Motivational Attitudes and Norms in a unified Agent Communication Language for open Multi-Agent Systems: A Pragmatic Approach

Rodrigo Agerri

A thesis submitted to City University for the degree of Doctor of Philosophy

Department of Computing
City University
London
September 2006

Contents

List of Tables

List of Figures

Acknowledgements

I wish to acknowledge that this work has been partially funded by the School of Informatics of City University. I am grateful to my supervisor for supervising this work. Thanks to my City colleages (Alex, Jacek, Panos, Panayiota, Penny, Tshiamo, Gilberto, Waraporn and the rest) for the early discussions and good atmosphere. I would like to thank Shant for his unvaluable help in my time at City, and for some great time out of it. I wish particularly to thank Eduardo and Esther for the best Mondays I ever had and for their overall support and friendship. The rest of my friends both in UK and the Basque Country were very kind for not asking how the thesis was going. I really appreciated that.

Thanks to my family and Arantza for everything.

Declaration

Abstract

In order to perform some tasks, agents need to interact with each other. Thus, a Multi-Agent System (MAS) is a system composed by several agents, capable of mutual interaction. Communication is a kind of interaction that allows agents to work more effectively by sharing knowledge and exchanging information. Thus, communication allow agents to make queries, transmit information, perform declarations and to commit themselves to execute an action. For agents to communicate, a method of sequencing messages is needed (conversations). For conversations to be successful, pragmatic principles to guide the linguistic interchange should be make available. These principles should not violate crucial properties of agency such as autonomy, heterogeneity and proactiveness. Various classes of agent communication languages (ACLs) have been proposed to handle these issues, but standardization is still a holy grail. We claim that a rethinking of the general principles on the foundations of ACLs is needed. More specifically, a redistribution of the role played by the semantics (speech acts) and pragmatics (protocols and policies) of ACLs will dissolve some of the most important problems currently affecting agent communication. Agent communication has traditionally focused on the semantics of speech acts, and many important advances have been done on that respect. But for some exceptions discussed later on the pragmatic component has often been the poor relative, consisting usually on low-level contextual free protocols that merely established the order in which speech acts may be used. This thesis aims to show how a high-level ACL pragmatics is crucial to facilitate the use of the semantic component in a variety of scenarios and a necessary step towards standardization. In the pragmatic turn for agent communication that we are proposing, ACL pragmatics will take the form of conversation norms. These principles can be specifically formulated by means of conversation protocols and policies that govern agents' message interchange taking into account contextual factors that affect agents' decisions. Once the theoretical issues are established, we ground the pragmatic principles in a computational model and study its applicability using a declarative programming language.

1

Introduction: Interacting Agents

Since the early 1990s, the development of agent-based software has become one of the most exciting areas in computing and information technology. The adoption of agent technology has increasingly become popular to develop software programs for electronic commerce activities such as banking and travel agencies. The agent paradigm aims to build societies of agents accessing the Internet on behalf of their owners, to provide services, to retrieve information and to perform various other tasks. Agents may need to communicate to be able to cooperate, negotiate, compete, and generally, to achieve their goals. Thus, agents require a common language to understand each other. Projects like Knowledge Query Manipulation Language (KQML) and Foundation for Intelligent and Physical Agents (FIPA) have developed complex high-level ACLs to provide such a common language.

1.1 Multi-Agent Systems and AI

Agents can be characterized as computational entities that are *intelligent*, *autonomous* and that can be *delegated* to perform certain tasks (Huhns and Singh, 1997). An agent being a computational entity means that it exists as a program that runs on computing devices. Being *autonomous* means that agents control their own behaviour and can act without intervention of an external agent, software or human. *Intelligent* refers to how agents reason about how to achieve their goals and perform different tasks.

Nowadays, it is argued that the main characteristic of agents is their capacity to interact. Thus, AgentLink's Roadmap (AgentLink, 2005), states that agents are computer systems capable of flexible autonomous action in dynamic, unpredictable domains, such as multi-agent domains. From this point of view, the social aspect of agents' behaviour becomes the most important characteristic of the agent paradigm in computing.

