is the world that has to match itself to the content of the mental state, and so there is a *world-to-mind* direction of fit. While some other types of mental states like being glad or sorry have a *null* direction of fit. This terminology was invented by Austin in 1961 [5].

Speaker meaning

When someone emits an acoustic blast in a normal speech situation is said that has performed a *speech act*. Therefore my initial question becomes: "how can human beings attribute a meaning to a speech act?" and is now possible to conclude that speaker meaning is a form of derived intentionality: the original, or intrinsic intentionality of the speaker's thought is transferred to words. When someone utters a sentence and intends to mean it he/she wants to attribute specific condition of satisfaction to the utterance. For example if he/she asserts something, he/she attributes truth condition to the utterance and after that he/she is committed to its truth or to the realization of its condition of satisfaction .

The language is then the fundamental type of symbolic system that is used by human beings to express and to progressively enrich their consciousness and more in general their knowledge. In fact during human being's growing phase the mind and the language enrich each other until, for adult, the mind is linguistically structured.

With the term "knowledge" I intend to remark that our mental states do not function in isolation. In order that I can have a certain belief or desire for example to go to ski, I need also to have some other beliefs, for example that there is a place where I can ski, or in general a lot of other connected metal states. I need also to have a set of capacities, abilities, tendencies, habits, dispositions, taken-for-granted presuppositions and "know-how" that forms my knowledge, that enables me to cope with the world and that are called "background" by Searle in [141]. I think is important to mention those recent psychological studies, especially the theory of *situated cognition*, proposed by Clancey in 1997 [26]. This theory remarked that human knowledge cannot be described entirely using a language or a symbolic system. Human knowledge represents our ability to interact in an efficient way with the environment and with other people, mainly with the goal of survive and it is tightly influenced by our architecture and by the fact that we live in a certain world. Consequently our knowledge is very rich and articulated, it is deep-rooted with our body (embodiment) as proposed by Clark in 1997 [27] and it is the result of our progressively and continuously interaction and adaptation with the environment.

Human Linguistic Communication

I have so far talked about the speaker meaning, but usually the actual objective of a speaker is to *communicate* that meaning to a hearer. The following step of my argumentation is then the investigation of human linguistic communication.

In order that human beings decide to communicate they need to have first the feeling that they are doing something with other intentional beings, like communicate, second the capacity to "understand" each other, that is, they need to know that there are other entities which have the capacity to have mental states like their ones. Further they need the capacity to have mental states like: "we believe", "we intend", "we hope", "we know" and so on. Where intuitively "we believe" means that "I believe" and that this information is shared among the members of a group, in the sense that "I believe that other people know the same as me" and "I know that each member of the group have a mental state like mine". Following Searle approach [144] this characteristic of certain mental states to be common cannot be reduced to a collection of individual intentionality plus mutual beliefs, as described above, but it is an irreducible and primitive concept that is called *collective intentionality.*

Thanks to collective intentionality human being can assign to objects present in the world, a function that is not strictly related to their physical aspects, but that is a common believed function, like for example assigning the function of "money" to a certain piece of paper. This special kind of function assignment is called *status function.* The assignment of status function has the form 'X counts as Y in (context) C' and it is very powerful because it can be iterated turning Y terms into X terms of a new status assignment. To be successful the assignment of a status function requires an agreement among the members of a community and a language or at least a language-like capacity for symbolization.

Often in an assignment of a status function the X term itself is a speech act. Then the imposition of a conventional meaning on words as well as the imposition of the speaker meaning in the performance of a speech act are cases of the imposition of a status function. Then a very important question is now: "how human being reach an agreement on how to attribute a meaning to a speech act and then can use speech acts to communicate?".

In order that I can propose the formulation of my answer to this question I have to analyze better the mechanisms of communication. A speech act uttered in a conversational context with the *intention*, of the speaker, to *mean* and to communicate something, is called an *illocutionary act.* It is the unit of meaning in communication and this term was introduced by Austin in 1962 in the book "How to Do Things with Words" [6]. A speaker has successfully performed an illocutionary act when he/she says something, and means something by what he/she says and tries to communicate what he means to the hearer. When we communicate to people, we succeed in producing understanding in them by getting them to recognize our intention to produce that understanding [78]. Thus the intention to communicate is the intention that the hearer should recognize my meaning, that is, the hearer should recognize that my utterance should have certain conditions of satisfaction.

In conclusion to communicate is necessary to construct an agreement on the conditions of satisfaction of illocutionary acts that goes behind the literal meaning of the words.

The process of the construction of an agreement on the conditions of satisfaction of illocutionary acts depends deeply on the common knowledge of the two interlocutors and on their relationships. The construction of the common knowledge among human beings is a very articulated and continuous process and it is usually rooted in a common culture as stated by recent psychological studies [106]. Then usually human beings construct such common knowledge and outline their relationship through communication. Therefore in order to have a successfully communication we need such common knowledge and we construct such common knowledge through communication. From a psychological point of view the communication process can be described exactly as this cooperative process of progressively building and increasing the portion of common knowledge [105]. In spite of communication needs cooperation to take place its goal is usually self-interested, the purpose of the speaker is to modify the mental states of his/her interlocutors and consequently to modify their behavior.

As will be widely discussed in the remaining parts of this chapter, actual artificial agents are obviously not able to create a common agreement on the meaning of speech acts, that is, on their conditions of satisfaction. Such agreement has to be reached among their designers and many researchers are looking for the definition of a commonly accepted semantics of Agent Communication Languages. Given that conditions of satisfaction are strictly connected to direction of fit next step will be to look for regularities in the set of all possible illocutionary acts.

5.1.1 Speech Act Theory

Speech act theory is a theory of language and communication that has pointed out some very important characteristics of communication. Its foundations can be found in "How to Do Things with Words" by Austin [6] and in "Speech Acts: An Essay in the Philosophy of Language" by Searle [139].

Since speaker meaning is a form of derived intentionality, like mental states, illocutionary acts may have a content, a type and a direction of fit. It is very hard to find a way to schematize the various propositional contents. They can be infinitely diverse and their conditions of satisfaction are related to the meaning of the words used as elements of a common language, and on the speaker meaning, which is very articulated because depends on the actual speaker intentions, on the common knowledge among the interlocutors and on their relationship.

However as discussed in the previous section our mind intentionality is limited, in fact our mental states can have three different directions of fit, from mind-to-world, from world-to-mind and null direction of it. Then following this schematization there must be a limited number of things that we can do with a language. Of course for illocutionary acts I shall speak about word-to-world or world-to-word direction of fit. Departing from this observation an interesting and reasonable question is now "how many different types of illocutionary act are there?".

Preconditions and Effects

Before to answer to this question it is important to outline better the properties that I want to use to univocally identify each type of illocutionary act. When we perform an illocutionary act we are trying to impose to it a type of status function, so we can try to express for each type of illocutionary act for what it "count as". Following speech act theory, the performance of an illocutionary act "count as" the performance of an action in the world. Then to better describe a type of illocutionary act we can try to delineate the corresponding action. The method historically used in artificial intelligence to describe an action is through its *preconditions* and its *effects* on the world and I will adopt it.

It is important to remark that when I speak about the effects of an illocutionary act I do not mean the *perlocutionary* effects on the hearer, but I mean the effects on the *social relationship* between the speaker and the hearer. The perlocutionary effect of a speech act, following Austin's terminology, have to do with the consequences on the hearer, and such effects can also be obtained without intending to do so. For example I can convince someone to do a certain action or to believe something without the intention to do it. Since the speaker can lie, the preconditions and the effects of the utterance of an illocutionary act, on the relationship between the interlocutors cannot be related to their mental states. In fact if, in order to assert something, the speaker has to believe its content or if the effects of the performance of an assertive acts are that the hearer has to believe its content, it is impossible to treat situations when, for example, the speaker lies.

A reasonable proposal can be that the preconditions of an illocutionary act are the conditions in order that it can be performed successfully and its effects can take place. They are usually conditions on the *role* of the speaker and on the context. The role depends on his/her position in the society and on his/her relationship with the hearer. The context depends on the actual state of the environment, usually some particular speech act has to be uttered before the current one.

While to take into account the effects of the utterance of an illocutionary act on the social relationship between the interlocutors we need something external from their mental states and that involves both parties. A reasonable proposal is that the effects of an illocutionary act are to create new *commitments* between the speaker and the hearer.

Taxonomy of Illocutionary Acts

Following Searle's taxonomy, there are only five different types of illocutionary acts.

1. The word-to-world direction of fit is characteristic of *assertives*. In fact the assertion has to reflect the actual state of affair and then assertives can be true or false. Examples of assertives are statements, description, classifications, and explanations. The mental state that is represented by an assertive is a belief. The performance of an assertion does not have specific preconditions and performing an assertive act the speaker commits himself to the truth of the content, that is, of the proposition.

2. The world-to-mind, or better world-to-word direction of fit, is characteristic of *di-*

rectives and *commissives*. In both this types of speech acts is the world that has to match itself to the words, but for directives is the hearer that takes the obligation to act in this direction, and for commissives is the speaker that takes the obligation to proceed. So it is better to distinguish between the hearer-based world-to-word direction of fit and the speaker-based world-to-word direction of fit.

Examples of directives are orders, commands, requests and questions. The mental state that is usually represented by directives is the desire. Directives cannot be true or false, but they can be obeyed, disobeyed, complied, answered, and so on. A precondition to successfully perform a directive act is that its propositional content must refer to the future. Performing a directive act the speaker is trying to get the hearer to behave in such a way as to make his behaviour match the propositional content of the directive. So its effect on the relationship among the interlocutors is to make the hearer aware of the speaker's directive. In the definition of a new artificial Agent Communication Language I shall formalize this kind of effect using the concept of *precommitment*.

3. *Commissives* as discussed above have a speaker-based world-to-word direction of fit. Examples of commissives are promises, vows, pledges, contracts, and guarantees. A threat is also a commissive but the difference is that it is against the interest of the hearer. The mental state that is usually expressed through commissives is the intention to do something. Commissives cannot be true or false, but they can be carried out, kept, or broken. A precondition to successfully perform a commissive act is that its propositional content must refer to the future. The effect of a commissive act is to commit the speaker to undertake the course of actions represented in the propositional content.

4. The null direction of fit is characteristic of *expressives* because the truth of their propositional content is simply taken for granted. If I apologize for hitting you I presuppose that previously I hit you. Examples of expressives are apologies, thanks, congratulations, welcomes and condolences. Expressives are usually related to a feeling of the speaker and cannot be true or false, they can be sincere or insincere and their effect is simply to express a feeling. They are very important for the construction of the social relationship among human beings but they are less important for artificial agents, which do not need to express their feelings.

5. *Declarations* are a very special type of illocutionary acts. The language creates the possibility, that mental states do not have, to combine both directions of fit. We cannot create a state of affair only thinking about it, but we can create a state of affair relative to institutional reality by way of successfully performing a declaration. Examples of declarations are "I pronounce you husband and wife" or "war is hereby declared" and so on. Declarations change the state of the institutional world, and thus they achieve the world-to-word direction of fit, and the changes happen by representing it as having been changed, so they have the word-to-world direction of

fit. The preconditions to successfully perform a declaration are that the speaker has the *authority* to perform the declaration and that the *context* is appropriate. The effect of declaration is to bring about a change in the institutional reality.

It is important to observe that even thought we have well characterized the five types of illocutionary acts, it is not easy in human communication given the illocutionary act to single its type out.

Speech Act Theory is very important for artificial agents studies because its view of language use as a form of action, makes possible to treat speech acts and other type of actions in a uniform way. Moreover given that it provides an account of human communication, its can also be used as a basis to deal with communication between artificial and human agents. Finally the identification of regularities in the set of all possible illocutionary acts makes possible to define the meaning of only a limited number of illocutionary acts without loosing generality. Thanks to these important aspects all existing proposals to define the semantics of Agent Communication Languages (see Section 5.2.3) are based on Speech Act Theory.

5.2 Agent Communication Languages

As discussed in previous chapters the possibility for artificial agents to exchange information, knowledge, and services is very important. At the beginning of the nineties, the recognized importance of knowledge sharing led to the creation of the Knowledge Sharing Effort by DARPA (the Defence Advanced Research Projects Agency of the US Department of Defence) [112, 117]. The central concept of the KSE was that knowledge sharing requires communication, which in turn requires a common language. In fact simple remote procedure call or remote method invocation are not sufficient to cover the expressiveness required in many applications, where usually messages have to be used to describe desired states in a declarative form. For these reasons the definition of a suitable and commonly accepted Agent Communication Language has been considered as a key step for the development of a truly operative multiagent systems [151, 100].

There are some fundamental requirements that every proposal for the semantics of ACLs has to take into account as will be discussed in the following sections. The most important one is that languages have to be used in nonproprietary multiagent applications. These kinds of applications are typically competitive and open, which means that autonomous and heterogeneous agents from different vendors have to be allowed to dynamically interact. Consequently a commonly accepted Agent Communication Language has to be strongly independent of the internal structure and reasoning strategies of the interacting agents.

The definition of an Agent Communication Language (ACL) includes the definition of the *syntax*, that is, the way in which single words are put together and the definition of the *semantics*, that is, the meaning of the communicative acts. As discussed in Section 5.1 about human communication, the meaning of communicative acts is a combination of the meaning of the words used, that is given by the *lexicon* as discussed in Section 5.2.2, and of the speaker meaning, that is, the illocutionary force of the act as discussed in Section 5.2.3. The lexicon of a language represents what is called in agents studies the *content languages* that is not completely independent respect to the illocutionary force of various speech acts. In a formalization of speech acts defining the boundaries and the interactions between these two components is not a simple task.

5.2.1 Syntax

An important component of a language is its syntax. The syntax describes the way in which the words of the language can be put together to form an utterance. It is possible to distinguish between *abstract syntax* and *concrete syntax.* The abstract syntax describes at a high level the structure of the utterances and depends on the complexity that the language presents at the semantic level. The concrete syntax is the actual implemented syntax. Given the abstract syntax there may be many different ways to render it in a concrete syntax. What is crucial is that different concrete syntax, that come from the same abstract syntax, have to be inter-translatable.

Abstract Syntax

A reasonable abstract syntax for the Agent Communication Language proposed in next chapter has to include the following parameters:

- Sender: the univocal identifier of agent that utters the message;
- Receiver: the univocal identifier of agent to whom is directed the utterance of the message;
- Content: the topic of the message;
- Force: the illocutionary force of the messages among the possible, for example: assertion, request, promise, etc;
- Language: the concrete syntax adopted;
- Ontology: the dictionary of terms and relationships between terms adopted;

There are other and more formal ways to define the abstract syntax of a language. For example the syntax can be defined using an algebraic formalization: a message is an element of the Cartesian product among six sets; or it is possible to define the syntax through a logic formalization: a message is a proposition made by a six arguments predicate where the arguments belong to different domains.

Concrete Syntax

Given an abstract syntax is possible to define many different concrete syntax. It is important that the various concrete syntax defined are all inter-translatable because it is not an easy task convincing all designers to use the same language, in fact each one has its own supporters.

The Foundation for Intelligent Physical Agents (FIPA), an international organization that is dedicated to promote standards in the field of agents proposes three different representation forms for Agent Communication Language messages.

- The "String Specification" that is very similar to an abstract syntax. A message is a sequence of strings, that is:

  $$message ='' ('' \; MessageType, \; MessageSlot^{\star} \; '')''$$

  Where "MessageType" is one of the Communicative Acts available in FIPA Library. "MessageSlot" includes: sender, receiver, content, reply-with, reply-by, in-reply-to, reply-to, language, encoding, ontology, protocol, conversation-id, UserDefineSlot.

- The "XML Specification" that is defined using the eXtensible Markup Language (XML) a language for the specification of the structure and data of documents on the Web.

- The "Bit-Efficient Specification" that is a bit-efficient encoding of the abstract syntax.

5.2.2 Lexicon

Assuming that the syntax of the language is given, it is necessary to define the *content language*, that is, the vocabulary with the words and their meaning that can be used by agents to interpret the messages exchanged. The term lexicon comes from the philosophical tradition and is equivalent to notion of dictionary.

A dictionary includes a set of *symbols* and their definitions plus a definition of their *relations*. In logic such relations are expressed giving a set of axioms, while in Artificial Intelligence in the last ten years such relations are expressed using what are called *ontologies*. Traditionally the term *ontology* refers to the science of being. However in computer science and other disciplines ontologies are interpreted as formalized dictionary of terms and relationships between terms and in some cases they can be interpreted and used by software systems, specifically agents.

Nowadays there are some working groups that are trying to define standard ontologies. Examples of these groups are: Standard Upper Ontology (SUO) Working Group inside the IEEE a leading authority in technical areas [79] and the group at Cycorp a firm that is the leading supplier of formalized common sense [37]. An *upper ontology* sometimes also called *top ontology* is an ontology limited to abstract and philosophical concepts, which in general are enough to address a broad range of domain areas. Concepts specific to a certain domain could be defined in *domain specific ontologies* that can be constructed using the standard structure provided in the upper ontology.

FIPA, the Foundation for Intelligent Physical Agents [60] proposes different representations for the content language adopted by FIPA ACL, that is, the standard agent communication language, based on agents' mental states, proposed by FIPA. All these content languages express, with the chosen formality, the objects, the propositions, the relations, and the functions available in the language.

- One possible content language is the FIPA Semantic Language (SL). A possible content expression of this content language can be: a proposition, that is a well-formed formula and which may be assigned a truth value in a given context; an action that can be performed as single action or in sequence or in alternative to other actions; an identifying reference expression (IRE), that identifies an object in the domain. Other valid content expressions may result from the composition of these basic cases.

- Another possible content language is the FIPA Constraint Choice Language (CCL). This content language is based on the representation of choice problems as Constraint Satisfaction Problems (CSPs) and supports: problem representation, information gathering, information fusion, and access to problem solution techniques.

- A description of another FIPA content language is based on the Knowledge Interchange Format (KIF) [69]. In KIF the universe of discourse, that is the set of all objects hypothesized to exist in the world, changes on the basis of the users, but every universe of discourse is required to include certain basic objects that are: all numbers, all ASCII characters, all finite strings, words, all finite lists of objects in the universe of discourse, a special object to treat the cases when a function is applied to arguments for which the function makes no sense.

- Another description of a FIPA content language may be based on the Resource Description Framework (RDF). The main strengths of the RDF language are its extensibility, reusability, simplicity and the fact that it is a standard for web applications. The RDF model proposes the eXtensible Markup Language (XML) as an encoding syntax. The RDF content language covers the definition of objects, propositions, and actions. In fact although FIPA does not require that a content language is able to represent actions, a lot of communicative acts require actions in their content and RDF schema can be extended to express them.

5.2.3 Semantics: Different Approaches

As already discussed in Section 5.1.1 all existing proposals for the definition of a commonly accepted semantics of Agent Communication Languages are based on Speech Act Theory. In particular on the proposed taxonomy of speech acts and on the view of language as a form of action. While deep and powerful, however, Speech Act Theory is not a formal theory of communication, and therefore cannot be directly used to implement communicative protocols among artificial agents. Moreover, the full theory is probably too complex to be completely formalized and it is possible to follow different approaches to transform at least part of it into a formal framework for the definition of the semantics of an ACL. In literature it is possible to identify mainly two approaches that differ in the method used to define the illocutionary force of basic communicative acts: the mentalistic approach and the commitment-based approach.

Mentalistic approach

The first approach, that can be called *mentalistic*, defines the meaning of speech acts using agent's *mental states*, like beliefs, desires and intentions. A complete treatment of the belief-desire-intention model of agency can be found in Section 2.3 and in [178]. Within the mentalistic approach, given a set of communicative acts deemed fundamental, the meaning of each act is expressed by specifying preconditions and intended post-conditions. Preconditions are expressed as the mental states that an agent must entertain before the emission. Post-conditions are expressed as the mental states that the emitting agent intends to achieve.