With the growth of the Internet, the agent-based paradigm thinks of computing as a social activity. Applications are not longer monolithic, or distributed applications managed by a single organization, but they conform societies of components. Agents are intelligent entities capable of interacting with other agents, and the society in which they operate is the multi-agent system. Summarizing, agents can be described as computational entities that: interact (through communication), organize in societies and that are intelligent and autonomous.

Agents and multi-agent systems are being developed within the Artificial Intelligence community and, in particular, within the field of Distributed Artificial Intelligence (DAI) which, according to Nwana (1996) consists of multi-agent systems, Distributed Problem Solving (DPS) and Parallel AI. In DPS and Parallel AI, cooperation is a requisite for the distributed entities to solve problems whereas in multi-agent systems its members may cooperate or compete (Ferber, 1999).

Agents are different from other software programs regarding the level of abstraction. Agents differ from Distributed Computing and Object Oriented programming because, among other reasons, they can be delegated tasks and will execute them autonomously (Wooldridge, 2002). Usually agents do not have all the knowledge or skills to execute their tasks, so they require cooperation from other agents. Since agents are high level objects, it is reasonable to expect them to communicate at a high level, using a language with enough complexity to express propositional attitudes.

Both ACLs and multi-agent systems can be seen as artificial counterparts of natural language and human societies. From this point of view, the development of a system would consist of the design of an artificial society of agents. Social structures will be defined, since agents may play certain roles in the society. Goal- and task-oriented coordination is a key feature of interaction between agents. Competition and cooperation are two important patterns of coordination. On the one hand, in a competitive situation, agents' goals are conflicting and, therefore, they work against each other. On the other hand, agents are cooperative when they work together to achieve their goals.

The initial development of standard languages (Finin and Weber, 1993) for agent communication was strongly influenced by the work done on planning and speech acts (Cohen and Perrault, 1979). As the interest on ACLs was growing, continuing research (Singh, 2000; Chaib-Draa and Dignum, 2002; Wooldridge, 2000b; Mayfield et al., 1996) was shaping the general view with respect to agent communication. We echo on the research done by those authors to compile a basic set of issues that need to be addressed when specifying standards for communication between software agents in open environments.

The first generally assumed ACL requirement is that the specifications of ACLs do not violate agents' **autonomy**. Moreover, for agents to use the ACL in a variety of scenarios and environments, we need to specify a **complete** range of speech acts in their semantics. For example, the majority of the ACLs proposed do not offer speech acts to perform commissive and declarative communicative actions. In relation to this, **contextual** factors have to be dealt with. So far, context in ACLs has been fixed with the sender. Hence, a high-level ACL pragmatics should be defined to complement the semantics in dealing with how to use the ACL in context. Given that we are interested in ACLs being high-level languages, their specifications need to be **declarative**. An specification defines the meaning of the messages or protocols, not the merely order in which they can be used. Providing the specifications using a precise **formal** language facilitates its application for heterogeneous agents. In particular, logic-based specifications offer many advantages to reason about agents' behaviour, communicative or otherwise. Being formal is not enough though. If we want to be able to verify whether agents conform to the specification, the ACL needs to be **grounded** in a computational model. This will allow to translate the properties of the agents of the system into program properties, which in turn is the first step towards pre-runtime verification. To achieve other types of verification, e.g post-runtime, ACL must be **public**. In other words, it should not depend on agents' internal states. Finally, some authors have pointed out that agent communication is not effective if we cannot find a method to facilitate that the receiver reacts according to the sender's interests and not simply ignore the incoming messages. In the terminology of speech acts theory, the **perlocutionary** effects of sending a message have to be dealt with.