Concrete examples of mentalistic semantics have been proposed for Knowledge Query Manipulation Language (KQML) [80, 57], and for FIPA ACL [59]. FIPA ACL is a language proposed in 1997 by the Foundation for Intelligent Physical Agent (FIPA) an international organization dedicated to promote standards in the field of agents. The latest delivered specification of FIPA ACL can be found in FIPA web site [60].

For example in KQML the semantics of the *tell* act performed by sender a and receiver b is expressed through the following preconditions that indicate the necessary states for agent a to send the message and for agent b to accept it successfully, and the post-conditions that describe the states of sender a after the successful utterance of the message and of the receiver b after the receipt and processing of the message [99]:

```
tell(a,b,X)
preconditions(a): believe(a,X) and know(a,want(b,know(b,S)))
preconditions(b): intend(b,know(b,S))
    where S may be any of believe(b,X) or not(believe(b,X))
Post-conditions(a): know(a,know(b,believe(a,X)))
Post-conditions(b): know(b,believe(a,X))
```

Over the past few years a multitude of applications and systems have been built around ACLs. The following implementations are well-known examples of systems that use variants of KQML as their ACL. The more representative of the various approaches and trends are the following. Infosleuth [114], that emphasizes the semantic integration of heterogeneous information in an open dynamic environment. Knowledgeable Agent-Oriented System (KAoS) [16] is a Boeing project aimed at providing an infrastructure for the development of agents that support specialized suites of interactions. Infomaster [70] is an information integration system. Java Agent Template Lite (JATLite) [93] is a package of Java programs, developed at Stanford, which allows users to create communicating agents quickly. The Java-based Agent Framework for Multi-Agent Systems [23] is a set of classes that support implementing communicating agents in Java. Finally Jackal [36], is another Java package that allows applications written in Java to communicate via an ACL. The best known system that uses FIPA ACL is JADE [9].

Social or commitment-based approach

The mentalistic approach discussed so far presents some problems. First of all it is easier to adapt such specification to a particular agent theory (Beliefs-Desires-Intentions agents), while it may be harder to apply it to other type of agents. Second, using mental states to define the meaning of speech acts may be adequate in cooperative multiagent systems, but it is problematic when the multiagent system is composed by competitive, heterogeneous agents made by different vendors [151], like for example in electronic commerce applications. In this kind of context it is impossible to trust other agents completely or to make strong assumptions about their internal way of reasoning. In competitive applications it is crucial to have an *objective* and *external* method to test compliance of agents to a specific ACL semantics and the usage of agent mental states does not seem to satisfy these requirements.

Recently, a similar position has been adopted by FIPA. In October 2001, FIPA has appointed a new Technical Committee to promote the development of a new "Semantic Framework". The declared workplan of this committee is the following [118]: "This workplan is concerned with reframing the semantic framework for FIPA to reflect the needs of verifiability and conformance. In particular, the objective is to adopt or define a semantic framework that can give an account of FIPA's existing communicative acts and interaction protocols as well as a number of additional constructs such as contracts, agreements, policies, trust, agent descriptions and so on. ... It is envisaged that any new semantic framework would focus on publicly visible behavior of agents and systems; not on internal mental models." Relevant proposals for a new social approach to agent communication have been considered by FIPA during a meeting in Lausanne in February, 2002, among them our proposal widely described in the following chapter and Andrew Jones's proposal [94].

An interesting solution to these problems, as discussed in Section 5.1, is to take into account the objective social consequences and new obligations of making a speech acts. The effects of an illocutionary act can be described as the effects, on the *social relationship* between the speaker and the hearer, of the performance of the speech act. To formalize those effects is necessary something external from their mental states that involves both parties. A reasonable proposal is that the effect of an illocutionary act is to create a new object: the *social commitment* between the speaker and the hearer.

Following this idea in the *social* or *commitment-based* approach the meaning of a speech act is expressed using commitments directed from one agent to another [21]. Formal proposals to treat speech acts in terms of commitments can be found in [165, 30, 152, 32, 189, 61]. Most of current proposals focus on formal logical definitions, while the operational proposal presented in the next chapter appears to be closer to practical applications. For instance, it has been adopted to support effective communication between agents interacting through a supply chain network [166]. Supply chains are ways of connecting business unit, they are becoming more and more global and are particularly interesting for electronic-business applications.

Other approaches

Besides the mentalistic and the commitment-based approaches, some other proposals can be found in the literature. One example is protocol-based semantics [120, 81], in which the meaning of a communicative act is specified in terms of the responses that are allowed at each state of a conversation. However, this approach makes the meaning of communicative acts relative to a set of conversation protocols: by themselves, communicative acts become meaningless. This makes it extremely difficult to define a standard ACL as independent of specific application domains.

5.3 Conclusions

In this chapter philosophical studies about human communication and language have been presented. Such studies result in Speech Act Theory a very important theory of communication on which all-existing proposals for the semantics of agent communication languages are based. Subsequently syntax, lexicon, and semantics that is, the main components of an artificial language have been discussed. In particular advantages and limitations of existing proposals for the definition of the semantics of a standard ACL have been presented. On the basis of the most important requirements for an ACL emerged from this analysis, in next chapter a new proposal for the operational specification of a commitment-based Agent Communication Language will be presented.

Chapter 6

A Commitment-Based Agent Communication Language

As discussed so far, in the last years there has been a growing necessity to develop real *open* and *dynamic* systems, where multiple autonomous, self-interested, and heterogeneous agents can interact. Autonomous means that the agent must have few constrains on its behavior, so that it can fulfill its own goals. In particular it has not to be forced to do a specific action in a particular instant of the interaction. Self-interested means that agents from different constructors have the goal of maximizing their own utility. Heterogeneous means that agents are designed and developed by different vendors, and thus have different internal structures.

To be able to really implement such type of applications is necessary to define *standard way of interaction*, in order that the same agent will be able to interact with different systems, and to make the *social context* of the system explicit, in order that different agents might be able to dynamically enter and leave different systems with different rules, roles, obligations, and permissions.

The necessity to define a new standard semantics for Agent Communication Languages has been confirmed by FIPA Technical Committee on Semantics. Such Committee has recently provided a set of example uses cases deemed very important for the actual development of multiagent system applications. In particular they consider crucial the definition of a notation or a set of notations to express the concepts involved in inter-agent interactions. Examples of such concepts are: individual communicative acts, ontologies, conversations, institution, authority, contracts, agreements, policies, and shared plans. An important requirement for such semantic framework is that it has to support verifiable compliance and accountability. Moreover they observe that: "this requires an approach that is focused on externally observable events as opposed to the internal mental states of agents" [119, p.1].

The focus of this chapter is on the operational definition of a standard way of interaction, an Agent Communication Language, that abides by the following requirements considered most important in the literature [107, 151, 100, 152, 35].

- The proposed semantics has to be *verifiable*. As regard as this property, it is possible to identify different interpretations for this term. One meaning is related to the possibility to check if agents' communicative acts conform to the semantics of the ACL, that is, to the meaning of the exchanged messages. For example if the communicative acts of an agent are contradictory then the agent is not respecting the semantics. On the basis of how the semantics is expressed such a verification process could be computational complex or even non-decidable [176]. Another meaning correspond to the possibility to check if an agent behavior satisfies the constrains for the successful performance of the communicative acts belonging to the language. Moreover verifiable can be interpreted as the capacity to determine if an agent communicative acts are coherent with its behavior or with the artificial reality represented in the system. For example to check if an agent satisfies or not its promises or if its assertions are true or false. The meaning of the term "verifiable" can be further extended if used referring to interaction protocols, as will be better discussed in Chapter 7. In fact it is possible to verify if a given specification of a protocol is sound, and this kind of verification can be made statically, and it is possible to verify if an agent is following the given protocol, and this can be made only during an actual interaction [62].

- The meaning of the exchanged communicative acts has to be expressed in an *objective*, *external*, and *public* way. Objective in order that everybody attributes the same meaning to the messages, external respect to the agent internal structure, and public so that any third part agent witness of the messages flow has to be able to draw similar inference from the interaction.

- The proposed semantics has to be *flexible* and *extensible* to let agent to cope with various new situations.

- The proposed semantics has to be *simple* to be correctly used by agent designers, and enough *expressive*.

The requirement of verifiability to a common agreed languages, or protocol, or set of rules is crucial. In fact, the possibility for an agent to check if other agents' behavior is fear creates an *expectation* about their behavior and then enable the agent to reason about the future actions of other agents. This is a very important aspect in order that interactions among artificial agents can actually take place.

In this chapter I will present and discuss a proposal for the definition of the semantics of an Agent Communication Language, which fulfills the requirements specified above and the ones discussed in the previous chapter. Our proposal is situated within the social approach (see Section 5.2.3), and is based on an analysis of the primitive notion of commitment. In particular we analyze the evolution of commitments through time, from pre-commitment or conditional commitment to active commitment, then to fulfilled or violated commitment. In particular in Section 6.1 the mains concepts on which our approach is based are presented. In Section 6.2 a technical specification of the main components of an interaction system necessary to define the meaning of communicative acts based on social commitments is given. While in Section 6.3 the notion of social commitment previously described will be used to define the meaning of a wide class of speech acts using a homogeneous framework. In Section 6.4 important use of the proposed semantics are presented. Finally in Section 6.5 some examples are reported to show the dynamic evolution of the state of an interaction system when some of the defined communicative acts are performed.

The content of this chapter is mainly based on the papers: "Operational Specification of a Commitment-Based Agent Communication Language" [61] and "Defining Interaction Protocols using a Commitment-based Agent Communication Language" [62] by Fornara and Colombetti.

6.1 Main Concepts

Our approach to the definition of the semantics of an Agent Communication Language is situated within the social approach as described in the previous chapter. Following Speech Act Theory, sending a message to an agent counts as the performance of an institutional action [144, 35], called an "illocutionary act". That is an action that changes the institutional reality resulting from the common agreement among agents, or to be precise of their designers. Then to define the meaning of an illocutionary act we decide to describe the *effects* that the performance of the illocutionary act has on the social relationship between the speaker and the hearer and the *pre-conditions* that have to hold in order that such effects will take place.

6.1.1 Social Commitment

A crucial point of our approach is the choice to describe the effects of the performance of illocutionary acts using an "objects" that is *unambiguous*, *objective*, and *observable* by other agents and by any super-ordinate entity: the social commitment [21, 66]. The notion of social commitment is different from the notion of internal commitments used by agents to treat intentions and goals as discussed for example in [88]. Following this approach the stage of the conversation among agents as a result of previously exchanged messages can be described through a set of commitment objects.

We conceive commitments as "institutional objects" [33], i.e., an external object that can be created and manipulated by agents according to a set of rules on which there is a common agreement inside a community of agents. It is considered as a primitive social concept and cannot be defined in terms of agent mental states. The function of commitments is to *stabilize* the social relationships among agents, by making the behavior of other agents predictable, to certain extent. In fact agents that abide by the defined meaning of the exchanged messages will behave not in a deterministic way but in every instant the set of their possible actions depends on the previous exchanged messages. For example if agent a promise to agent b to carry out a certain task, if agent a abides by the meaning of the promise act performed, its future rational behavior is to some extent predictable.

Intuitively a social commitment object represents a commitment made by an agent (the debtor), relative to another agent (the creditor), that some fact holds or some action will be carried out (the content). We decided to propose an operational specification of a commitment-based ACL because we believe that the semantic definition of an ACL should be primarily directed to the community of software professionals actually involved in the design of real applications. The notion of social commitment has been formalized introducing an abstract data type: the *commitment class*, with the description of its characteristics, its structure, its dynamic evolution in time, and the methods available for its manipulation. An instance of this class will be called a *commitment object* (see Section 6.2.1). In particular the content of a commitment is represented by a formal proposition that may take different truth values and is relative to a certain time period, during which the fact is due to hold or the action is due to be carried out. We formalize this concept as a *temporal proposition* (see Section 6.1.2).

The dynamic evolution in time of a social commitment is traced using the *state* of the commitment. A commitment is *active* when it has not yet been fulfilled or violated; this is the case when the truth-value of the commitment's content is still undefined. An active commitment may become *fulfilled* (when its content becomes true) or *violated* (when its content becomes false). Finally a commitment may be *cancelled*, and this simply means that it does not exist any more. The creation and manipulation of commitments are based on proper methods. Operations, like the transfer of a commitment to another debtor or another creditor, are possible, but for the moment they will not be dealt. Furthermore some event driven routines are used to automatically update the state of commitment objects.

Conditional Commitment

The concept described so far is sufficient to account for basic situations. However in many applications, like for example in electronic commerce, agents need to make commitments not in absolute terms, but under given conditions. For example an agent may commit to paying a sum of money to another agent only after it has received a particular product. Conditional commitment expresses the idea that the debtor is committed to a certain proposition ψ if a certain condition φ holds, while if the condition is false the debtor is no longer committed to ψ. We think that the notion of *conditional commitment* can be considered as the basic notion in our formalization, while the notion of a commitment without any condition can be considered as a special case.

The concept of commitment can be formalized using deontic logic [169, 1], i.e. the study of the logical relationships among propositions that asserts that certain actions or states of affairs are obligatory, permissible, right or wrong. There are at least three different alternatives for the formalization of commitments using deontic logic as listed below.

- One possible approach is to define a monadic deontic operator C to express commitments, and use propositional logic to express the content and the condition of commitments. Conditional commitment can then be expressed with the utterance $C(\varphi \rightarrow \psi)$ with φ that represents the condition of the commitment and ψ that represents its content. This formalization presents the problem that when φ is false, $\varphi \rightarrow \psi$ is true, then the debtor is committed to something that is true; but this does not coincide to the intended meaning of conditional commitment.

- Another approach proposed by Yolum and Singh in [189] could be to define a monadic deontic operator C to express commitments, and use an extension of propositional logic to express the content and the condition of commitments. This extension contains a new operator "$\leadsto$" that according to the authors has to be false when the antecedent is false and has to be true only when the antecedent and the consequent are true. Conditional commitment can then be expressed with the utterance $C(\varphi \leadsto \psi)$. But this operator cannot be a Boolean operator otherwise it simply coincides with conjunction. Possibly "$\leadsto$" could be defined as a modal conditional operator [25], however the authors do not develop a complete proposal in this direction.

- The third approach, from which our operational formalization is derived, is inspired by existing formalization of conditional obligations in deontic logic [170]. It defines a dyadic deontic operator to express commitments, and use propositional logic to express the content and the condition of commitments. Conditional commitment can then be expressed with the statement $C(\varphi \mid \psi)$. This choice to separate the content and the condition may creates some problems when is necessary to formalize particular temporal dependence between them.

Therefore to manage the possibility to have commitment with a condition we introduced a further field in our formalization of commitments: the condition. The state of a commitment with a condition that has to become true in order that the commitment becomes active is called *pending*.

Pre-commitment

Using commitment objects as defined so far, it is possible to express the meaning of various speech acts like *assertions* and *promises*; however, it is not possible to express the meaning of directive speech acts, like *requests* and *questions*. When an agent requests another agent to do something, it is trying to induce the other agent to make a commitment. In fact usually when the context of the interaction does not define a relationship of authority between the interacting agents, it is impossible for an agent to create a commitment of which some other agent is the debtor; in other words, one cannot directly commit somebody else. To solve this problem it is important to notice that by making a request, an agent sets up a situation in which the hearer can make a commitment just by *accepting* the request. For example with the request "Can you bring this book to John?" the speaker tries to induce the hearer to commit to the future performance of an action. After receiving the request, the hearer can create such a commitment just by saying "Yes, sure". This example shows that a request creates a particular social situation between the speaker and the hearer, a pre-commitment. Some requests can also be conditional, for example "Would you please close the window as soon as it rains?" in this case the commitment object has also a nonempty condition field.

To manage the situation when a commitment has a debtor, a creditor, a content, and a condition but has been proposed by the debtor and the creditor may accept or reject it, we introduce a new possible state: *unset*.

6.1.2 Temporal proposition

From the previous discussion it emerges that a formalism has to be defined to express the content and the condition of social commitments. In particular, given that such propositions present temporal aspects, a sufficiently expressive temporal logic could be used to express them. For example Computation Tree Logic CTL^* [53, 121, 122, 125] could be an adequate formal logic. CTL^* is a logic of discrete time, branching in the future, with future-directed operators; given that the content and the condition of commitments can also refer to the past, such logic has to be augmented with past-directed operators.

Usually it is not difficult to express temporal specification of contents and conditions separately, but we still do not know how to express temporal constrains between them. We plan to investigate these aspects in detail in the future. However the aim of this work is on giving a definition of the meaning of speech acts and to reach this purpose we focus on the analysis of the notion of social commitment. As regard as temporal aspects, to treat the contents and the conditions of commitments we introduce a simple operational model, i.e. a new abstract data type the *temporal proposition class* (see Section 6.2.2), sufficient for the treatment of examples that will be reported in Section 6.5 and of the English Auction Protocol specified in the next chapter.

A temporal proposition includes a statement that may be in one of three different states: *true*, *false* or *undefined*. To be used as the content or condition of a commitment, a statement is typically relative to a time period and it may happen that such a time period is in the future, in this case in the present it is impossible to say if the statement is true or false then we decide to introduce another state: undefined. The time period is described by an initial and a final instant of time. Statements may relate to the associated time period in two different temporal modes: they may be due to be true for the whole time period (like for example in the assertion "the price of this product is going to be 5$ for one week from now"); or they may be due to become true at some moment within the time period (like for example in the assertions "it rained yesterday" and "it will rain tomorrow"). Using these kinds of temporal propositions is possible to cover a wide spectrum of commonly used propositions.

6.2 Technical Specification

Our open interaction system consists of:

- A variable group of registered agents $\{a, b, ...\}$; a procedure is supposed to be provided by which a new agent can register and join the group.

- A variable set of commitment objects $\{C_1, C_2, ...\}$; each commitment object is an instance of the commitment class described below.

- A variable set of temporal proposition objects $\{P, Q, ...\}$; which are instances of the corresponding class discussed below. They are used to express propositions about the application domain and the interaction process needed to represent the content or the condition of commitment objects, and their dynamic evolution is used to check if a commitment is fulfilled or violated.

- A fixed set of *actions* that the agents are able to perform. In particular some of these actions correspond to the performance of one of the speech acts available in the system and can be expressed in term of operations on commitments objects, while other are application domain actions.

- A fixed set of event-driven routines that automatically update the state of commitment objects as a function of their content or condition. These routines are represented by *update rules* as described in Section 6.2.4.

- A set of domain-specific objects $\{O_1, O_2, ...\}$, which represent entities of the application world. Such entities may possess both "natural" or and "institutional" attributes; for example, the color of a product being sold is a natural attribute, while the price of the same product is an institutional attribute. Natural attributes are assumed to reflect the physical properties of the corresponding entities of the real world, and typically cannot be changed during an interaction (of course, they might be changed if some of the interacting agents were assumed to be physical robots). On the contrary, institutional attributes can be affected by the performance of certain communicative acts, in particular by declarations (as discussed in Section 6.3). We assume that each domain-specific object has a value-setting method for each of its institutional properties; for example the method "setState()" can be invoked to set the "state" property.

- A fixed set of *roles* $\{role_1, role_2, ...\}$. This concept is introduced to abstract from the specific agent that takes part in an interaction and to express authorizations as discussed blow. A formal and accurate analysis of the notion of role is beyond the scopes of this thesis, an intuitive description of the notion of role can be found in Section 3.1.3.

- A fixed set of *authorizations* associated to roles, that specify which agent, on the basis of its roles, is authorized to perform a particular declaration (see Section 6.3 for details).

The global state s_t of this open system at time t, is composed by the registered agents, by the proposition and commitment objects and by domain-specific objects. When an agent performs an action, the system evolves to another global state s_{t+1}, where the internal states of commitment, proposition objects, and domain-specific objects are changed according to the effects of the action performed and by the update rules.

6.2.1 The Commitment Class

In this section we give a formal description of the commitment abstract data type. Following the object oriented paradigm, we formalize it as a class, defining the private fields that characterize a commitment object and the public methods used to manipulate it. As mentioned before the notion of conditional commitment is taken as primitive.

Fields

A commitment has the following fields:

- *Identifier* is the univocal identifier of the object in the system.
- *Debtor* is the name of the registered agent that has the commitment.
- *Creditor* is the name of the registered agent relative to which the commitment is made.
- *State* is used to keep trace of the dynamic evolution of the commitment object. This field can assume one of following values:
  - *unset* (u): the commitment has been proposed (by its debtor, its creditor, or a third party) but it has not yet been accepted nor refused;
  - *cancelled* (c): the commitment has been withdrawn by its creditor or refused by its debtor, or the condition has not become true;
  - *pending* (p): the commitment has been accepted but its condition has not yet become true;
  - *active* (a): the commitment's condition has become true;
  - *fulfilled* (f): the content of an active commitment has become true;
  - *violated* (v): the content of an active commitment has become false;

  The life-cycle of a commitment is shown in Figure 6.1. The state of a commitment can change as an effect of the invocation of its basic methods (solid lines) or of environmental events (dotted lines), that is, of events that change the truth-value of a commitment's conditions or content.
- *Content* is a temporal proposition object (see below) that represents the state of affairs or a course of action to which the debtor is committed.
- The commitment's *conditions*, that is, a list of propositions that have to be satisfied, within the interval of time indicated, in order for the commitment to become active;

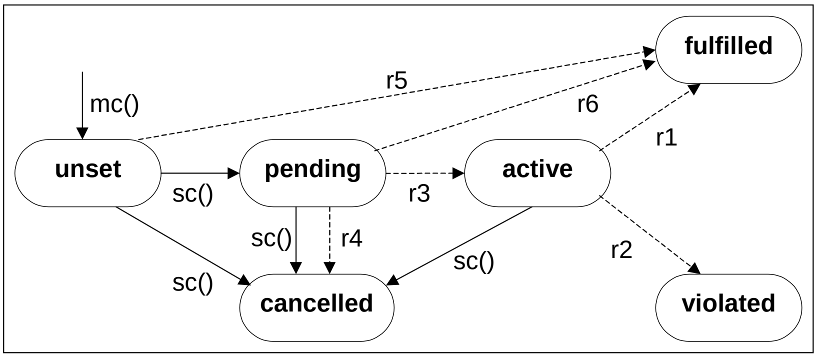

Figure 6.1: The life-cycle of commitments.

- *Time-out* which is relevant only in the case of unset commitments, and will therefore be treated as an optional parameter. It expresses the duration of the life of an unset commitment.

Between the interval of time of the content field and of the condition field there is a temporal constraint. When it is crucial that the condition has to be performed in order that the performance of the content makes sense, the time interval of the condition must precede the time interval of the content. Following Allen's approach [3] it is possible to say that between the two intervals there must be a relation of $before$. An example of this type of conditional commitment could be "if I win the lottery at time t_1, I will give you everything I have at a time subsequent t_1". There are commitments where it is not crucial that the condition has to become true before the content becomes true. In other words if the content is true but the condition is still undefined the fulfilment of the commitment makes sense. In these cases the time interval of the condition is in a relation of "*start*" with the time interval of the content as shown if Figure 6.2.

Commitment objects are represented with the following notation:

$$C_{id}(state,\ debtor,\ creditor,\ content|\ condition\ \{, time-out\}).$$

For example:

- $C_i(active, a, b, P|TRUE)$ is an active commitment by a, relative to b, that P will be satisfied, $TRUE$ is the identically true temporal proposition object.

- $C_i(pending, a, b, P|Q)$ is a pending commitment by a, relative to b, that if Q is satisfied then also P will be satisfied.

As we have already said, temporal proposition objects are used to represent content and conditions of a commitment. In particular the conditions of a commitment consist of a list $[P, Q, ...]$ of temporal proposition objects that have to be satisfied in order for the commitment to become active. The truth value of a list of temporal proposition objects is computed as follows:

- an empty list of temporal proposition objects is true;

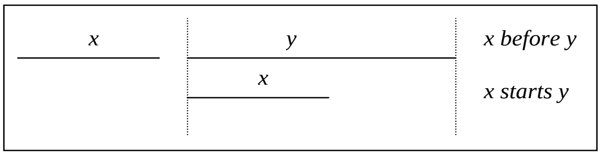

Figure 6.2: Interval Relations.

- a true temporal proposition object is removed from the list;
- a list containing a false proposition object is false.

To make the notation simpler, when the list of conditions contains one temporal proposition object the square brackets are dropped. We also remark that a temporal proposition object, used to express the content or a condition of a commitment object, may in turn represent another commitment object. In particular temporal proposition objects can be used to represent conditions on the temporal evolution of commitments. An example of this is given in Subsection 6.3.5.

Basic Methods

We assume that when a commitment object is declared, the constructor of the class allocates a memory block and creates an empty commitment object $C_i()$. In our notation the invocation of a method of the commitment object is represented by the name of the object followed by a dot and by the name of the method with its parameter list.

Commitments are created and manipulated through the following basic operations:

- *Make commitment.* Both the debtor and the creditor are authorized to invoke the operation $mc(a, b, P, Q)$ with arbitrary creditor a, debtor b, content P, and condition Q. The effects of the invocation of this method are the creation of a new unset commitment object.

$$C_i().mc(a, b, P, Q) \rightarrow C_i(unset, a, b, P|Q).$$

- *Set commitment.* This method $sc(s)$ changes the state of an existing commitment object to the one specified s. There are two different preconditions on who can invoke this method on the base of the current and future state of the commitment object (as is shown also in Figure 6.1).
  - If the state is unset the debtor is authorized to set it to either cancelled or pending;
  - If the state is pending or active the creditor is authorized to set it to cancelled;

$$C_i(-, a, b, P|Q).sc(s) \rightarrow C_i(s, a, b, P|Q).$$

- *Add condition.* The method $ac(R)$ adds a new temporal proposition object R to the conditions of the commitment:

$$C_i(s, a, b, P|Q).ac(R) \rightarrow C_i(s, a, b, P|R \bullet Q),$$

where the symbol $\bullet$ denotes the operation of inserting a new element in a list.

Basic operations should not be viewed as actions that are directly performed by agents. Rather, they are low-level primitives used to implement operations on commitment objects, more specifically, agents manipulate commitments through a communicative act library (see Section 6.3). It is possible to extend such basic operations with the ones necessary to further manipulate the content and the condition of an existing commitment object an important option in a negotiation process.

Finally note that we defined the conditions under which commitments are fulfilled or violated, but we are not concerned with the management of violations e.g. in terms of sanctions, because this aspect lies beyond the use of commitments for the definition of ACL semantics.

6.2.2 The Temporal Proposition Class

To express the content and the condition of a commitment object it is necessary to define or use a proper *content language* as discussed in Section 5.2.2. We decide to introduce the temporal proposition object to express the possible values of the content and condition fields. Temporal proposition object are referred to by a capital letter (P, Q,...). The definition of a temporal proposition class has the following fields. The methods to get and to set the current value of each field are supposed to exist.

- *Statement* is a sentence in a suitable formal language, for example predicative logic. The atomic formulas of such predicative logic allow stating that:
  - A state of affairs (in the application domain) holds.
  - An action (see Section 6.2.3) has been performed. Example are "agent a do α" or "Inform(a,b,P)", a communicative act presents in the library.
  - A commitment with certain specific value of its fields holds. In this case the temporal proposition object becomes true when a commitment object, corresponding to this sentence, is actually created. This third case allows us to express conditional commitments, which become active only if another specific commitment object is created.
- *Time Interval* expresses the time period to which the sentence in the statement field is referred.
- *Mode* expresses the temporal qualification of the statement, which can be required to be true for a whole time interval ($\forall$) or to become true during the time interval ($\exists$).

- *State* can be true (1), false (0) or undefined ($\perp$). This is not a three value logic but a logic of *future contingents*. The problem of expressing absence of knowledge about the future was a well-known classical problem formalized by Prior in 1957 [121]. It is up to a "notifier", connected with the meaning of the sentence to change the state from undefined to true or false, according to the following rules:
  - If the mode is '$\forall$' the notifier sets the state to false if the sentence becomes false at any point of the time interval; otherwise the notifier sets the state to true when the time interval expires.
  - If the mode is '$\exists$' the notifier sets the state to true if the sentence becomes true at any point of the time interval; otherwise the notifier sets the state to false when the time interval expires.

  In particular cases it is possible to infer in advance that the statement of a temporal proposition object can no longer become true (false) within the associated time interval. In this case the notifier may set the truth value to false (true) before the time interval expires. To do so, the notifier may exploit specific inference rules. For example if the statement of the temporal proposition object requires that a specific commitment object becomes active to become true, and the state of the commitment instead becomes cancelled it is possible to infer from the allowed dynamic evolution of commitments that the commitment cannot any more becomes active and then the temporal proposition object can be set to false. Given that the finite state machine described in Figure 6.1 is deterministic it is possible to exhaustively express all inference rules that the system will need to infer in advance the truth value of a temporal proposition object whose statement refers to a certain commitment object.

  It is important to underline that when the state changes from undefined to true or false it cannot change anymore. We denote with the symbol $TRUE$ a constant temporal proposition object with an empty statement and a state that is true.

Finally, any Boolean combination of temporal proposition objects is still a temporal proposition object. In this case the truth-value of the resulting temporal proposition object can be obtained from the truth-value of the component temporal proposition objects using the extended truth table reported in Appendix B.

6.2.3 Actions

An agent can perform actions in the environment. The set of executable actions includes:

- all acts in *communicative acts library*, where each communicative act is defined using basic operation on commitment objects or other communicative acts already defined.
- all actions in a set of *application domain actions*. For example, in an interaction system used to negotiate some commodities, "pay x\$ to a" and "deliver object o to a" can be application domain actions.

6.2.4 Update Rules

When a temporal proposition object, associated with the content or condition field of a commitment object, changes its state from undefined to true or false, it is necessary to update the state of the commitment object. We shall formalize the event driven routines that are automatically invoked using the update rules described in Table 6.1.

In the first column there is the event that triggers the rule, while the associated state transition is described in the second column. The method *state()* is a method of the temporal proposition class that returns the state of a temporal proposition object.

- *Rule 1.* When a commitment object is active and its content becomes true, the commitment becomes fulfilled.
- *Rule 2.* When a commitment object is active and its content becomes false, the commitment becomes violated.
- *Rule 3.* When the condition of a pending commitment becomes true the commitment object becomes active.
- *Rule 4.* When the condition of a pending commitment becomes false the commitment object becomes cancelled.
- *Rule 5.* This rule is necessary to cover situations when an agent performs a requested action without accepting the request explicitly. It is important that the action described by the proposition P is an action performed by the debtor of the commitment. In this case commitment is unset, but the execution of the action described by P by the debtor b of the commitment changes the commitment's state in active, then Rule 1 changes it to fulfilled. A possible solution to make explicit the connection between the performance of an action and the request to perform that action is through an explicit reference made by the actor to the associated request.
- *Rule 6.* When the content of a commitment object becomes true, even if the condition is still undefined, the commitment becomes fulfilled.
- *Rule 7.* When the time-out value of an unset commitment expires the commitment becomes cancelled.

Table 6.1: Update Rules

event	action	rule
$P.state() = 1$	$C_i(active, a, b, P\|TRUE) \rightarrow C_i(fulfilled, a, b, P\|TRUE)$	1
$P.state() = 0$	$C_i(active, a, b, P\|TRUE) \rightarrow C_i(violated, a, b, P\|TRUE)$	2
$Q.state() = 1$	$C_i(pending, a, b, P\|Q) \rightarrow C_i(active, a, b, P\|TRUE)$	3
$Q.state() = 0$	$C_i(pending, a, b, P\|Q) \rightarrow C_i(cancelled, a, b, P\|Q)$	4
$P.state() = 1$	$C_i(unset, a, b, P\|TRUE) \rightarrow C_i(active, a, b, P\|TRUE)$	5
$P.state() = 1$	$C_i(pending, a, b, P\|Q) \rightarrow C_i(fulfilled, a, b, P\|Q)$	6
$curr\_time > t$	$C_i(unset, a, b, P\|Q, t) \rightarrow C_i(cancelled, a, b, P\|Q)$	7

6.3 Definition of Main Speech Acts

In this section the previously defined basic operations on commitment objects are used to express the meaning of some important speech acts, especially the ones common used in multi agent interaction in open environments. The set of defined communicative acts compose a Library of Communicative Acts. In our analysis, we follow the taxonomy by Searle [140] that classifies illocutionary acts into five categories: *assertives*, *commissives*, *directives*, *declaratives*, and *expressives*. Here we do not treat expressives, because we think they are not relevant for artificial agent communication. While we do treat *proposals* and *offers* for they importance in electronic commerce applications. They do not belong to such five categories and can be defined using commissives and directives acts.

In the following definitions every speech act has the following arguments: a sender, a receiver, a content and in some cases a condition. The "$=_{def}$" sign means that performing the action represented on the left-hand side is the same as performing the action or executing the operations represented on the right-hand side. The ":=" symbol means that the act represented on the left-hand side is actually performed through the invocation of the methods listed on the right-hand side. An expression of the form $\{op_1; ...; op_n\}$ means that the basic operations $op_1, ..., op_n$ are executed in sequence; an expression of the form $\{op_1 \wedge ... \wedge op_n\}$ means that the basic operations $op_1, ..., op_n$ are executed in parallel. While if the symbol "$\vee$" is used as separator, it means that the act on the left-hand side of the definition is actually performed when one of the acts on the right-hand side is performed.

6.3.1 Assertives

According to Speech Act Theory the point of an assertive act is to commit the speaker relative to the hearer to the truth of what is asserted. In human language, the simplest type of assertive act is *asserting*. However, we conform here to the practice, common in the ACL field, of considering *informing* as the prototypical assertive act.

- *Inform.* This act is used when an agent a wants to inform agent b that P is the case, for example P can be "from now agent a is on-line", or "my name is ...". In a commitment-based approach, an act of informing can be defined as follows:

  $$\begin{aligned}&\mathbf{inform}(a, b, P) :=\\ &\{C_i().mc(a, b, P, TRUE);\ C_i(unset, a, b, P|TRUE).sc(pending)\}.\end{aligned}$$

  The final results thanks to the intervention of update Rule 3 is the creation of an active commitment.

- *InformIf.* This is an abstract act, that is an agent cannot actually do an informIf act but performing one of two specific inform acts is equivalent as performing an informIf act. Therefore, when the statement of a temporal proposition object is "informIf" it is enough that will be done one of the two specific inform act to make the object true. This abstract act will be used in the following sections to define questions.

An agent does an informIf act when it informs another agent about the truth value, either true or false, of a proposition. Let P be a temporal proposition object with sentence φ in its statement field. We shall call $P^{\neg}$ the temporal proposition object that has in the statement field the negation of the sentence φ. The informIf act can be defined as follows:

$$\begin{aligned}&\mathbf{informIf}(a, b, P) =_{def}\\&\{inform(a, b, P) \ \lor \ inform(a, b, P^{\neg})\}.\end{aligned}$$

- *InformRef.* This is another abstract act that will be used in the following sections to define questions. We shall call $S(\mathbf{x})$ a temporal proposition object whose statement is a sentence containing a meta-variable $\mathbf{x}$ standing for an arbitrary constant. An agent actually does an informRef act about a given temporal proposition object $S(\mathbf{x})$ when it performs an inform act about a temporal proposition object $S(c)$, where $S(c)$ is obtained by $S(\mathbf{x})$ substituting variable $\mathbf{x}$ with the constant c.

$$\mathbf{informRef}(a, b, \mathbf{x}) =_{def} \{inform(a, b, S(c))\}$$

6.3.2 Directives

The point of a directive act is to get the hearer to perform an action within an interval of time, even a particular speech act. As we already pointed out, an agent cannot directly commit another agent to something but it can request to another agent if it accepts or rejects to be committed to a certain commitment. Therefore, to define directives we use the concept of pre-commitment (see Section 6.1) that is formalized in our model as an unset commitment.

- *Request.* We choose as basic directive act the request. There are requests that attempt to get the hearer to accept a commitment without any condition, for example "would you like to borrow to me this book?". While there are requests that attempt to get the hearer to accept a commitment with a condition, an example can be, "if it rains, can you give me a lift?". It is important to note that the realization of the condition is not in the power of the speaker.

  The request from agent a to agent b to bring about P within a certain interval of time if condition Q is satisfied is defined as:

$$\mathbf{request}(a, b, P, Q) := \{C_i().mc(b, a, P, Q)\}.$$

  When the condition does not exist is enough to perform the request act with the constant symbol true $TRUE$ as condition.

  Some requests can be satisfied immediately, that is the answer may be the execution of the requested action or a rejection. This can happen for example when the request is: "please close the door". If the debtor of the unset commitment immediately executes the action requested, the update Rule 5 is activated and the state of the commitment object changes from unset to active. On the contrary, if the request is rejected the unset commitment is cancelled. In any case, thanks to Rule 7 an unset commitment is automatically cancelled after a predefined time-out has expired. There are also requests to do actions that cannot be immediately executed, this can happen for example with the request "please send me a copy of your book". Here the hearer (i.e., the debtor of the unset commitment) may react by *accepting* or *rejecting* the request. If the debtor of the unset commitment accepts it, then it is committed to do the action, otherwise it is not. To summarize the different possibilities: the hearer of a request can react in three different ways: it can perform the action, or accept the request, or reject the request. If the time-out has elapsed, the unset commitment is cancelled.

- *Question* A question act is a particular directive act, by which the speaker tries to get the hearer to commit to the truth of a certain proposition. A question can be seen as a *request to inform* about something; therefore we define questions using the assertive acts previously defined. Three type of questions will be analyzed.

  - *Question.* The simplest form of a question is: "tell me that ...". It is rarely used in human communication and usually the answer is directly the assertion requested or a reject. It can be formalized using the inform act. The method *statement()* of the temporal proposition class returns the content of the statement field.

  $$\begin{aligned}&\textbf{question}(a,b,P) := \{request(a,b,P)\}\\&\quad where\ P.statement() = inform(b,a,S)\end{aligned}$$

    Usually a positive answer to a question is the performance of the inform act instead of an accept act.

  - *query-if.* Query-if acts are used to get the hearer to commit to the truth value of a given proposition, for example "is it raining?". A query-if act can be defined using the informIf act:

  $$\begin{aligned}&\textbf{query-if}(a,b,S)) := \{request(a,b,P)\}\\&\quad where\ \ P.statement() = informIf(b,a,S)\end{aligned}$$

  - *query-ref.* Query-ref acts are used to get the hearer to identify an individual having a given property. The speaker provides a "frame" for the answer, and the hearer has to fill it with the correct piece of information. For example, a query-ref act may be: "What is your name?", that has the same meaning as "Please tell me the actual value of the meta-variable **x**", to which the hearer has to assign a constant value. A query-ref act can be defined using the InformRef act.

  $$\begin{aligned}&\textbf{query-ref}(a,b,S(\mathbf{x}))) := \{request(a,b,P)\}\\&\quad where\ P.statement() = informRef(b,a,S(\mathbf{x}))\end{aligned}$$

Indeed, as remarked by Searle [142] the concept of a question is more general: by a question, an agent may request the execution of a non-assertive communicative act (like a directive, or a commissive). However, our definition above easily generalizes to such cases, an example can be found in the next chapter.