Ideally, these properties should constitute the starting point in the development of an agent communication language. Unfortunately, none of the several approaches proposed up to date comply with the entire set of properties, so not even the starting point towards standardization in agent communication can be taken for granted. We claim that many of the problems can be resolved by simply reorganizing the semantic and pragmatic components of ACLs. In fact, ACL pragmatics have remained underdeveloped; it seems to be widely understood that the meaning of a message in agent communication can be captured by merely specifying its semantics. We strongly claim that is not the case. Agents perform communicative actions in specific scenarios, playing specific roles, and with different strategic objectives (cooperation, competition, etc.). All these contextual factors are part of what an agent will understand when receiving a message and they also will affect its subsequent response. Note that these issues are left out in the traditional view of ACL pragmatics in which we simply need to specify an order in which messages are performed. On the contrary, we say that the meaning of a message consists of both of its semantics and pragmatics, and it is the main task of this thesis to precisely re-define the meaning of ACL semantics and pragmatics in a way that their particular spefications will comply with the properties discussed above.

1.2 Open Normative Multi-Agent Systems

Open multi-agent systems are understood as open, dynamic and failure prone contexts where the agents that constitute the system can come from anywhere and may be buggy or even malicious. The specification of social principles or laws to regulate agents' behaviour has encountered a number of shortcomings:

- Many emerging projects (electronic commerce, large product development projects, multi-national rescue operations) require the ability to rapidly assemble virtual organizations on the Internet with partners who may have never worked together before. Agent systems represent a highly promising approach for addressing such challenges, but work to date has focused mainly on closed systems consisting of well-behaved and sincere agents developed under centralized control and running on reliable infrastructures.
- It is also possible that the constituent agents may represent individuals or organizations with conflicting interests. In these kind of scenarios, the specification of agents are not public and concepts like deception become crucial since agents could benefit by lying in a negotiation process, or discriminate other agents against a competitor, etc. To coordinate the actions of these agents to execute delegated tasks, some interaction protocols, *mechanisms for interactions*, based on game theory, have been proposed. According to this non-normative approach to coordination, agents decisions are based on maximizing their individual utilities.
- The problem with this line of research is that there is no concept of social action as agents' joint actions. On the contrary, agents calculate individually

their best choice and will not trust each other in any negotiation. Therefore, coordinated cooperation is not achievable (Alonso, 2004). To solve this, the general trend within the agents community has been to constrain the coordination rules. Off-line social laws which agents must follow automatically have been proposed, strongly constraining the *autonomy* of agents.

Normative systems constitute an alternative to solve these problems. In this thesis, the society (multi-agent system) formed by the agents is regulated by norms that respect properties of agents such as autonomy and heterogeneity. Social relations between agents may be specified in terms of rights, obligations and roles. We provide norms of communication that help agents to compete and cooperate. Message exchange will modify social relations according to the conventions of the open society. NPRAG partially depends on the general normative concepts that structure the society (the open multi-agent system). Next section discusses how the strategies to build ACLs so far do not result in a suitable ACL to be used in open environments. Their main shortcomings offer a general and precise view of what agent communication languages lack. Moreover, these problems will constitute the objectives that this thesis will attempt to achieve.

1.3 Approaches to Communication

In open multi-agent systems, it is not feasible for a single agent to have complete knowledge of the system. Therefore, it needs to interact to share knowledge and to be able to use others' capabilities. In the case where an agent cannot achieve a goal alone, it must find some other agents to help and a method by which it can expect to get that help. In other words, agents need a standardized language to communicate.