6.3.3 Commissives

The point of a commissive act is to commit the debtor, relative to the creditor, to the execution of an action of a given type within a given interval of time. Here will be defined the basic commissive act, promising.

- *Promise.* In some cases an agent a can promises to agent b that a will bring about P within a time interval in the future, for example the promise can be "I will visit you tomorrow". In other situations agent a can perform a conditional promise to agent b, in this case a commits to bring about P (within a given time interval) if condition Q becomes true (within a given time interval that precedes the time interval of P). Agent b may or may not have the responsibility for condition Q to become true. An example where the condition does not depend on the hearer is, "If it rains tomorrow, I shall give you 5$ the day after tomorrow". An example where the condition depends on the hearer is, "If you give me this book within time t_1, I will give you 5$ within time t_2", with $t_1 < t_2$.

$$\begin{aligned}&\mathbf{promise}(a, b, P, Q) := \\ &\{C_i().mc(a, b, P, Q);\ C_i(unset, a, b, P|Q).sc(pending)\}.\end{aligned}$$

  When the condition does not exist is enough to perform the promise act with the constant symbol true $TRUE$ as condition, in this case the final result for the intervention of Rule 3 is the creation of an active commitment.

  It appears from this definition that there is no difference between agent a inform agent b that a will do an action in the future, and agent a promising to agent b that a will do an action. In Searle's speech act theory the difference between assertives and commissives is dealt with in terms of a property called *direction of fit* (see Section 5.1). Direction of fit may be relevant in a commitment-based approach, especially because it is likely to influence the management of commitment violations. However, at the present stage of development we do not include this aspect in our operational specification.

- *Accept, Reject, and Conditional Accept.* Three speech acts can be performed only in response to an unset commitment, they are the accept, the reject, and the conditional accept act. The act of accepting an unset commitment transforms it in an active commitment or in a pending commitment (if there is a condition), while the act of rejecting an unset commitment has the effect of cancelling it.

$$preconditions:\ \exists\ C_i(unset, b, a, P|Q))$$

$$\begin{aligned}&\textbf{accept}(b,a,C_i(\mathit{unset},b,a,P|Q)) :=\\&\{C_i(\mathit{unset},b,a,P|Q).sc(\mathit{pending})\}\end{aligned}$$

$$\begin{aligned}&\mathit{preconditions}:\ \exists\, C_i(\mathit{unset},b,a,P|Q))\\&\textbf{reject}(b,a,C_i(\mathit{unset},b,a,P|Q)) :=\\&\{C_i(\mathit{unset},b,a,P|Q).sc(\mathit{cancelled})\}\end{aligned}$$

Another useful commissive act is "conditional accept", which may be used by agents to negotiate the condition of an unset commitment. In particular, conditional acceptance will appear in the example proposed in the next chapter. In fact, in the English Auction Protocol at every round of the bidding process the auctioneer accepts the currently highest bid on condition that no higher bids will be accepted later. In general, the debtor of an unset conditional commitment C_i can accept it provided that an additional condition, represented by a temporal proposition object, holds. Conditional acceptance transforms an unset commitment into a pending commitment, and adds a new condition to the original condition list of the unset commitment:

$$\begin{aligned}&\mathit{preconditions}:\ \exists\, C_i(\mathit{unset},b,a,P|Q))\\&\textbf{condAccept}(b,a,C_i(\mathit{unset},b,a,P|Q),R) :=\\&\{C_i(\mathit{unset},b,a,P|Q).ac(R);\\&C_i(\mathit{unset},b,a,P|R\bullet Q).sc(\mathit{pending})\}\end{aligned}$$

Note that when condition R becomes true, the debtor is left with a pending conditional commitment of the form $C_i(\mathit{pending},b,a,P|Q)$.

6.3.4 Declarations

Declarations are a special type of communicative acts. Examples of declarations are "I pronounce you man and wife" or "I declare the auction open". The point of a declaration is to bring about a change in the world, obviously not in the physical or natural world but in an institutional world [35], that is, a conventional world relying on common agreement of the interacting agents (or, more precisely, of their designers). Declarations actually change the institutional world simply in virtue of their successful performance. In our interaction framework, to treat declarations we introduce objects with *institutional properties*, that is, conventional properties that result from common agreement, like for example the ownership of a product. Such properties can be affected by declaration acts. It is however necessary to identify which agents are *authorized* or *empowered* to perform a given declaration act in the system. Typically, authorizations are granted to agents in virtue of the *role* they play in an interaction, and thus authorizations are naturally associated to roles. To do so, we need to introduce a construct to express that an agent having a given role in the interaction system is empowered to bring about an institutional change of a given kind:

$$preconditions : empowered(role_i, O_k.setProp_j()) \wedge a.role() = role_i$$
$$\mathbf{declare}(a, O_k.prop_j = x) := \{O_k.setProp_j(x)\}.$$

6.3.5 Proposals

Proposals do not belong to the five basic categories of illocutionary acts introduced by Speech Act Theory. Therefore a proposal act can be defined using the acts previously defined. We decide to introduce proposals in our Communicative Acts Library for their importance in electronic commerce applications. For example in certain type of electronic auctions when a buyer submit its bid to the auctioneer it is performing a proposal as defined below.

A proposal is a conjunction of a conditional directive and a conditional commissive. For example, we can analyze the following proposal made by agent a to agent b, "Will you give me object o within time t_2 if I give you x€ within time t_1?", with $t_1 < t_2$. Note that a proposal is similar to a request with a non-empty condition. The crucial difference is in the condition, because in proposals the realization of the condition depends on the speaker.

$$\begin{aligned}&\mathbf{propose}(a, b, P, Q) :=\\&\{request(a, b, P, Q) \;\wedge\; promise(a, b, Q, S)\}\\&\quad where \;\; S.statement() = C_i(pending, b, a, P|Q)\}\end{aligned}$$

It is important to put in evidence that the promise is related to the request through the common proposition object Q.

6.3.6 Offers

Offers do not belong to the five basic categories of illocutionary acts introduced by Speech Act Theory. Therefore an *offer* act can be defined using the acts previously defined. This speech act is very important is electronic commerce applications. For example in the Dutch (descending) auction the seller continuously lowers the price until one of the bidders takes the item at the current price. In practice the seller continuously perform an offer to the bidders as defined below.

A proposal is a conjunction of a directive and a conditional commissive. An example of an offer in natural language can be: "Would you like to give me x€ within time t_1 if I will give you the object o within time t_2", with $t_1 < t_2$.

$$\begin{aligned}&\mathbf{offer}(a, b, P, Q) :=\\&\{request(a, b, Q) \;\wedge\; promise(a, b, P, S)\}\\&\quad where\; S.statement() = Q \wedge R\\&\quad where\; R.statement() = C_i(active, b, a, Q|TRUE)\end{aligned}$$

It is important to put in evidence that the promise is related to the request through the common proposition object Q. The statement field of the condition S of the promise is the Boolean *and* of the other two temporal propositions, in order that if the offer is rejected the system will reach a final state as it is shown in Table 6.4.

6.4 Important Use of the Proposed Semantics

The definition of a standard and objective semantics for Agent Communication Languages is a crucial task mainly for its various and important use in the field of artificial communication. From my point of view the proposed commitment-based semantics can be used mainly to realize the following functions:

1. To define in unambiguous objective way the messages available in various interaction systems or those that form interaction protocols (as will be described in next chapter), in order to be used by human and artificial agents. For example it could be possible to express the meaning of the messages available in the auctions of the Trading Agent Competition game.

2. The proposed semantics for Agent Communication Languages does not make any assumption about the internal structure of the artificial agents that may use it; they can be either reactive or deliberative. Deliberative agents may use the defined semantics to reason about their action and those of other participants in order to plan their communicative acts. While in other situations, for example when the time available between two interactions is very short, reactive agent would still be better than deliberative agents. In such situations interaction protocols, which represent allowed interactions among communicative agents, would have to be used.

3. To keep trace of the dynamic evolution of the system on the base of the actions performed by the agents. In this way it possible for every agent or for some institutional entity to *verify* if an agent is behaving in accordance with its commitments.

4. Stabilize the interactions among agents. This is possible because commitments create the expectation that the other agents will behave in certain ways, and a means to deal with situations in which such expectations are not met.

5. To design interaction protocols that do not over constrain the actions of participant agents. An arbitrary sequence of messages in general does not describe a "sensible" interaction. To do so the interaction must satisfy certain *soundness conditions* that guarantee some important properties to the interaction or to the protocol as will be described in detail in next chapter.

6. To test if an agent is behaving in accordance with the commonly adopted interaction protocol. This is called *compliance* to the interaction protocol and will be discussed in next chapter.

7. Given that the proposed semantics is a formalization of studies about human communication, the adoption of this semantics is crucial to obtain successful mixed interactions among human beings and software agents.

8. The notion of commitment defined so far together with the definition of the meaning of the basic speech acts form what is called in [33] a "Core Institution" that can be used to define further special institutions to treat conversations, authority, obligations, and permissions.

6.5 Samples of Application

In this section three simple examples of interactions will be formalized using the library of communicative acts previously defined. In particular to deal with significant examples we will analyze interaction protocol widely used in electronic commerce application like the protocol of query, the protocol of proposal and the protocol of offer. The reason why we present such examples is to show :

- how it is possible to use commitment-based definitions of speech acts to express the meaning of the various messages exchanged in a negotiation;

- the dynamic evolution of the system states during an interaction; state transitions are computed following the semantics previously defined;

Whereas in the next chapter a general, verifiable and application-independent method for the definition of interaction protocols to be used in open multiagent systems will be presented.

A protocol is based on the set of communicative acts previously defined as operations on commitment objects. It is described by an *interaction diagram*, that is, a graph whose nodes represent system states, and whose edges represent certain types of state transitions. In an interaction diagram, state transitions correspond either to communicative acts performed by the interacting agents, or to environmental events strictly related to the interaction.

To each state, a *content* is associated, consisting of commitment objects, temporal proposition objects, and domain-specific objects holding at that particular state. Such objects are computed in the following way.

- The diagram has a distinguished *initial state* (*start*), whose content is a set of empty commitments.

- If state s_j is reached from state s_i by a speech act, then the content of state s_i must satisfy the preconditions of the speech act, and the content of s_j is obtained by modifying the content of s_i according to the definition of the speech act.

- If on the contrary state s_j is reached from state s_i by an environmental event, then the content of s_j is obtained by modifying the content of s_i according to the relevant update rule.

- States with no outgoing edges are classified as *final*; when a final state is reached, the interaction ends. Final states are marked by double blocks.

Many types of interaction include a first phase of *negotiation* and a second phase of *execution*. The negotiation phase has the start state, and ends in one or more *contract states*, whose content includes only commitments that are either active or conditional. From a contract state it is possible to enter the execution phase that ends into final states. It is during the execution phase that the artificial interaction system needs to be connected with the real world. For example if a contract is reached among a seller and a buyer on the exchange of a book during the execution phase the money and the book have to be actually exchanged and this facts have to be reported in the artificial system.

6.5.1 Query

This example shows the dynamic evolution of the system states when a query-ref act is performed by agent a addressing agent b. Figure 6.3 reports the interaction diagram, while the content of each state is described in Table 6.2.

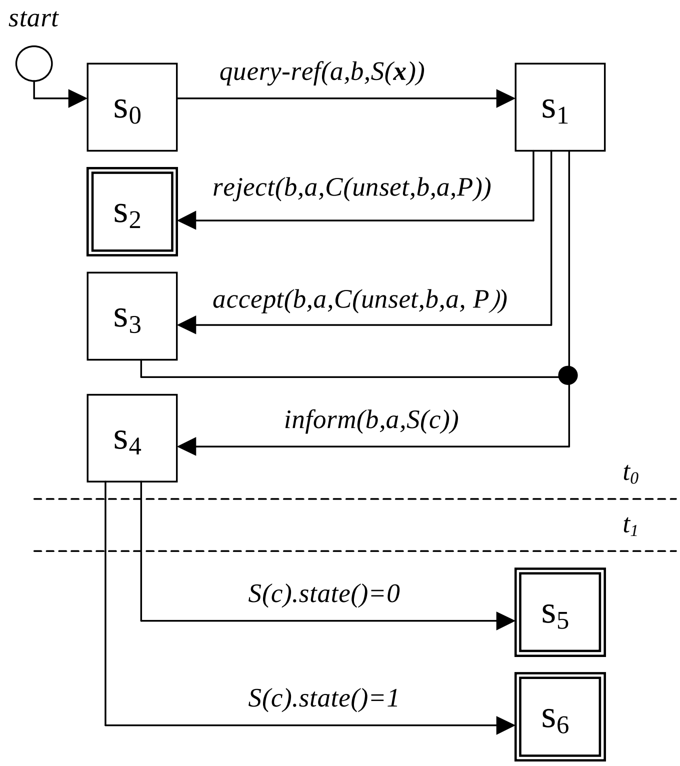

Figure 6.3: Interaction diagram of the query protocol (see Table 6.2 for state contents)

Table 6.2: Contents associated to the states of the interaction diagram of the query protocol

$S(\mathbf{x})(''actual\ value\ of\ \mathbf{x}\ is...'', now \ldots t_1, \forall, \perp)$ $P(informRef(b, a, S(c)), now \ldots t_0, \exists, \perp)$ $t_0 < t_1$

s	reason of the action	content
s_0	$start$	$C_i()$ $C_j()$
s_1	$query - ref(a, b, S(\mathbf{x}))$	$C_i(unset, b, a, P\|TRUE)$ $C_j()$
s_2	$reject(b, a, C_i(unset, b, a, P\|TRUE))$	$C_i(cancelled, b, a, P\|TRUE)$ $C_j()$
s_3	$accept(b, a, C_i(unset, b, a, P\|TRUE), rule\ 3$	$C_i(active, b, a, P\|TRU)E$ $C_j()$
s_4	$inform(b, a, S(c))$ $P.state() = 1$	$P.state() = 1, C_j(active, b, a, S(c)\|TRUE)$ $C_i(fulfilled, b, a, P\|TRUE)$
s_5	$S(c).state() = 0, rule\ 2$	$C_j(violated, b, a, S(c)\|TRUE)$ $C_i(fulfilled, b, a, P\|TRUE)$
s_6	$S(c).state() = 1, rule\ 1$	$C_j(fulfilled, b, a, S(c)\|TRUE)$ $C_i(fulfilled, b, a, P\|TRUE)$

6.5.2 Proposal

This example shows the dynamic evolution of the system states when a propose act is performed by agent a addressing agent b. Figure 6.4 reports the interaction diagram, while the content of each state is described in Table 6.3. States $s_0, ..., s_3$ represent the negotiation phase of the interaction protocol; in particular, the content of state s_3 represents the contract between the two agents. States $s_4, ..., s_9$ represent the execution of the contract. State s_7 can be reached following two different paths, both paths lead to the same state with the same content but the reasons why such a state is reached are different in one path respect to the other. To express this fact in Table 6.3 the reason of the action for state s_7 are reported in two different rows.

6.5.3 Offer

The example shows the dynamic evolution of the system states when an offer act is performed by agent a addressing agent b. Figure 6.5 reports the interaction diagram, while the content of each state is described in Table 6.4. States $s_0, ..., s_3$ represent the negotiation phase of the interaction protocol; in particular, the content of state s_3 represents the contract between the two agents. States $s_4, ..., s_7$ represent the execution of the contract.

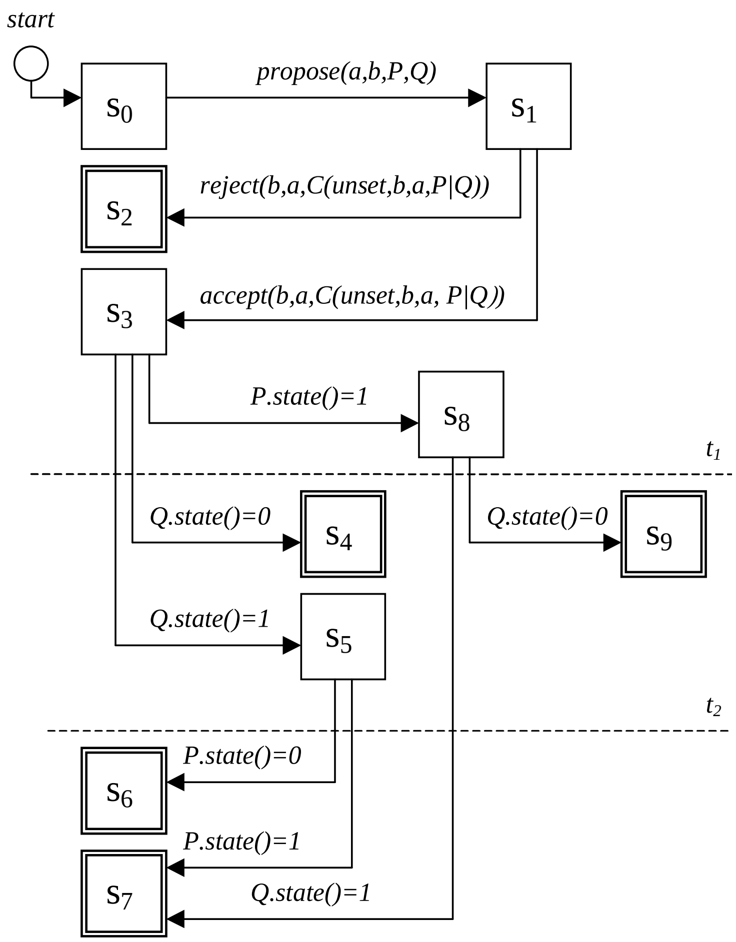

Figure 6.4: Interaction diagram of the proposal protocol (see Table 6.3 for state contents)

Table 6.3: Contents associated to the states of the interaction diagram of the proposal protocol

$$P(b\ do\ \beta, now \ldots t_2, \exists, \perp)$$
$$Q(a\ do\ \alpha, now \ldots t_1, \exists, \perp)$$
$$S(C_i(pending, b, a, P|Q), now \ldots t_1, \exists, \perp)$$
$$t_1 < t_2$$

s	reason of the action	content
s_0	$start$	$C_i()$ $C_j()$
s_1	$propose(a, b, P, Q)$	$C_i(unset, b, a, P\|Q)$ $C_j(pending, a, b, Q\|S)$
s_2	$reject(b, a, C_i(unset, b, a, P\|Q))$ $notifier$ rule 4	$C_i(cancelled, b, a, P\|Q)$ $S.state() = 0$ $C_j(cancelled, a, b, Q\|S)$
s_3	$accept(b, a, C_i(unset, b, a, P\|Q)$ $notifier$ $rule$ 3	$C_i(pending, b, a, P\|Q)$ $S.state() = 1$ $C_j(active, a, b, Q\|S)$
s_4	$Q.state = 0$, $rule$ 4 $rule$ 2	$S.state() = 1$ $Q.state = 0$, $C_i(cancelled, b, a, P\|Q)$ $C_j(violated, a, b, Q\|S)$
s_5	$Q.state = 1$, $rule$ 3 $rule$ 1	$S.state() = 1$ $Q.state = 1$, $C_i(active, b, a, P\|Q)$ $C_j(fulfilled, a, b, Q\|S)$
s_6	$P.state = 0$, $rule$ 2	$S.state() = 1$ $Q.state = 1$ $C_i(violated, b, a, P\|Q)$ $C_j(fulfilled, a, b, Q\|S)$
s_7 $from$ s_5	$P.state = 1$, $rule$ 1	$S.state() = 1$ $Q.state = 1$ $P.state = 1$, $C_i(fulfilled, b, a, P\|Q)$ $C_j(fulfilled, a, b, Q\|S)$
s_8	$P.state = 1$, $rule$ 6	$S.state() = 1$ $P.state = 1$, $C_i(fulfilled, b, a, P\|Q)$ $C_j(active, a, b, Q\|S)$
s_7 $from$ s_8	$Q.state = 1, rule\ 1$	$S.state() = 1$ $P.state = 1$ $C_i(fulfilled, b, a, P\|Q)$ $Q.state = 1, C_j(fulfilled, a, b, Q\|S)$
s_9	$Q.state = 0$, $rule$ 2	$S.state() = 1$ $P.state() = 1$ $C_i(fulfilled, b, a, P\|Q)$ $C_j(violated, a, b, Q\|S)$