Inspired on some trends in the logic-philosophical tradition (Searle, 1969; Grice, 1989) and the work on BDI agents, most of the early work on agent communication generally assumed that, for agents to communicate with one another, it is necessary to take into account their mental states and those of the other agents (FIPA ACL, 2002; Finin et al., 1997). In other words, agents should be able to express intentions and reason about them. Moreover, they also need to reason about second order beliefs and intentions (reasoning by attribution). However, some authors (Fornara and Colombetti, 2004) have argued that, in order to specify efficient communication protocols, the mental aspect of communication should be abandoned in favour of a social approach (based on the idea of commitment). We argue that the social aspect of communication is part of the pragmatics of the language, stressing in the semantics the illocutionary component of speech acts. However, we are inspired by (Stalnaker, 1989; Fagin et al., 1995), to see mental states not as agents' internal states, but as a sort of common ground between agents with respect to their purposes, beliefs or knowledge about the world. Agents execute actions with the intention to communicate some meaning and to produce a response in the receiver to satisfy that intention. This thesis shall offer a unified framework for agent communication in which the social aspect is introduced as Normative Pragmatics (NPRAG) for agent communication.

Still today, the most extended and traditional way of describing agent communication is by means of cognitive notions. KQML and FIPA ACL both present a mentalistic approach to communication. One of the key problems in agent communication is which point of view to take to specify the language. At least two more paradigms as alternatives to the mentalistic one have been proposed recently: a procedural approach and a social approach.

1.3.1 Mentalistic Approach

The first developments in agent communication used mental (propositional) attitudes to describe the semantics of the language, that is, the semantics is specified in terms of the beliefs and intentions. This approach has its origins in natural language pragmatics (Searle, 1969; Cohen and Perrault, 1979); it was first adopted to agent communication by Finin et al. (1994), Cohen and Levesque (1997) and FIPA ACL (2002). The semantics of FIPA ACL and KQML present a STRIPS (Fikes and Nilsson, 1971) style definition and are based on a multi-modal logic of possible worlds.

Agents are conceived from an intentional point of view. This is coherent with the BDI paradigm based on beliefs, desires an intentions. These attitudes are associated with an agent and a propositional content. Its origins can be found in Denett's work on the intentional stance and its application to the agent paradigm by Rao and Georgeff (1991b).

Dennett (1981) argued that a theory of agency should present the following six properties: rationality, intentionality, stance, reciprocity, communication and consciousness. The first three define a basic intentional system. According to Dennett, 'rationality' refers to reason following the laws of logic; 'intentionality' is defined as the rational intentions of the system based on agent's beliefs about both the world and its own preferences. There are three different types of stances: physical, design, and intentional. The fourth condition, 'reciprocity', is the ability to have *second-order* or complex intentions, that is, intentions that refer to other agents' intentions, desires and beliefs.

The communicative condition is closely related with reciprocity, since an agent should consider that its messages are (or may be) used by the addressee to update its own beliefs (including its second-order beliefs). The last condition, 'consciousness', is the ability of an agent to learn and/or obtain new skills.

According to Singh (2000), mentalistic ACLs assume that agents can reason and infer all the implications of their beliefs, a claim that is not computationally realistic. He also argues that mentalistic semantics are not based on any computational model and that they are not able to account for deceptive behaviour; an agent could be lying and not having the propositional attitude specified in the semantics.

We believe that some of the Singh's criticism is based on the misconception of the role that semantics should play in agent communication. In particular, it is based on the idea of a general purpose semantics that not only specifies the meaning of the linguistic particles (i.e., communicative acts), but that also has to provide interpretation rules to guide agent communication. Usually, ACLs proposed so far lack of a separate but complementary pragmatic theory.

1.3.2 Procedural Approach

This approach introduces the idea of conversation policies that coordinate the use of ACL messages. Both Akkermans et al. (1998) and Greaves et al. (2000) use finite state diagrams to order the message sequencing. This allows the specification of sets of conversation templates. From an implementation point of view, conversation templates are attractive because they are mere *prescriptions* on the communicative behaviour of agents. However, predefined behavioural templates constrain too much the autonomy of agents. Conversation policies work more like fixed protocols than inference rules. Moreover, this approach to communication is not high-level enough, in the sense that the meaning of a message is only operational, that is, its meaning resides in the order in which messages occur. In our view, procedural approaches reduce agent communication to a meaningless exchange of ordered messages, in which agents' beliefs and goals no longer play any role.

1.3.3 Social Approach

Singh (2000) provides a semantics for ACLs in terms of the changes it causes in the social relations existing between the participants in the conversation. Singh argues that mentalistic semantics is not suitable for most open multi-agent applications. Agents' heterogeneity makes it impossible to uniformly determine their beliefs and intentions.

According to this approach, communicative actions are part of an ongoing social interaction. If it is not possible to determine whether agents have a specific mental state, then agents must follow social laws on which conversations are based. Because concepts such as commitment, obligation, power and convention are implicitly included in the behaviour of the agent, the result is that different systems present incompatibilities.

Social-based approaches eliminate the intentional aspect of communication to substitute it with the social notion of commitment. In other words, in those approaches it is not possible to express that an agent has performed a communicative act with the intention of achieving a particular goal, because it will be expressing the commitment of doing something or the commitment that the receiver commits to do something. We believe that an ACL should capture the general idea that agents communicate to achieve goals sometimes with the intention that some particular agent perform some specific action.

1.4 Thesis Objectives

The thesis objectives are motivated by the previous discussion. Agent communication needs to partially rethink their principles if we want to develop a general purpose and efficient high-level communication language for multi-agent systems. We have already discussed a number of properties that are regularly mentioned in the literature and we echo the voices of authors such as Singh (2000); Chaib-Draa and Dignum (2002); Wooldridge (2000b); Mayfield et al. (1996) to propose a number of requirements that an ACL for open multi-agent systems should possess as a starting point. These requirements correspond to the objectives we want to achieve in this thesis. Specifically, the unified ACL presented here must exhibit the following characteristics:

- **Autonomous:** Agent's behaviour must not be too constrained. Sincerity must not be a requirement.

- **Complete**: An ACL should be complete, that is, it must include at least those categories defined in Searle's taxonomy.
- **Contextual**: The context of FIPA ACL is fixed with the sender. This impedes to use the language in different contexts, which affects the heterogeneity of agents.
- **Declarative**: The semantics should state the meaning of the messages, and not the order in which can be used. Guiding the use of ACLs should be done contextually. Thus, it would be possible to adapt the ACL by constraining the use of a subset in a specific context.
- **Formal**: It should offer a formal specification of its semantics and pragmatics. A clear and explicit specification would facilitate interoperability for open multi-agent systems.
- **Grounded**: The ACL presented should be grounded into a computational model. This will allow to translate the properties of the agents of the system into program properties. This also facilitates the verification of the ACL.
- **Public**: Communication must be public. Social aspects from the context should include the status of the participants and the relation of this part of the conversation to other parts of the discourse.
- **Perlocutionary**: ACLs should be structured in two different layers: the set of speech acts and the ACL pragmatics that regulate agents' communicative behaviour. ACL pragmatics should also facilitate the achievement of the perlocutionary effects.

From the normative pragmatics standpoint, communication is still understood as a process in which the sender aims to achieve some goal(s). Ideally, agents need mechanisms that help them choose the speech act that best expresses its communicative intention. In the same way that agents reason about the actions they have to execute to achieve their goals, agents should be able to choose the speech act that is most appropriate to achieve their goals. It is highly undesirable that agents spend resources reasoning about each other goals, beliefs and intentions in order to understand a message, and the normative pragmatics presented in this thesis aims to avoid such reasoning through a set of norms of conversation that facilitate the achievement of their goals. In this sense, we believe that our proposal can be embedded in a general open environment in which social behaviour is ruled by norms.

1.5 Contributions

The topic of research of this thesis can be located at an internal or external level with respect to multi-agent systems. The internal level focuses on the design of the internal architectures of an agent, such as deliberative (Brooks, 1991) and reactive (Rao and Georgeff, 1991b) architectures, among others. The external level focuses on the description and specification of the social structure of the system including theories on communication, negotiation (Sierra et al., 1998), argumentation (Amgoud et al., 2000), electronic institutions (Esteva et al., 2001), and so on (there is also the engineering point of view, which provides tools to build agent-based technology (Ciancarini and Wooldridge, 2001)).