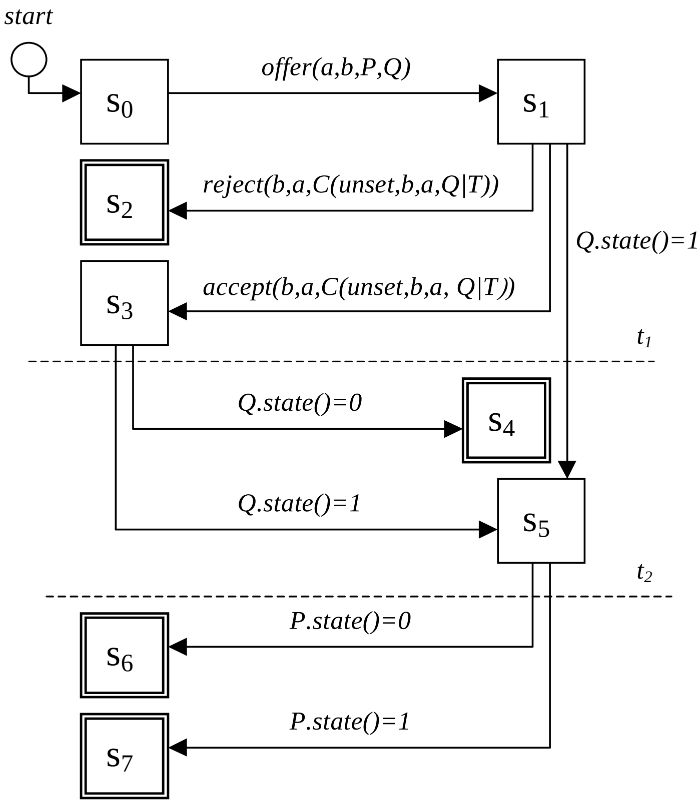

Figure 6.5: Interaction diagram of the offer protocol (see Table 6.4 for state contents)

Table 6.4: Contents associated to the states of the interaction diagram of the offer protocol

$$P(a\ do\ \beta, now \ldots t_2, \exists, \perp)$$
$$Q(b\ do\ \alpha, now \ldots t_1, \exists, \perp)$$
$$S(Q \wedge R, now \ldots t_1, \exists, \perp)$$
$$R(C_i(active, b, a, Q|TRUE)), now \ldots t_1, \exists, \perp)$$
$$t_1 < t_2$$

s	reason of the action	content
s_0	$start$	$C_i()$ $C_j()$
s_1	$offer(a, b, P, Q)$	$C_i(unset, b, a, Q\|TRUE)$ $C_j(pending, a, b, P\|S)$
s_2	$reject(b, a, C_i(unset, b, a, Q\|TRUE))$ $notifier$ $notifier$ $rule\ 4$	$C_i(cancelled, b, a, Q\|TRUE)$ $R.state() = 0$ $S.state() = 0$ $C_j(cancelled, a, b, P\|S)$
s_3	$accept(b, a, C_i(unset, b, a, Q\|TRUE)), rule\ 3$ $notifier$	$C_i(active, b, a, Q\|TRUE)$ $C_j(pending, a, b, P\|S)$ $R.state() = 1$
s_4	 $Q.state = 0, notifier$ $rule\ 4$ $rule\ 2$	$R.state() = 1$ $Q.state = 0,\ S.state() = 0$ $C_j(cancelled, a, b, P\|S)$ $C_i(violated, b, a, Q\|TRUE)$
s_5 $from$ s_3	 $Q.state = 1$, notifier $rule\ 3$ $rule\ 1$	$R.state() = 1$ $Q.state() = 1, S.state() = 1$ $C_j(active, a, b, P\|S)$ $C_i(fulfilled, b, a, Q\|TRUE)$
s_5 $from$ s_1	$rule\ 5, Q.state() = 1, rule\ 1$ $notifier$ $rule\ 3$	$R.state() = 1, C_i(fulfilled, b, a, Q\|TRUE)$ $Q.state() = 1, S.state() = 1$ $C_j(active, a, b, P\|S)$
s_6	 $P.state = 0,\ rule\ 2$	$R.state() = 1$ $Q.state = 1$ $S.state = 1$ $P.state = 0, C_j(violated, a, b, P\|S)$ $C_i(fulfilled, b, a, Q\|TRUE)$
s_7	 $P.state = 1,\ rule\ 1$	$R.state() = 1$ $Q.state = 1$ $S.state = 1$ $P.state = 1,\ C_j(fulfilled, a, b, P\|S)$ $C_i(fulfilled, b, a, Q\|TRUE)$

6.6 Conclusions

In this chapter the problem of defining the semantics of standard Agent Communication Languages has been analyzed. An operational proposal for the definition of the semantics of communicative acts using the notion of commitment has been presented. Important properties and possible use of the proposed semantics has been discussed. In particular the use of commitments has the important function to stabilize the interaction among agents, through the creation of an expectation about other agents' behavior. Moreover some examples to show the detailed dynamic evolution of the system, as consequence of the performance of various speech acts, have been reported.

A first important feature of our proposal is that it is operational, in fact to define the meaning of crucial communicative acts we gave the complete design specification of the various components necessary to implement an interaction system of multiple agents able to communicate. Whereas many of the semantic frameworks proposed and discussed so far [165, 22, 152, 30], are based on formal tools that are typical of academic research, but are not well known to the larger community of software professionals. In any case even if the operational definition of commitment is a very important step for the realization of practical applications, a thorough treatment of commitment-based semantics requires the development of a full *logic of commitment*. This important theoretical aspect is currently under development.

Another important characteristic of our proposal is that it is modular and allows for the reuse of various components. Starting from a small set of basic operations on commitments, is possible to define the meaning of simple speech acts, which in turn can be used to define a new layer of more complex speech acts, and so on. In this way, ACL messages are given a formal semantics, thus eliminating any ambiguity in their use.

In particular, we believe that an interesting contribution of this work is the definition of the proposal and of the offer act as a combination of a request and of a promise, two fundamental communicative acts in electronic commerce applications.

In the presented model temporal aspects are only sketched, we plan to develop this point in the future. In future studies we plan also to extend the model in order to cover some other speech acts, in particular orders and commands treated in hierarchical contexts. Moreover the expressiveness of the model may be increased with the addition of new methods to the commitment class to let interacting agents to fully negotiate the content and the condition of a commitment object. For the moment only the "add condition" method to add a new temporal proposition object to the condition of a commitment object has been introduced, it is a fundamental method in the definition of the "conditional accept" communicative act, an act necessary when the hearer would like to accept to perform what the speaker has requested but on more restrictive conditions. Finally an important point, which will be introduced in next chapter, is the concept of soundness of interaction protocols.

Our approach is strongly related to the one proposed by Yolum and Singh [189] even if their focus is on the definition of interaction protocols. However, between the two approaches there are some significant differences. First, our proposal provides a more complete account of how different types of speech acts can be defined in terms of operations on commitments. Our proposal makes possible to keep trace of the complete dynamic evolution in time of social commitments in order to detect their violations and be aware of their fulfilment. Second, and more important, we also provide a commitment-based analysis of directive speech acts, like requests, relying on the notion of pre-commitment. Third we introduced the treatment of declarations, a particular category of communicative acts that to be modelled requires the introduction of the notion of institutional world. It is important to remark that the complete formalization of an electronic institution is a very challenging research topic. While P. Yolum and M. P. Singh, in their approach, provide a formal language $\mathcal{P}$ for the treatment of commitments, but their treatment of the evolution of the state of commitments and of the definition of the meaning of various speech acts is limited to some examples. Another difference is that in our approach we have started to make some hypothesis, which will be further developed in future studies, on how to treat the notion of time presents in the content and the condition of commitments, while they did not face this aspect. Finally the two approaches differ in their formalization of the conditional commitment. Till now they have used a monadic deontic operator C and the special operator "$\leadsto$" to formalize conditional commitments, even if in their most recent paper on this subject they have started to adopt a dyadic conditional commitment operator [188], a formalization similar to the one that we have adopted.

As regard as the introduction of a state in the commitment class to keep trace of its dynamic evolution, Dellarocas in [41] proposed a similar approach (even if he did not study it in depth) but he delineates a situation where every contract has its own state transition graph with its state transition rules which in particular describes the sanction associated with each state transition. This approach, from my point of view, has the limitation to require very sophisticated agents able to take decision on complex concepts, while in our approach also very simple reactive agents can take part to the interaction.

Chapter 7

A Method for the Definition of Interaction Protocols

Interaction Protocols are patterns of behavior that agents have to follow to engage in a communicative interaction with other agents within a multiagent system (MAS). The specification of interaction protocols is crucial for the development of a MAS: in fact the advent of Internet makes it urgent to develop general, application-independent methods for the definition of interaction protocols, to be used as components of open, dynamic, heterogeneous, and distributed interaction frameworks for artificial agents. Indeed, the definition of new interaction protocols is a critical task, because a badly designed protocol may lead to unsuccessful interactions; thus there is a need for general methods, criteria, and tools for protocol design. We think that there are some important properties that interaction protocols for open frameworks have to satisfy. In particular, an interaction protocol should:

- Specify legal sequences of communicative acts that form a complete interaction within a system. Every communicative act used in a protocol should maintain its meaning, as defined in a general, application-independent communicative act library of a standard Agent Communication Language (ACL).

- Enable interactions among purely reactive agents, that blindly follow a given protocol, and deliberative agents, that are able to reason about the consequences of actions, and decide whether to take or not to take part in an interaction.

- Allow for effective verification that agents behave in accordance to the specifications of the interaction protocol.

- Moreover, a general method for the development of interaction protocols should allow a designer to verify whether a protocol is "sound" with respect to general, application-independent soundness criteria.

So far, several approaches to the definition of interaction protocols have been proposed. Some authors define interaction protocols as finite state machines or Petri nets (see for example [46] and [108]), but do not take into account the meaning of the exchanged messages, which in our opinion is crucial to obtain the properties listed above. Other approaches take into account the meaning of the exchanged messages, but do not rely on a standard ACL with application-independent semantics; for instance Esteva *et al.* [54] specify the protocols available in an electronic institution using finite state machines, but define the meaning of only some of the message types using *ad-hoc* rules. An example of interaction protocol specification which fully takes into account the meaning of the exchanged messages is proposed by Yolum and Singh [188], who introduce a method based on event calculus to define protocols that may be used by artificial agents to determine flexible paths of interaction complying with the specifications. The main difference between Yolum and Singh's proposal and the one put forward in this chapter is that with the method described here all the preconditions and effects of the performance of communicative acts on the state of the interaction are completely specified; we also propose a method through which protocol designers may verify if a protocol is sound with respect to a number of general, application-independent soundness criteria related also to the meaning of the exchanged messages.

In this chapter a *verifiable* and *application-independent* method for the definition of interaction protocols to be used in open multiagent systems and that satisfies the properties listed above is proposed. In particular in Section 7.1 the method is described in detail. Of course, an arbitrary collection of rules does not necessarily define a reasonable interaction protocol. We therefore propose in Section 7.2 a set of application-independent and verifiable *soundness conditions*, which guarantee that protocols possess certain properties that are crucial for a successful interaction. Such conditions are expressed in terms of the content of the system state at each stage of the interaction, as consequence of the performance of communicative acts. In Section 7.3, the method proposed is exemplified by defining a widely used electronic commerce interaction protocol: the English auction protocol. In particular we choose to formalize the English auction protocol for various reasons: first such a protocol is the one actually used in the Trading Agent Competition to sell hotel rooms (see Appendix A) and is widely used in many auction web sites, second because its, in our opinion, more complex than the other auction types present in the TAC game, third it is an interesting example of iterative protocol. Finally in Section 7.4 some conclusions about the proposed method will be drawn.

The content of this chapter is mainly based on the paper "Defining Interaction Protocols using a Commitment-based Agent Communication Language" [62] by Fornara and Colombetti.

7.1 Definition of Interaction Protocols

In this Section an application-independent method for the definition of interaction protocols to be used in open interaction systems is proposed. Such a method is verifiable, that is, given a specification it is possible to check if an agent is following the protocol and satisfies the important properties listed above.

The method proposed is based on the assumption that the fundamental component of every interaction is the *institution of language* and that the specification of interaction protocols fully rely on the application-independent meaning of communicative acts. Starting from the commitment-based semantics for agent communication languages proposed in the previous chapter the method proposed consists of the following steps:

1. definition of the meaning of every single communicative act present in the protocol following the commitment-based semantics presented in Chapter 6. This is possible exploiting the property of the proposed commitment-based semantics to be extendible. It is important to remark that all communicative acts preserve their general meaning when used within a protocol;

2. definition of the *environment* where the interaction takes place, which is composed by the elements described in Section 6.2 that are necessary for the definition of the meaning of communicative acts and by some new further components necessary for the specification of the protocol;

3. specification of the *interaction diagram* of the protocol which describes the actions that may be performed by agents at each stage of the interaction;

4. identification of specific conditions (*guards*) that have to be satisfied in order that the effect of performance of some communicative acts used in the protocol takes place. Obviously in order that the communicative act is successfully performed also the preconditions defined in the library of communicative acts have to be satisfied.

The Environment

In particular, a protocol's environment defines:

- A nonempty set of *roles* that agents can play in the interaction. To each role, a set of specific authorizations may be associated.
- A nonempty set of *participants*, which are the agents interacting by using the protocol. Every participant must play a well-defined role in the interaction. The set of participants may vary during the execution of the protocol, but is always finite.
- A possibly empty set of global *constants* and *variables*, that may be subject to global *constraints*.
- Besides the set of communicative acts belonging to a library of communicative acts, a set of domain specific actions that agent can perform.
- A collection of commitment objects, temporal proposition objects, and domain-specific objects used to represent all entities involved in the interaction.
- A set of *authorizations*, associated to roles, to perform certain institutional actions, in particular declarations.

Interaction Diagram

A protocol's interaction diagram specifies which actions may be performed by the agents at each stage of the interaction. More precisely, an interaction diagram (see for example Figure 7.1) is defined by a finite graph in which:

- Every node represents a state of the interaction. To every state we can associate a representational content, that is, the set of all facts that hold at the state, expressed in terms of: protocol variable values, commitment objects, temporal proposition objects, and domain-specific objects.

- There is a single distinguished *initial node*, with no incoming edge, and a set of distinguished *final nodes*, with no outgoing edge and marked by double blocks. The interaction starts from the initial node and ends when a final node is reached.

- Every edge describes a transition from a state to another state. A transition may correspond to the execution of a communicative act or to the occurrence of a relevant environmental event. When the transition occurs because of a communicative act is performed the content of the target state can be completely computed from the content of the source state, and from the semantics of the communicative act; whereas when an environmental event is responsible for the transition the content of the target state can be completely computed from the content of the source state, and from the application of the relevant update rule (see Section 6.2.4) fired by the event.

- When more than one communicative-act edge goes out of a given node, it is possible to specify the conditions (defined as arbitrary Boolean expressions) under which each act may be executed. As a whole, the set of condition-action pairs going out of a node behaves like a *guarded command* [45]: at least one of the actions must be executed, but the agent specified as the actor of the action is free to choose which action to perform among those whose *guard* is true. If all guards are mutually exclusive, the guarded command is equivalent to a sequence of if-then statements.

- It is possible to associate a cardinality to communicative act edges. In particular cardinality "1 to n" means that the same message is sent by one agent to n agents, and cardinality "1 to 1" means that a message is send by one agent to another agent. Declarations are particular communicative acts without a specific hearer, or better the hearer is the interaction system as a whole, in fact the effect of declarations is to change institutional facts resulting from a common agreement among the agents, or better of their designers. For this reason declarations have not an associate cardinality. A uniform approach for all communicative acts could have been followed if the entire system had been formalized as a "special" artificial agent, but studies about the feasibility and consistency of such an approach are beyond the scope of this thesis.

7.2 Soundness Conditions

To describe a sensible interaction pattern, an interaction protocol must satisfy a number of general, application-independent *soundness conditions*. A first, fairly trivial, set of conditions concerns the topology of the interaction diagram:

- Every node of the interaction diagram must be reachable from the initial node.
- There must be at least a final node.

Another, less trivial, set of soundness conditions concerns the content of states. Such conditions, contrary to existing approaches based on Petri net [108], express constrains related to the meaning of the exchanged messages, as defined by the communicative act library adopted.

- All communicative acts that are allowed by a protocol at state s must have their preconditions satisfied by the content associated to s when their guard is true. This condition guarantees that all communicative acts allowed by the interaction protocol may actually be executed.

- All commitments included in the content of a final state must be cancelled, fulfilled, or violated. This condition guarantees that the whole interaction has been completed.

An interesting problem is raised by the fact that during the execution of a protocol, the same state may be reached from the start state following different paths (i.e., performing different chains of actions). For example, a certain state of an interaction could be reached because an agent has autonomously made a promise or because the agent was requested to make a promise, accepted the request, and then fulfilled the resulting commitment by actually making the promise. If we abstract from the different paths, we intuitively feel that the interaction has reached the same state; however, if we compute the content of the state we get different results. The point is that these results, although different, are equivalent from the point of view of the interaction, in that they have the same "commissive import". More precisely, we say that state s is *equivalent* to state s' if and only if the contents of s and s' are identical, with the only exception of commitments that are fulfilled, violated, or cancelled. We can therefore formulate another soundness condition:

- If a state of an interaction can be reached through different paths, the contents of the state computed along the different paths must be equivalent.

The situation is even more complex when the definition of an interaction protocol has a loop, that is, a cycle in the interaction diagram. Interaction loops naturally appear when a sequence of communication acts can be repeated several times, like for example in the English Auction Protocol (Section 7.3). The existence of loops makes it possible to reach the same state following different paths in the interaction diagram. In this case, however, the notion of equivalence discussed above is still necessary but no longer sufficient. This problem is well known in computer programming and it can be solved by introducing the concept of a *loop invariant*. For example, consider again the protocol for an English Auction. At a generic iteration, the auctioneer is committed to selling the product currently under the hammer to a specific agent for a specific price, on condition that no higher price will be offered. Of course, the specific agent that made the highest offer, as well as the associated price, will change from one iteration to another one. However, we can describe the situation in terms of loop invariants, by saying that the auctioneer is committed to selling the product to *the agent that made the highest offer*, for *the price defined by such an offer*, on condition that no higher offer will be made. The soundness condition given above can now be reformulated as follows:

- If a state of an interaction can be reached through different paths, the contents of the state computed along the different paths, expressed in terms of suitable invariants, must be equivalent.

7.3 The English Auction Protocol

In this section we present a specification of a form of English auction protocol using the framework proposed so far. We chose this protocol as an example because it is used in many electronic commerce applications on the web and also in the TAC game (see Section 4.3.1), and because it is fairly complex: in particular, it is an interesting example of iterative interaction protocol. Given that the English auction protocol is much more complex than the protocols discussed in the previous chapter, in particular because it is iterative, the contents of all possible states of the interaction will not be reported in detail but can be easily computed using the definition given in Section 7.3.2. In this example we consider the interaction process needed to sell a single product o, which can obviously be repeated to sell several products.

7.3.1 The Environment

The environment of the English Auction Protocol includes the following elements:

- **Roles**. *Auctioneer* and *Client*.

- **Participants**. One agent, a, in the role of *Auctioneer* and n agents, $\{p_1, ..., p_n\}$, in the role of *Client*.

- **Constants and Constraints**. t_{max}, the maximum duration of the auction; t_1, the deadline for the payment; t_2, the deadline for the delivery. $t_{max} < t_1 < t_2$.

- **Domain-specific objects and actions**.
  Object o, representing the product on sale, with a $resPrice$ field for the reservation price.
  Object A, representing the auction, with fields for the following variables (initialized as indicated): $state$ = "*closed*"; $askPrice$ = 0; t_{end} automatically set to t_{system}

when *state* is set to "closed"; $t_{inactivity}$ i.e. the maximum time of inactivity between two subsequent bids.

The action of transferring the ownership of an object or of a sum of money to another agent. It is an institutional action involving the institutional notions of ownership and money. For the sake of simplicity, we treat here this action as a primitive domain action. The fact that agent a transfers to agent b the ownership of x (an object or a sum of money) is represented by $give(a, b, x)$.

- **Variables** The environment has the following variables (initialized as indicated): $newAskPrice = o.resPrice()$; $value_{win} = 0$; t_{system}, a global clock accessible to all participants; $t_{bid} = 0$, the time of the last accepted bid; i a counter that is automatically incremented every time the bidding process is iterated.
- **Authorizations**. The auctioneer is empowered to open and close the auction and to set the ask price of the product on sale:
$empowered(Auctioneer, A.setState())$,
$empowered(Auctioneer, A.setAskPrice())$.

Scheme of Temporal Proposition Objects

In the interaction framework proposed so far the *content language* used to express the content and the condition fields of commitment objects is based on the use of temporal proposition objects. Given that in complex interactions like the ones that follow the English Auction Protocol many temporal proposition objects are involved, we concisely describe them through *schemes*, that represent possible temporal proposition object in parametric form. Parameters will be bound to specific values when the interaction actually takes place and an instance of the temporal proposition object is created (truth values are always initialized to $\perp$, and therefore are not indicated in schemes). In our example, parameter *now* is initialized at t_{system} when the temporal proposition object is created; and parameter v and v' are used to indicate an amount of money.

- Scheme P_j represents the proposition "the auctioneer a gives product o to client p_j, in the time interval from the end of the auction to t_2":

$$P_j(give(a, p_j, o), t_{end}...t_2, \exists);$$

- Scheme $Q_{j,v}$ represents the proposition "client p_j gives the amount v of money to the auction house represented by the auctioneer a, in the time interval from the end of the auction to t_1":

$$Q_{j,v}(give(p_j, a, v), t_{end}...t_1, \exists);$$

- Scheme $S_{j,i}$ represents the proposition "client p_j makes a proposal during iteration i":

$$S_{j,i}(propose(p_j, a, P_j, Q_j(\mathbf{x})), now...now + t_{inactivity}, \exists);$$

- Scheme $U_{j,v}$ represents the proposition "the auctioneer a is committed, relative to client p_j, to proposition P_j under condition $Q_{j,v}$, in the time interval from now to the end of the auction":

$$U_{j,v}(C_{id2j}(pending, a, p_j, P_j|Q_{j,v}), now...t_{end}, \exists);$$

- Scheme $W_{v'}$ represents the proposition "the auctioneer a does not accept any proposal with value greater than v' in the time interval from now to the end of the auction":

$$W_{v'}(\neg\exists j\ (condAccept(a, p_j, C_{id2j}(unset, a, p_j, P_j|Q_{j,v}), W_v) \wedge\ v > v')), now...t_{end}, \exists);$$

7.3.2 Communicative Acts and Guards

In this section specific conditions (guards) for the performance of the communicative acts used in the English Auction Protocol are given. Obviously in order that the communicative act is successfully performed also the preconditions defined in the Library of Communicative Acts have to be satisfied. Some of these communicative acts have to be repeated at every round i of the bidding process. In order to be able to refer to commitment objects created by previously preformed communicative acts we report also the effects of the performance of communicative acts: they can also be computed from the definitions given in the library of communicative acts. Moreover the performance of certain communicative acts changes the value of some environmental variables as reported below. The names of the states written between brackets refer to Figure 7.1.

- The auctioneer declares the auction open (state s_0).

$$guards : A.state() = "closed"$$
$$\mathbf{declare}(a, A.state = "open")$$

- The auctioneer declares the current ask-price of the ongoing auction (states s_1, s_6, s_{11}).

$$guards : A.askPrice() < newAskPrice$$
$$\mathbf{declare}(a, A.askPrice() = newAskPrice)$$

- The auctioneer makes the "call for proposals" (states s_2, s_5, s_7, s_{10}).

$$\mathbf{request}(a, p_j, S_{j,i})$$
$$effects : C_{id1j}(unset, p_j, a, S_{j,i})$$

- One participant makes its proposal (states s_3, s_8).

$$guards : \{(t_{system} < t_{max}), (t_{system} - t_{bid} < t_{inactivity})\}$$
$$\mathbf{propose}(p_j, a, P_j, Q_{j,v})$$
$$effects : \{C_{id2j}(unset, a, p_j, P_j|Q_{j,v}), C_{id3j}(pending, p_j, a, Q_{j,v}|U_{j,v}),$$
$$S_{j,v}.truth\_value() = 1\}$$

- If the value of the proposal is greater than the current ask-price the auctioneer has to accept it (states s_4, s_9).

$$guards:\ v > A.askPrice();$$
$$\textbf{condAccept}(a, p_j, C_{id2j}(unset, a, p_j, P_j|Q_{j,v}), W_v)$$
$$effects: \{C_{id2j}(pending, a, p_j, P_j|[Q_{j,v}, W_v]), \forall\, v' < v\ W_{v'}.truth\_value() = 0\}$$
$$variable\ updates: \{newAskPrice = v,\ t_{bid} = t_{system}\}$$

- If the value of the proposal is less than or equal to the current ask-price the auctioneer has to reject it (states s_4, s_9).

$$guards:\ v \leq A.askPrice();$$
$$\textbf{reject}(a, p_j, C_{id2j}(unset, a, p_j, P_j|Q_{j,v}))$$
$$effects: \{C_{id2j}(cancelled, a, p_j, P_j|Q_{j,v}), U_{j,v}.truth\_value() = 0\}$$

- The auctioneer can declare closed the auction only if the time of inactivity is equal to the constant value defined at the beginning of the auction or when the fixed end time of the auction is reached (states s_3, s_8).

$$guards: \{(t_{system} \geq t_{max}) \vee (t_{system} - t_{bid} > t_{inactivity}), A.state() = "open"\}$$
$$\textbf{declare}(a, A.state = "closed")$$
$$effects: \{W_{value_{win}}.truth\_value() = 1, U_{j,value_{win}}.truth\_value() = 1\}$$
$$variable\ updates: \{t_{end} = t_{system}, value_{win} = newAskPrice\}$$

7.3.3 Interaction Diagram

The interaction diagram that specifies the English auction Protocol is reported in Figure 7.1. Writing down the content of each state of the interaction it is possible to prove the soundness of this protocol specification with respect to the soundness condition detailed above. In particular the contents of each state accessible through different paths results equivalent. For states s_9, s_{10}, s_{11}, that are in the loop of the protocol, it is possible to identify a loop invariant describing p_j as the client who made the highest offer. That is, in such states the commitment objects, at every iteration of the bidding process, change only in the fields that refer to the participant who made the highest bid. For example if the current highest bidders is participant p_1 in the content of state s_9 there are the following commitment objects (where i is the index of the iteration:

$$C_{id11}(fulfilled, p_1, a, S_{1,i}),$$
$$C_{id21}(unset, a, p_1, P_1|Q_{1,v_1}),$$
$$C_{id31}(pending, p_1, a, Q_{1,v_1}|U_{1,v_1}).$$

Next, if in another iteration of the bidding process the highest bidders becomes participant p_2 the content of state s_9 becomes:

$$C_{id12}(fulfilled, p_2, a, S_{2,i}),$$
$$C_{id22}(unset, a, p_2, P_2|Q_{2,v_2}),$$
$$C_{id32}(pending, p_2, a, Q_{2,v_2}|U_{2,v_2}).$$

7.4 Conclusions

In this chapter we presented an application independent method for the definition of interaction protocols, based on the meaning of the exchanged messages, that can be used to define patterns of interaction in open, dynamic, and heterogeneous agent systems. The method proposed is based on a application-independent ACL, whose semantics is defined in terms of commitments, and on a further component defining protocol-specific interaction rules. The resulting interaction protocols are verifiable, in the sense that is possible to test whether an agent is behaving in accordance to it.

Moreover, a number of soundness condition are proposed to verify if a the structure of a given interaction protocol is reasonable. Such conditions are stated relying only on intuition and without any deep theoretical justification. We believe, however, that it is possible to derive such conditions from a set of general interaction principles expressed in terms of *conversational commitments*, that is, of meta-level commitments that regulate conversational interactions among agents. A similar problem is addressed in [163]. To gain a more concrete idea of this concept, consider the *time-out* field of the commitment class. This field represents the time limit for the debtor of an unset commitment to accept, fulfill, or reject it. In fact, it is possible to view the time-out of an unset commitment as the *deadline* of another commitment, namely the conversational commitment that every agent makes when it decides to take part in an interaction, and that binds the agent to react to the speech acts performed by the other participants. The systematic development of this concept is an aim of our further research.

We also show how our method can be effectively used to define in a flexible way a complex, iterative, and widely used interaction protocol: the English Auction Protocol and check if it verifies the soundness conditions defined in terms of speech acts.

Our method for the definition of interaction protocols differs from most existing proposals, in that it is based on the use of an application-independent library of communicative acts, whose meaning is fully preserved when they occur in a protocol. With respect to the proposal put forward by Yolum and Singh in [188], our approach is focussed on the protocol design phase more than on the possibility of shortcutting predefined interaction patterns at run time. Indeed, we expect that agents used in practical application will mostly be simple reactive agents; if this idea is correct, proving the soundness of a protocol at design time is more important than allowing agents to plan intelligent variations of existing protocols. In principle, however, a deliberative agent with reasoning capabilities could understand our protocols on the basis of an ontology of commitment, linguistic

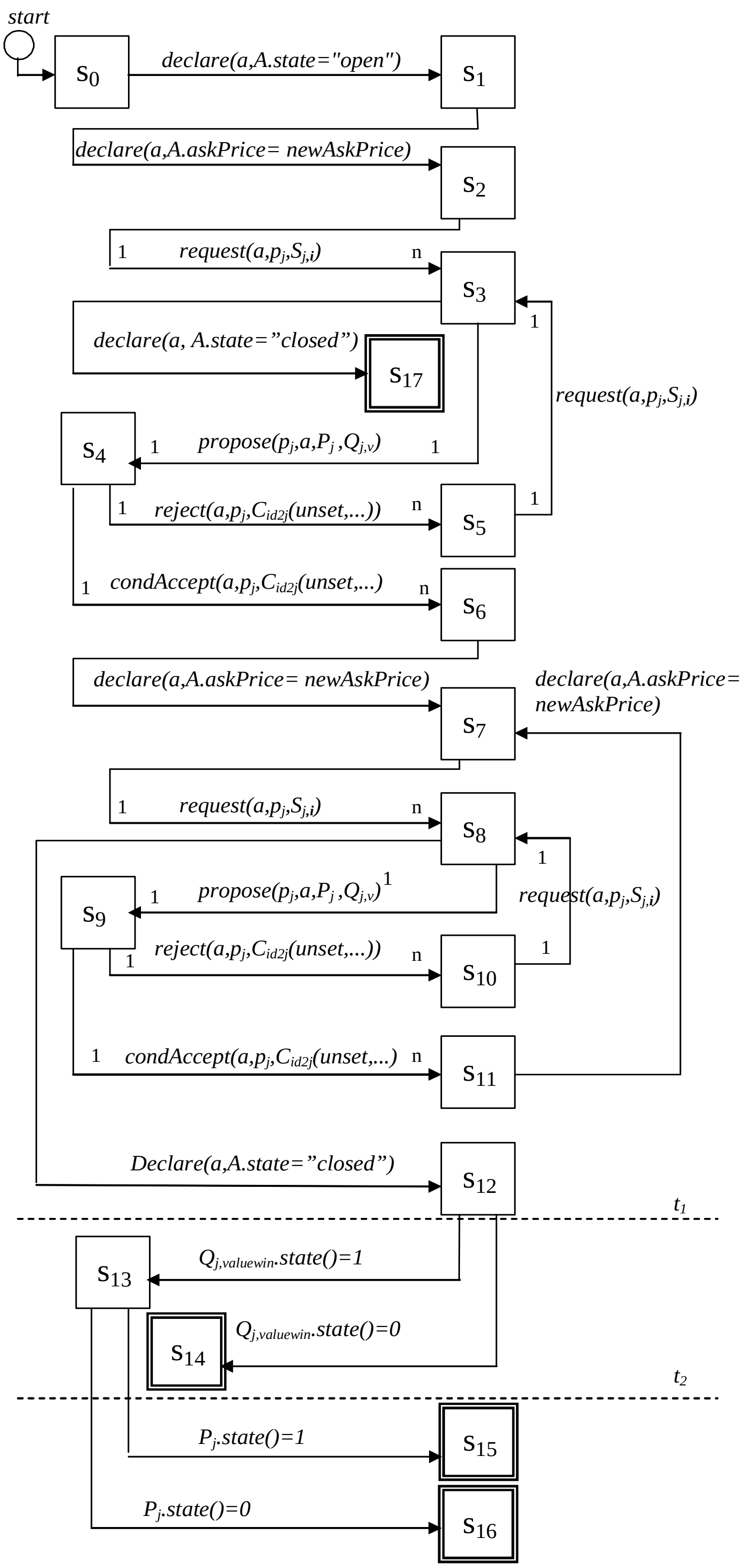

Figure 7.1: Interaction diagram of the English Auction Protocol.

knowledge (i.e., knowledge of a Communicative Act Library with semantics), and the ability to reason on interaction diagrams (i.e., a version of finite state machines).

Concluding we think that the definition of the commitment-based semantics of ACLs and the definition of an application-independent method to formalize interaction protocols are the two fundamental components of every open interaction system. An initial description of our ongoing research about the definition of a complete social framework to describe agent interactions, which includes the concept of institutions, roles and registration rules, interaction rules and authorizations can be found in [33].

Chapter 8

Conclusions

8.1 Contributions

In recent years the advent of Internet faced computer scientists with new technological challenges, one in particular is the problem of defining open, dynamic, and distributed systems where autonomous and heterogeneous agents, usually having different owners, may interact exchanging services, resources, information, and may form dynamic coalitions for example to span organizational boundaries.

This problem has received much attention in the last few years but the solutions proposed so far are considered unsatisfactory by a large number of specialists. In particular, most current proposals are unable to support heterogeneity of agents, verifiable compliance to standards and accurate delimitation of responsibility for action, that are all crucial properties when interacting agents belong to different companies and usually are competitive or have conflicting goals.

In order to carry out the research work presented in this thesis we started to analyze in detail a fundamental field of application of methods for the definition of open interaction frameworks: electronic commerce. During this phase we studied the structure and the main components of an existing but not open interaction system for testing trading artificial agents: the Trading Agent Competition (TAC) System [175]. In order to understand better which are the critical aspects of an interaction system like that we decided to take part to the competition by developing a trading artificial agent.

The main contributions of this phase of our research work are two. First the definition and evaluation of bidding policies for buying complementary and substitutable products in parallel auctions based on the idea of maximizing the expected utility of each possible available action [63]. In particular an important aspect of the proposed approach is that it is not tailored to the particular auction mechanism of TAC. As a result, the implementation of our algorithm required some strong simplifying assumptions. Nonetheless, our agent overall performance illustrates the promise of this general method. Second we understand better which are the main limitations of an interacting environment of that kind. In particular it is a closed environment, in the sense that only agents designed for that system can interact with it through a predefined limited set of methods. Given this limitation the same trading agent cannot actually bid in different market places to get a certain product at best price. Moreover given that there is not a formal way to express the meaning of each message available in the system we have had some troubles in interpreting them and in understanding the rules of the various types of auctions. Another limitation in developing a trading agent to interact with such a system is that it became useless when the rules of the game and the interface of the system have been changed in the following years.

In the second phase of the research work presented in this thesis we concentrate our studies on the problem of defining a method for modelling in a standard and objective way open interaction frameworks for artificial agents. Following John Searle's philosophical studies on the construction of social reality [143] we think that the main component of every interaction system is the definition of an Agent Communication Language (ACL). The two agent communication languages extensively used at the moment are: KQML [80, 57] and FIPA ACL whose latest delivered specification can be found at FIPA web site [60]. For example JADE (the Java Agent DEvelopment Framework), a tool for the implementation of multiagent systems, is comply with the FIPA specifications and many agent platforms in the Agentcities Network, a global, collaborative effort to construct an open network of on-line systems hosting diverse agent based services, comply with FIPA ACL to enhance interoperability.

But as far as we know these two languages are used as a standard to define the syntax of the exchanged messages, while their semantics based on agent mental states is not actually used to develop agent able to reason about other agents actions or to test if an agent is behaving in accordance with its previously performed communicative acts. This problem arises mainly because of the complexity of such a test.

The main contribution of this thesis is the operational definition of the semantics of a verifiable and objective Agent Communication Language based on the notion of social commitment [61]. Such language is extensible to allow for the reuse of previously defined communicative acts to define the meaning of new ones. It is simple to be understood by agent designers but enough expressive as is demonstrated by examples of its use to define the meaning some communicative acts commonly used in electronic commerce applications, like for example the proposal and the offer act. Moreover given that the proposed semantics is based on Speech Act Theory it can be used also to enhance human and artificial agents communications in order to tackle a very important question on how a task can be delegated from a user to an artificial agent.

The second fundamental component, from our point of view, of every open interaction framework is the definition of protocols of interaction to let simple reactive agents and complex deliberative agents to interact following the rules defined by the interaction system. Starting from the definition of the semantics of an ACL based on the notion of commitment we propose an application independent method for the definition of flexible interaction protocols, in order that it could be possible to test if an agent is following the given protocol. Furthermore given that not every patter of interaction is a sensible interaction we propose also a set of reasonable soundness conditions to test if the definition of a given protocol is sound with respect to the meaning of the exchanged messages [62].

The proposed method differs from most existing proposals, in that it takes into account the meaning of the messages and is based on the use of an application-independent library of communicative acts, whose meaning is fully preserved when they occur in a protocol.

Finally in order to test the proposed method and to tackle the problem of defining an electronic commerce interaction systems with multiple auctions, we showed how our method can be effectively used to define in a flexible way a complex, iterative, and widely used interaction protocol: the English Auction Protocol and we checked that it verifies the soundness conditions defined in terms of speech acts.

8.2 Future Works

The proposal for the definition of an operational model to formalize open, dynamic, and heterogeneous interaction frameworks presented in this doctoral thesis to become effective has to be extended in several directions.

Starting from the two main components presented: the definition of an operational semantics for agent communication languages and a method to define and test interaction protocols and to verify their soundness, studies about how to define completely the social context of an interaction system has to be performed. The concepts that need to be better investigated are:

- the definition of roles through the concepts of authorizations, permissions, and obligations, the relation among roles in order to express authority, and the evolution of roles;
- the extension of basic communicative acts to cover for example the possibility to perform orders;
- the study of the limitations and possible improvements of the proposed content language;
- the possibility to use the proposed framework in different kinds of interactions: from negotiation by means of protocols to argumentations. In this case it will be crucial for interacting agents to be able to negotiate the content and the condition of a commitment where the debtor and the creditor are already set;
- the problem of defining contracts and agreements entered into by agents, understood as sets of interdependent commitments. In particular a contract is an agreement certified by a suitable institution and the difference between a contract and an agreement is strictly correlated with the idea to have different levels of violation of a single social commitment.

An important distinction that has to be pointed out in the definition of open interaction systems is the one between two possible meanings of the word *institution*. In one interpretation an institution can be defined as a set of shared concepts and rules, defining a fragment of social reality, like for example the fundamental institution of language, the institution of conversations, and the institution of property. In another interpretation an institution can be defined as an organization of social entities deliberately constructed and reconstructed to seek specific goals. Starting from this distinction we think that there are at least three level of abstraction in the definition of open agent interaction systems:

- the definition of institutions in general;
- the definition of a specific version of an institution;
- the actual realization of a specific organization that *reify* the specific institution. This means that the interactions within the system rely on the social ontology defined in the related specific institution.

For example given the general institution of auctions it is possible to define a specific version: the institution of English auctions, finally an auction house reifies the specific version of the institution of auctions. Besides the definition of electronic institutions at the different levels of abstraction described above, there is the problem of how to model their relations and interdependencies at different levels. Some of our preliminary studies and considerations on electronic institution can be found in [33, 34].

Another important future work is the formalization of such a conceptual model of open interaction systems to obtain an operational description of agent communicative interactions. Given that the proposed conceptual model is based on the notion of object oriented programming a possible approach is to formalize it using the Unified Modelling Language (UML) [14]. It may be possible that such a modelling language has to be extended to be used in practical applications to specify and develop open interaction environments for multiagent systems.

Finally a demonstrative implementation and testing of the proposed operational definition of interaction frameworks has to be developed. We think that if will be possible the demo will be implemented using an existing development environment like for example JADE (Java Agent Development Environment, a FIPA-compliant agent framework [9]). Following the approach chosen in Chapter 7 such an implementation might be placed in the area of electronic business applications.

Other important research aspects correlated with this work are: (i) the study of a model to define commitment-based agents able to reason about their and other actions on the basis of the proposed language, (ii) the definition of mechanisms to let agents to discover new interaction systems and to register into the system, and (iii) the definition of an appropriate super parties entity to treat different level of violations: violation of the semantic of the language, violation of interaction protocols, or violation of commitments.

Appendix A

TAC game Auction Types

A schematic description of the three auction types used in the Trading Agent Competition.

A.1 Flights

number of sellers: single seller auction;
number of auctions: one auction for each day and direction;
time of clear: the auctions will clear continuously;
clear price: if the current quoted price is less than your bid, you will get the ticket at the current price. If the current quoted price is greater, your bid will sit in the auction until the price becomes less than or equal to your bid, at which point you will get it for the price you bid. Of course, this may never happen if the price never goes that low. Matching is based on earliest submission time;
time of price quote: quotes are continuous;
price quote: the ask quote is set according to a stochastic function. The process used to update flight prices is a random walk, starting between $250 and $400 and perturbed by -$10 to $10 every 30 to 40 seconds. Prices will always be in the range $150 to $600. All distributions are uniform;
availability of goods: unlimited;
bid admittance conditions: all buy bids are accepted, multi-point buy bids are allowed;
bid withdrawal conditions: bids can be withdrawn at any time if they have not transacted;
reservation price: the ask quote;

A.2 Hotel Rooms: English Ascending Auction, M^{th} Price

auction type: standard English ascending auction, M^{th} price;
number of sellers: single seller auction;
number of auctions: there are two hotels in Boston: the Boston Grand Hotel and Le Fleabag Inn, so there is one auction for each combination of hotel and night.
time of clear: it will clear at the earliest, 1 minute after the game has started or at latest when the game ends. It might clear earlier if there has been no activity (no bids) for a

random chosen period of time;
clear price: is the price of the lowest winning bid, matching is based on earliest submission time;
time of price quote: continuous quotes;
price quote: the ask quote is the 16th highest price;
availability of goods: 16 rooms;
bid admittance conditions: must beat quote, multiple buy points in bid allowed;
bid withdrawal conditions: no withdrawal, but since the hotel auctions are ascending, once a bid is "out of the money" (i.e., the ask price is above the offer) it is effectively decommitted;
reservation price: no reserve price, minimum bid is $0;

A.3 Entertainment Tickets: Continuous Double Auction

auction type: continuous double auction (CDA);
number of sellers: multiple seller;
number of auctions: there will be one auction for each event-night combination. time of clear: continuous clears;
clear price: when a new bid is processed, the auction check whether the offered price would match the lowest (best) existing sell bid, and vice versa. The transaction price is the price that was put in first;
time of price quote: continuous quotes;
price quote: the ask quote is the price an agent would have to bid over in order to place a winning buy bid (the least of the sell bids). The bid quote is the price an agent would have to bid under in order to place a winning sell bid (the biggest of the buy bids);
availability of goods: depends on the seller;
bid admittance conditions: any type of bid allowed;
bid withdrawal conditions: bids in the entertainment auctions can be withdrawn at any time if they have not transacted;
reservation price: no reserve price;

Appendix B

Truth-Tables for Temporal Proposition Objects

Truth-tables of Boolean operators of propositional logic can be used to obtain the extended truth-tables of the same operators when the propositions can be: true (1), false (0), or undefined({0,1} or $\perp$).

For example, the truth table of the *and* operator ($\wedge$) is shown in Table B.1, while the extended truth table of the same operator is shown in Table B.2. The second truth-table is obtained from the first one writing {0,1} when it is impossible to write only one value. For example {0,1} $\wedge$ 0 gives 0 because the column of 0 in Table B.1 has only 0s, while {0,1} $\wedge$ 1 gives {0,1} because the column of 1 in Table B.1 has both 0 and 1.

$\wedge$	0	1
0	0	0
1	0	1

Table B.1: Truth table of the "and" operator

$\wedge$	0	1	{0,1}
0	0	0	0
1	0	1	{0,1}
{0,1}	0	{0,1}	{0,1}

Table B.2: Extended truth table of the "and" operator

Bibliography

[1] L. Åqvist. Deontic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of philosophical logic II*, pages 605–714, Reidel, Dordrecht, 1984.

[2] ALFEBIITE. Project presentation. http://alfebiite.ee.ic.ac.uk/, 1999.

[3] J. F. Allen. Toward a general theory of action and time. *Artificial Intelligence*, 23(2):123–154, 1984.

[4] J. F. Allen, J. Hendler, and A. Tate. *Readings in Planning.* Representation and Reasoning Series. Morgan Kaufmann, San Mateo, California, 1990.

[5] J. L. Austin. How to talk: Some simple ways. In J.O. Urmson and G.J. Warnock, editors, *Philosophical Papers.* Oxford University Press, Oxford, 1961.

[6] J. L. Austin. *How to Do Things With Words.* Oxford University Press, Oxford, 1962.

[7] R. Axelrod. *The Evolution of Cooperation.* Basic Books, New York, 1984.

[8] M. Becht, T. Gurzki, J. Klarmann, and M. Muscholl. ROLE: Role Oriented Programming Environment for Multiagent Systems. In *Proceedings of the Fourth International Conference on Cooperative Information Systems (CoopIS'99)*, pages 325–333, 1999.

[9] F. Bellifemine, A. Poggi, and G. Rimassa. JADE - A FIPA-compliant agent framework. In *Proceedings of the 4th International Conference on the Practical Applications of Agents and Multi-Agent Systems (PAAM-99)*, pages 97–108, 1999.

[10] R. E. Bellman. *Dynamic Programming.* Princeton University Press, 1957.

[11] D. Billings. The first international RoShamBo programming competition. *International Computer Games Association Journal*, 23(1):42–50, 2000.

[12] E. Bonabeau, M. Dorigo, and G. Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems.* Oxford University Press, Oxford, 1999.

[13] A. H. Bond and L. Gasser. *Readings in Distributed Artificial Intelligence.* Morgan Kaufmann Publishers Inc., San Mateo, CA, USA, 1988.

[14] G. Booch, J. Rumbaugh, and I. Jacobson. *The Unified Modeling Language User Guide*. Addison Wesley, 1999.

[15] C. Boutilier, M. Goldszmidt, and B. Sabata. Sequential auctions for the allocation of resources with complementarities. In D. Thomas, editor, *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99-Vol1)*, pages 527–534. Morgan Kaufmann Publishers, 1999.

[16] J. M. Bradshaw, S. Dutfield, P. Benoit, and J. D. Woolley. KAoS: Toward an industrial-strength open agent architecture. In J. M. Bradshaw, editor, *Software Agents*, pages 375–418. AAAI Press / The MIT Press, 1997.

[17] M. E. Bratman. *Intentions, Plans, and Practical Reason*. Harvard University Press, Cambridge, Massachusetts, USA, 1987.

[18] P. Bretier and D. Sadek. A rational agent as the kernel of a cooperative spoken dialogue system: Implementing a logical theory of interaction. *Lecture Notes in Computer Science*, 1193:189–204, 1997.

[19] R. A. Brooks. How to build complete creatures rather than isolated cognitive simulators. In K. Van Lehn, editor, *Architectures for Intelligence*, pages 225–239. Lawrence Erlbaum Associates, Hillsdale, NJ, 1989.

[20] A. Byde, C. Priest, and N. Jennings. Decision procedures for multiple auctions. In C. Castelfranchi and W. L. Johnson, editors, *Proceedings of the First International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2002)*, pages 613–620. ACM Press, 2002.

[21] C. Castelfranchi. Commitments: from individual intentions to groups and organizations. In V. Lesser, editor, *Proceedings of the First International Conference on Multi–Agent Systems*, pages 41–48. MIT Press, 1995.

[22] B. Chaib-draa and D. Vanderveken. Agent Communication Language: Toward a Semantics Based on Success, Satisfaction, and Recursion. In J. Müller, M. P. Singh, and A. S. Rao, editors, *Proceedings of the 5th International Workshop on Intelligent Agents V : Agent Theories, Architectures, and Languages (ATAL-98)*, volume 1555 of *LNAI*, pages 363–380. Springer, 1999.

[23] D. Chauhan and A. D. Baker. JAFMAS: A multiagent application development system. In K. P. Sycara and M. Wooldridge, editors, *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)*, pages 100–107. ACM Press, 1998.

[24] A. Chavez and P. Maes. Kasbah: An agent marketplace for buying and selling goods. In *First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'96)*, pages 75–90. Practical Application Company, 1996.

[25] B. F. Chellas. *Modal Logic.* Cambridge University Press, Cambridge, 1980.

[26] W. J. Clancey. *Situated Cognition. On human knowledge and computer representations.* Cambridge University Press, Cambridge, 1997.

[27] A. Clark. *Being there.* The MIT Press, Cambridge, Mass., 1997.

[28] P. R. Cohen and A. Cheyer. An open agent architecture. In O. Etzioni, editor, *Software Agents — Papers from the 1994 Spring Symposium (Technical Report SS–94–03)*, pages 1–8. AAAI Press, 1994.

[29] P. R. Cohen and H. J. Levesque. Intention is choice with commitment. *Artificial Intelligence, AI*, 42(2–3):213–261, 1990.

[30] M. Colombetti. A commitment-based approach to agent speech acts and conversations. In *Proc. Workshop on Agent Languages and Communication Policies, 4th International Conference on Autonomous Agents (Agents 2000), Barcelona, Spain*, pages 21–29, 2000.

[31] M. Colombetti. Commitment-based semantics for agent communication languages. In *Proc. 1st International Workshop on the History and Philosophy of Logic, Mathematics and Computation (HPLMC-00) San Sebastian, Spain*, pages 9–11, 2000.

[32] M. Colombetti. A language for artificial agents. *Studies in Communication Sciences*, 1(12):1–32, 2001.

[33] M. Colombetti, N. Fornara, and M. Verdicchio. The role of institutions in multiagent systems. Workshop on Knowledge based and reasoning agents VIII Convegno AI*IA 2002, Siena, Italy, 2002.

[34] M. Colombetti, N. Fornara, and M. Verdicchio. Speech acts in artificial agent communication. In *Proceedings of International Conference on Linguistics & the New Professions, Pavia, Italy & Lugano, Switzerland*, 2002.

[35] M. Colombetti and M. Verdicchio. An analysis of agent speech acts as institutional actions. In C. Castelfranchi and W. L. Johnson, editors, *Proceedings of the First International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2002)*, pages 1157–1164. ACM Press, 2002.

[36] R. S. Cost, T. Finin, Y. Labrou, X. Luan, Y. Peng, I. Soboroff, J. Mayfield, and A. Boughanam. Jackal: A Java-based Tool for Agent Development. In *AAAI-98, Workshop on Tools for Agent Development*, Madison, WI, 1998.

[37] Cyc. Upper Cyc Ontology. http://www.cyc.com/cyc-2-1/cover.html, 2002.

[38] K. Dautenhahn. Getting to know each other - artificial social intelligence for autonomous robots. *Robotics and Autonomous Systems*, 16:333–356, 1995.

[39] R. Davis and R. Smith. Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence*, 20(1):63–109, 1983.

[40] K. Decker, K. Sycara, and M. Williamson. Middle-agents for the internet. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, Nagoya, Japan, 1997.

[41] C. Dellarocas. Contractual agent societies: Negotiated shared context and social control in open multi-agent systems. In *Proceedings of Workshop on Norms and Institutions in Multi-Agent Systems (Autonomous Agents'2000), Barcelona, Spain*, 2000.

[42] C. Dellarocas and M. Klein. Civil agent societies: Tools for inventing open agent-mediated electronic marketplaces. In *Proceedings of the Workshop in Agent-Mediated Electronic Commerce (IJCAI'99), Stockholm, Sweden*, 1999.

[43] D. C. Dennett. Intentional systems. *The Journal of Philosophy*, 68:87–106, 1971.

[44] D. C. Dennett. *The intentional stance.* The MIT Press, Massachusetts, 1987.

[45] E. W. Dijkstra. *A Discipline of Programming.* Prentice-Hall International, 1976.

[46] M. d'Inverno, D. Kinny, and M. Luck. Interaction protocols in agentis. In *Proceedings of the Third International Conference on Multi-Agent Systems*, pages 112–119. IEEE Press, 1998.

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

[48] E. H. Durfee. *Coordination of distributed problem solvers.* The Kluwer international series in engineering and computer science ; Artificial intelligence. Kluwer, Boston, 1988.

[49] E. H. Durfee. Distributed problem solving and planning. In G. Weiss, editor, *Multi-agent Systems: A Modern Approach to Distributed Artificial Intelligence*, chapter 3, pages 121–164. The MIT Press, Cambridge, MA, USA, 1999.

[50] E. H. Durfee and J. Rosenschein. Distributed problem solving and multiagent systems: Comparisons and examples. In M. Klein, editor, *Proceedings of the 13th International Workshop on DAI*, pages 94–104, Lake Quinalt, WA, 1994.

[51] eBay. URL: http://www.ebay.com/.

[52] A. Eisenberg. In online auctions of the future, it will be bot vs. bot vs. bot. *New York Times*, 17 August 2000, p. D8.

[53] E. A. Emerson. Temporal and Modal Logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics*, chapter 16, pages 995–1072. Elsevier Science Publishers, Amsterdam, The Netherlands, 1990.

[54] M. Esteva, J. A. Rodríguez-Aguilar, C. Sierra, P. Garcia, and J. L. Arcos. On the formal specification of electronic institutions. In F. Dignum and C. Sierra, editors, *Agent Mediated Electronic Commerce, The European AgentLink Perspective (LNAI 1991)*, pages 126–147. Springer, 2001.

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

[56] J. Ferber and O. Gutknecht. A meta-model for the analysis and design of organizations in multi-agent systems. In *Proceedings of the International Conference on Multi Agent Systems ICMAS'98*. IEEE Press, 1998.

[57] T. Finin, Y. Labrou, and J. Mayfield. KQML as an agent communication language. In J. M. Bradshaw, editor, *Software Agents*, chapter 14, pages 291–316. AAAI Press / The MIT Press, 1997.

[58] Fishmarket FM. URL: http://www.iiia.csic.es/projects/fishmarket/.

[59] Foundation for Intelligent Physical Agents. FIPA 97 specification part 2: Agent communication language, 1997.

[60] Foundation for Intelligent Physical Agents. Agent communication language. FIPA 2000 Specification. http://www.fipa.org, 2000.

[61] N. Fornara and M. Colombetti. Operational specification of a commitment-based agent communication language. In C. Castelfranchi and W. L. Johnson, editors, *Proceedings of the First International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2002)*, pages 535–542. ACM Press, 2002.

[62] N. Fornara and M. Colombetti. Defining interaction protocols using a commitment-based agent communication language. In *Proceedings of the Second International Joint Conference on Autonomous Agents and MultiAgent Systems AAMAS 2003, Melbourne, Australia*, 2003.

[63] N. Fornara and L. M. Gambardella. An autonomous bidding agent for simultaneous auctions. In *Proceedings of the Fifth International Workshop on Cooperative Information Agents (CIA-2001) LNAI 2182*, pages 130–141. Springer-Verlag, 2001.

[64] D. Friedman and J. Rust editors. *The Double Auction Market: Institutions, Theories, and Evidence*. Addison-Wesley, Redwood City, CA, 1993.

[65] L. M. Gambardella, A. E. Rizzoli, N. Fornara, A. Alessandrini, and C. Epifani. The use of simulation in the socio-economical evaluation of the management of an intermodal terminal. In *Proceedings of the International Workshop on Harbour, Maritime & Multimodal Logistics Modelling and Simulation, Portofino - Italy*, pages 71–76. Society for Computer Simulation, 2000.

[66] L. Gasser. Social conceptions of knowledge and action: DAI foundations and open systems semantics. *Artificial Intelligence*, 47(1–3):107–138, 1991.

[67] L. Gasser, C. Braganza, and N. Herman. MACE: A Flexible Testbed for Distributed AI Research. In M. N. Huhns, editor, *Distributed Artificial Intelligence*, pages 119–152. Pitman Publishing, 1987.

[68] L. Gasser and eds. M. N. Huhns. *Distributed Artificial Intelligence, Volume II.* Research Notes in Artificial Intelligence. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA, 1989.

[69] M. R. Genesereth and R. E. Fikes. Knowledge Interchange Format, Version 3.0 Reference Manual. Technical Report Logic-92-1, Computer Science Department, Stanford University, Stanford, CA, USA, June 1992.

[70] M. R. Genesereth, A. M. Keller, and O. M. Duschka. Infomaster: an information integration system. In Joan M. Peckman, editor, *Proceedings, ACM SIGMOD International Conference on Management of Data: SIGMOD, Tucson, Arizona, USA*, volume 26(2) of *SIGMOD Record (ACM Special Interest Group on Management of Data)*, pages 539–542, New York, 1997. ACM Press.

[71] M. R. Genesereth and S. P. Ketchpel. Software agents. *Communications of the Association for Computing Machinery*, 37(7):48–53, 1994.

[72] M. Georgeff and A. Rao. A profile of the Australian AI Institute. *IEEE Expert*, 11(6):89–92, 1996.

[73] M. P. Georgeff and F. F. Ingrand. Decision-making in an embedded reasoning system. In N. S. Sridharan, editor, *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pages 972–978. Morgan Kaufmann, 1989.

[74] M. P. Georgeff and A. L. Lansky. Procedural knowledge. In *Proceedings of the IEEE Special Issue on Knowledge Representation*, volume 74, pages 1383–1398, 1986.

[75] E. M. Gerson. On 'quality of life'. *American Sociological Review*, 41:793–806, 1976.

[76] A. Greenwald and J. Boyan. Bid determination in simultaneous auctions a case study. In *Proceedings of the Third ACM Conference on Electronic Commerce*, pages 115–124, 2001.

[77] A. Greenwald and P. Stone. Autonomous bidding agents in the trading agent competition. *IEEE Internet Computing*, 5(2):52–60, March/April 2001.

[78] H. P. Grice. Meaning. *Philosophical Review*, 66:377–388, 1957. Republished in H.P. Grice, Studies in the Way of Words, Harvard University Press, 1989.

[79] Standard Upper Ontology (SUO) Working Group. http://suo.ieee.org/, 2002.

[80] The DARPA Knowledge Sharing Initiative External Interfaces Working Group. Specification of the KQML agent-communication language (tech report), 1993.

[81] F. Guerin and J. Pitt. A semantic framework for specifying agent communication languages. In *Proceedings, Fourth International Conference on Multiagent Systems (ICMAS2000)*, pages 395–396, 2000.

[82] A. Haddadi. *Communication and cooperation in agent systems: a pragmatic theory*, volume 1056 of *Lecture Notes in Artificial Intelligence and Lecture Notes in Computer Science*. Springer Verlag, New York, USA, 1996.

[83] M. N. Huhns. *Distributed Artificial Intelligence*. Pitman/Morgan Kaufmann, San Mateo, CA, 1987.

[84] C. Iglesias, M. Garrijo, and J. Gonzalez. A survey of agent-oriented methodologies. In J. Müller, M. P. Singh, and A. S. Rao, editors, *Proceedings of the 5th International Workshop on Intelligent Agents V : Agent Theories, Architectures, and Languages (ATAL-98)*, volume 1555 of *LNAI*, pages 317–330, Berlin, 1999. Springer.

[85] Artificial Intelligence Research Institute IIIA. URL: http://www.iiia.csic.es/.

[86] F. F. Ingrand, M. P. Georgeff, and A. S. Rao. An architecture for real-time reasoning and system control. *IEEE Expert*, 7(6):34–44, 1992.

[87] INSPIRE and INSS. URL: http://interneg.carleton.ca/interneg/tools/.

[88] N. Jennings. Commitments and conventions: The foundation of coordination in multi-agent systems. *The Knowledge Engineering Review*, 8(3):223–250, 1993.

[89] N. Jennings. Coordination techniques for distributed artificial intelligence. In G. M. P. O'Hare and N. Jennings, editors, *Foundations of Distributed Artificial Intelligence*, pages 187–210. John Wiley & Sons, 1996.

[90] N. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge. Automated negotiation: prospects, methods and challenges. *Int. J. of Group Decision and Negotiation*, 10(2):199–215, 2001.

[91] N. Jennings, S. Parsons, P. Noriega, and C. Sierra. On argumentation-based negotiation. In *Proceedings of the International Workshop on Multi-Agent Systems*, Boston, USA, 1998.

[92] N. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Journal of Autonomous Agents and Multi-Agent Systems*, 1(1):7–38, 1998.

[93] H. Jeon, C. Petrie, and M. R. Cutkosky. JATLite: A Java Agent Infrastructure with Message Routing. *IEEE Internet Computing*, 4(2):87–96, March/April 2000.

[94] A. J. I. Jones. Conventional signalling acts. Unpublished manuscript.

[95] A. J. I. Jones and M. Sergot. A formal characterisation of institutionalised power. *Journal of the Interest Group in Pure and Applied Logic*, 4(3):427–443, 1996.

[96] E. Kendall. Role modelling for agent system analysis, design and implementation. In *Proceedings of the First International Symposium on Agent Systems and Applications (ASA'99)/Third International Symposium on Mobile Agents (MA'99)*, Palm Springs, CA, USA, 1999.

[97] D. Kinny and M. Georgeff. Modelling and design of multi-agent systems. In J. P. Müller, M. Wooldridge, and N. Jennings, editors, *Proceedings of the ECAI'96 Workshop on Agent Theories, Architectures, and Languages: Intelligent Agents III*, volume 1193 of *LNAI*, pages 1–20, Berlin, 1997. Springer.

[98] S. Kraus, K. Sycara, and A. Evenchik. Reaching agreements through argumentation: A logical model and implementation. *Artificial Intelligence*, 104(1–2):1–69, 1998.

[99] Y. Labrou and T. Finin. A semantics approach for KQML, a general purpose communication language for software agents. In *Proceedings of the Third International Conference on Knowledge and Information Management (CIKM-94)*, Silver Spring, 1994.

[100] Y. Labrou, T. Finin, and Y. Peng. Agent communication languages: The current landscape. *IEEE Intelligent Systems*, 14(2):45–52, 1999.

[101] V. R. Lesser. 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, 1998.

[102] J. Lind and K. Fischer. Transportation scheduling and simulation in a railroad scenario: A multi-agent approach. Technical Memo TM-98-05, Deutsches Forschungszentrum für Künstliche Intelligenz, 1998.

[103] M. Ljungberg and A. Lucas. The OASIS air-traffic management system. In *Proceedings of the Second Pacific Rim International Conference on Artificial Intelligence (PRICAI '92)*, Seoul, Korea, 1992.

[104] A. R. Lomuscio, M. Wooldridge, and N. Jennings. A classification scheme for negotiation in electronic commerce. *Lecture Notes in Computer Science*, 1991:19–33, 2001.

[105] G. Mantovani. *New Communication Environments From Everyday to Virtual*. Taylor & Francis, London, 1996.

[106] G. Mantovani. *Exploring Borders Understanding Culture and Psychology*. Routledge, London, 2000.

[107] J. Mayfield, Y. Labrou, and T. Finin. Desiderata for agent communication languages. In *AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, 1995.

[108] H. Mazouzi, A. El Fallah, and S. Haddad. Open protocol design for complex interactions in multi-agent systems. In C. Castelfranchi and W. L. Johnson, editors, *Proceedings of the First International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2002)*, pages 517–526. ACM Press, 2002.

[109] J. McCarthy. Ascribing mental qualities to machines. In M. Ringle, editor, *Philosophical Perspectives in Artificial Intelligence*, pages 161–195. Humanities Press, Atlantic Highlands, New Jersey, 1979.

[110] MSNauctions. URL: http://auctions.msn.com/.

[111] N. Muscettola, P. Pandurang Nayak, B. Pell, and B. C. Williams. Remote agent: to go boldly where no AI system has gone before. *Artificial Intelligence*, 103(1–2):5–47, 1998.

[112] R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, and W. R. Swarton. Enabling technology for knowledge sharing. *AI Magazine*, pages 36–56, March 1991.

[113] H. P. Nii. Blackboard systems. In A. Barr, P. R. Cohen, and E. A. Feigenbaum, editors, *The Handbook of Artificial Intelligence*, volume 4, pages 1–82. Addison-Wesley, Reading, MA, 1989.

[114] M. H. Nodine and A. Unruh. Facilitating open communication in agent systems: The InfoSleuth infrastructure. In M. P. Singh, A. Rao, and M. Wooldridge, editors, *Proceedings of the 4th International Workshop on Agent Theories, Architectures, and Languages (ATAL-97)*, volume 1365 of *LNAI*, pages 281–296, Berlin, 1998. Springer.

[115] P. Noriega and C. Sierra, editors. *Proceedings of the 1st International Workshop on Agent Mediated Electronic Commerce (AMET-98)*, volume 1571 of *LNAI*, Berlin, 1999. Springer.

[116] G. M. P. O'Hare and N. Jennings. *Foundations of Distributed Artificial Intelligence*. Wiley-Interscience: New York, 1996.

[117] R. S. Patil, R. E. Fikes, P. F. Patel-Schneider, D. McKay, T. Finin, T. Gruber, and R. Neches. The DARPA knowledge sharing effort: progress report. In C. Rich, B. Nebel, and W. R. Swartout, editors, *Principles of knowledge representation and reasoning: proceedings of the third international conference*. Morgan Kaufmann, 1992.

[118] FIPA Semantics Technical Committee: J. Pitt and F. G. McCabe. Semantic framework work plan, 2001.

[119] FIPA Semantics Technical Committee: J. Pitt and F. G. McCabe. FIPA TC Semantics Use Cases, 2002.

[120] J. Pitt and A. Mamdani. A protocol-based semantics for an agent communication language. In D. Thomas, editor, *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99-Vol1)*, pages 486–491, S.F., 1999. Morgan Kaufmann Publishers.

[121] A. N. Prior. *Time and modality.* Oxford University Press, Oxford, UK, 1957.

[122] A. N. Prior. *Past, Present and Future.* Oxford University Press, Oxford, 1967.

[123] M. L. Puterman. *Markov Decision Processes—Discrete Stochastic Dynamic Programming.* John Wiley & Sons, Inc., New York, NY, 1994.

[124] A. S. Rao and M. P. Georgeff. Modeling rational agents within a BDI-architecture. In J. F. Allen, R. Fikes, and E. Sandewall, editors, *KR'91: Principles of Knowledge Representation and Reasoning*, pages 473–484. Morgan Kaufmann, San Mateo, California, 1991.

[125] N. Rescher and A. Urquhart. *Temporal Logic.* Springer-Verlag, Berlin, 1971.

[126] D. Riehle and T. Gross. Role model based framework design and integration. In *Proceedings of the 13th Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA-98)*, volume 33, 10 of *ACM SIGPLAN Notices*, pages 117–133. ACM Press, 1998.

[127] A. E. Rizzoli, N. Fornara, and L. M. Gambardella. A simulation tool for combined rail/road transport in intermodal terminals. *Mathematics and Computers in Simulation*, 59(1-3):57–71, 2002.

[128] J. A. Rodríguez-Aguilar, F. J. Martín, P. Noriega, P. Garcia, and C. Sierra. Competitive scenarios for heterogeneous trading agents. In K. P. Sycara and M. Wooldridge, editors, *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)*, pages 293–300. ACM Press, 1998.

[129] J. S. Rosenschein and G. Zlotkin. *Rules of Encounter.* MIT Press, Cambridge, Massachusetts, 1994.

[130] S. J. Russell. Rationality and intelligence. In C. Mellish, editor, *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 950–957. Morgan Kaufmann, 1995.

[131] S. J. Russell and P. Norvig. *Artificial Intelligence. A Modern Approach.* Prentice-Hall, Englewood Cliffs, New Jersey, 1995.

[132] J. Rust, J. Miller, and R. Palmer. Behavior of trading automata in a computerized double auction market. In D. Friedman and J. Rust, editors, *The Double Auction Market: Institutions, Theories, and Evidence*. Addison-Wesley, Redwood City, CA, 1992.

[133] T. Sandholm. An implementation of the contract net protocol based on marginal cost calculations. In *Proceedings of the 11th National Conference on Artificial Intelligence*, pages 256–263. AAAI Press, 1993.

[134] T. Sandholm. Limitations of the Vickrey auction in computational multiagent systems. In V. Lesser, editor, *Proceedings of the First International Conference on Multi–Agent Systems*. MIT Press, 1995.

[135] T. Sandholm. Distributed rational decision making. In G. Weiss, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, chapter 5, pages 201–258. The MIT Press, Cambridge, Massachusetts, 1999.

[136] T. Sandholm and V. Lesser. Issues in automated negotiation and electronic commerce: Extending the contract net framework. In V. Lesser, editor, *Proceedings of the First International Conference on Multi–Agent Systems*, pages 328–335. MIT Press, 1995.

[137] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. *Computer*, 29(2):38–47, 1996.

[138] W. R. Scott. *Organizations: Rational, Natural, and Open Systems*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1987.

[139] J. R. Searle. *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press, Cambridge, United Kingdom, 1969.

[140] J. R. Searle. A taxonomy of illocutionary acts. In K. Gunderson, editor, *Language, Mind, and Knowledge. Minnesota Studies in the Philosophy of Science, Vol. 7*, pages 344–369. University of Minnesota Press, Minneapolis, Minnesota, 1975.

[141] J. R. Searle. *Intentionality: An Essay in the Philosophy of Mind*. Cambridge University Press: Cambridge, England, 1983.

[142] J. R. Searle. Conversation. In H. Parret and J. Verschueren, editors, *(On) Searle on Conversation*, pages 7–29. J. Benjamins Publishing Company, Amsterdam, 1991.

[143] J. R. Searle. *The construction of social reality*. Free Press, New York, 1995.

[144] J. R. Searle. *Mind, Language, and Society*. Basic Books, New York, 1998.

[145] O. Shehory. Optimality and risk in purchase from multiple auctions. In M. Klush and F. Zambonelli, editors, *Cooperative Information Agents V, Proceedings of the 5th International Workshop CIA 2001*, number 2182 in Lecture Notes in Artifical Intelligence, pages 142–153. Springer-Verlag: Heidelberg, Germany, 2001.

[146] Y. Shoham. Agent-oriented programming. *Artificial Intelligence*, 60:51–92, 1993.

[147] Y. Shoham and M. Tennenholtz. On social laws for artificial agent societies: off-line design. *Artificial Intelligence*, 72(1-2):231–252, 1994.

[148] J. S. Sichman, R. Conte, C. Castelfranchi, and Y. Demazeau. A social reasoning mechanism based on dependence networks. In A. G. Cohn, editor, *Proceedings of the Eleventh European Conference on Artificial Intelligence*, pages 188–192. John Wiley and Sons, 1994.

[149] C. Sierra and F. Dignum. Agent-mediated electronic commerce: Scientific and technological roadmap. In F. Dignum and C. Sierra, editors, *Agent Mediated Electronic Commerce, The European AgentLink Perspective (LNAI 1991)*, pages 1–18. Springer, 2001.

[150] H. A. Simon. Rational choice and the structure of the environment. In *Models of Bounded Rationality*, volume 2. MIT Press, Cambridge, Massachusetts, 1958.

[151] M. P. Singh. Agent communication languages: Rethinking the principles. *IEEE Computer*, 31(12):40–47, December 1998.

[152] M. P. Singh. A social semantics for agent communication languages. In F. Dignum and M. Greaves, editors, *Issues in Agent Communication*, pages 31–45. Springer-Verlag: Heidelberg, Germany, 2000.

[153] R. G. Smith. The contract net protocol: High-level communicaation and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113, 1980.

[154] R. G. Smith. *A Framework for Distributed Problem Solving.* UMI Research Press, 1980.

[155] P. Stone. Multiagent Competitions and Research: Lessons from RoboCup and TAC. In *Proceedings of RoboCup 2002 International Symposium*, 2002.

[156] P. Stone, T. Balch, and G. Kraetzschmar (eds.). *RoboCup 2000: Robot Soccer World Cup IV.* Lecture Notes in Computer Science. VOL. 2019. Springer Verlag, 2001.

[157] P. Stone and A. Greenwald et al. The first international trading agent competition: Autonomous bidding agents. *Electronic Commerce Research Journal*, (to appear), 2002.

[158] P. Stone, M. L. Littman, S. Singh, and M. Kearns. ATTac-2000: An adaptive autonomous bidding agent. *Journal of AI Research (JAIR)*, 15, 2001.

[159] P. Stone and M. Veloso. (Formerly citable as Carnegie Mellon University CS technical report number CMU-CS-97-193. december, 1997) Multiagent Systems: A Survey from a Machine Learning Perspective. *Autonomous Robots*, 8(3):345–383, 2000.

[160] A. Strauss. *Negotiations: Varieties, Processes, Contexts, and Social Order*. Jossey Bass, San Francisco, 1978.

[161] K. P. Sycara. Multiagent systems. *The AI Magazine*, 19(2):79–92, 1998.

[162] Trading Agent Competition TAC. URL: http://auction2.eecs.umich.edu/.

[163] D. R. Traum and J. F. Allen. Discourse obligations in dialogue processing. In J. Pustejovsky, editor, *Proceedings of the Thirty-Second Meeting of the Association for Computational Linguistics*, pages 1–8. Association for Computational Linguistics, Morgan Kaufmann, 1994.

[164] uBid. URL: http://www.ubid.com/.

[165] M. Venkatraman and M. Singh. Verifying Compliance with Commitment Protocols. *Autonomous Agents and Multi-Agent Systems*, 2(3):217–236, 1999. Special Issue on Coordination Mechanisms and Patterns for Web Agents.

[166] M. Verdicchio and M. Colombetti. Commitments for agent-based supply chain management. *ACM SIGecom Exchanges*, 3(1):13–23, 2002.

[167] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.

[168] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behaviour*. Princeton University Press, 1944.

[169] G. H. von Wright. Deontic logic. *Mind*, 60(237):1–15, 1951.

[170] G. H. von Wright. A note on deontic logic and derived obligation. *Mind*, 65:507–509, 1956.

[171] C. J. Watkins. *Models of Delayed Reinforcement Learning*. PhD thesis, Psychology Department, Cambridge University, Cambridge, United Kingdom, 1989.

[172] G. Weiss. *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*. MIT Press, Cambridge, MA, USA, 1999.

[173] G. Weiss and S. Sen editors. *Adaptation and Learning in Multiagent Systems*. Springer, 1996.

[174] M. P. Wellman. A computational market model for distributed configuration design. In *Proceedings of the 12th National Conference on Artificial Intelligence. Volume 1*, pages 401–407. AAAI Press, 1994.

[175] M. P. Wellman, P. Wurman, K. O'Malley, R. Bangera, S. Lin, D. Reeves, and W. E. Walsh. Designing the market game for a trading agent competition. *IEEE Internet Computing*, 5(2):43–51, March/April 2001.

[176] M. Wooldridge. Verifiable Semantics for Agent Communication Languages. In Y. Demazeau, editor, *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS'98)*, pages 349–356, Paris, France, July 1998. IEEE Press.

[177] M. Wooldridge. Intelligent agents. In G. Weiss, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, chapter 1, pages 27–78. The MIT Press, Cambridge, MA, USA, 1999.

[178] M. Wooldridge. *Reasoning about Rational Agents.* Intelligent Robots and Autonomous Agents. The MIT Press, Cambridge, Massachusetts, 2000.

[179] M. Wooldridge. *An Introduction to Multiagent Systems.* John Wiley and Sons, 2002.

[180] M. Wooldridge and N. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.

[181] M. Wooldridge and N. Jennings. The cooperative problem-solving process. *JLC: Journal of Logic and Computation*, 9, 1999.

[182] M. Wooldridge, N. Jennings, and D. Kinny. A methodology for agent-oriented analysis and design. In Oren Etzioni, Jörg P. Müller, and Jeffrey M. Bradshaw, editors, *Proceedings of the Third International Conference on Autonomous Agents (Agents'99)*, pages 69–76, Seattle, WA, USA, 1999. ACM Press.

[183] M. Wooldridge, N. Jennings, and D. Kinny. The gaia methodology for agent-oriented analysis and design. *Autonomous Agents and Multi-Agent Systems*, 3(3):285–312, September 2000.

[184] P. Wurman. Toward flexible trading agents. In *AAAI Spring Symposium on Game Theoretic and Decision Theoretic Agents*, pages 134–140, March 2001.

[185] P. Wurman, W. Walsh, and M. Wellman. Flexible double auctions for electronic commerce: Theory and implementation. *Decision Support Systems*, 24:17–27, 1998.

[186] P. Wurman, M. P. Wellman, and W. E. Walsh. The Michigan Internet AuctionBot: A configurable auction server for human and software agents. In K. P. Sycara and M. Wooldridge, editors, *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)*, pages 301–308. ACM Press, May 9–13, 1998.

[187] Yahoo!auctions. URL: http://auctions.yahoo.com/.

[188] P. Yolum and M. P. Sing. Flexible protocol specification and execution: Applying event calculus planning using commitments. In C. Castelfranchi and W. L. Johnson, editors, *Proceedings of the First International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2002)*, pages 527–534. ACM Press, 2002.

[189] P. Yolum and M. P. Singh. Synthesizing finite state machines for communication protocols. Technical Report TR-2001-06, Department of Computer Science, North Carolina State University, June 27 2001.

[190] F. Zambonelli, N. Jennings, and M. Wooldridge. Organizational abstractions in the analysis and design of multi-agent systems. In *Proceedings 1st International Workshop on Agent-Oriented Software Engineering at ICSE 2000, Limerick (IR)*, 2000.