This thesis is mainly focused on the specification of ACLs semantics and pragmatics, that is, on the external level of open and normative MAS. Thus, the description of the internals of the agent programs is beyond the purpose of this thesis. However, to provide a complete picture of our model, we will show how Simple Programming Language (Manna and Pnueli, 1995) can be used to describe agent programs in multi-agent systems. We will also discuss the verification and applicability of our pragmatic approach. As it was stated in the previous section, the mentalistic approach is the most widely extended in agent communication.

The main topics of research on agent communication are communicative acts, interaction protocols and ontologies.

- *Communicative Acts* are the counterparts of speech acts (Searle, 1969) in natural language for artificial agents.
- *Interaction Protocols* are templates that establish and describe the permitted use of the available communicative actions.
- *Ontologies* refer to the lexicon of the content language, that is, ontologies relate the terms to the objects they denote.

The two first components are the main topics of this work and, in particular, the level of interaction protocols which constitutes the pragmatics of agent communication. Definitions of concepts like semantics, pragmatics, policies and protocols will be provided.

This thesis focuses on the area of formal tools to specify and design ACLs to facilitate *interoperability* in open normative multi-agent systems. Our main *hypothesis* states that:

> In order to specify a public and grounded ACL for open multi-agent systems, a pragmatic component to account for the social aspects of

communication must complement a motivational ACL semantics. Having two separate but complementary levels allows us to define a set of goal-based speech acts, and a set of conversation policies to regulate their use. Besides, a set of pragmatic principles may facilitate the achievement of the perlocutionary effects by helping agents in the interpretation process.

In order to meet with the objectives discussed in the previous section, the contributions of this thesis include:

1. **Conceptual issues**:
   a) Chapters 2 and 3 offer a review of existing ACLs, focusing on FIPA ACL and KQML. It will also be shown the problems these languages pose
   b) Chapter 4 provides a formal specification of a unified framework for ACLs, which include semantic and pragmatic levels of communication.
2. **Grounded** and **Formal**:
   a) In chapter 4, a computational and formal model in which our ACL is **grounded** to reason about the properties and behaviour of agents is defined.
   b) 5 presents extension of the Interpreted Systems Logics to express informational and Motivational concepts ($MLTL_I$) which works as a specification language for the ACL semantics, and a language $NLTL_I$ to define normative and dynamic notions, which constitutes the specification language of the ACL pragmatics. Both $MLTL_I$ and $NLTL_I$ are embedded into a computational model previously defined.
3. **Autonomous**: The Contextual and Perlocutionary requirements are addressed by the ACL pragmatics in chapter 6, so our ACL semantics are broad enough to allow agents enough autonomy to take decisions.
4. **Complete**: We add the commissive and declaratives types of speech acts to our ACL semantics (see chapter 5.2. Most of the ACLs do not define speech acts for these two categories. We built on the FIPA Communicative Acts Library to define our set of speech acts. The reason is that we aim to contribute towards the specification of standard ACLs by extending and improving the FIPA ACL specification.
5. **Public**: We contribute a grounded logic to specify the set of speech acts in chapter 5. Furthermore, their semantics are not dependent on agents' private mental states.
6. **Contextual** and **Perlocutionary**:

a) A formal definition of *right* as the central concept to be used for the pragmatics of the ACL.
b) A theory of agent communication pragmatics (NPRAG) which is based on normative and organizational concepts.
c) The specification of well-known FIPA protocols (Request, Query-if) using the ACL normative pragmatics (NPRAG).
d) We also present examples of normative conversational policies that regulate the use of the speech acts in specific scenarios.
e) We study the applicability and verifiability of our proposal. In particular, we will show how to code NPRAG policies in Prolog and how to give several levels of verification to the ACL pragmatic theory proposed in this thesis.

Being most of the contributions of this thesis a set of formal rules and definitions to specify analyze and reason about communication in open multi-agent systems, we present proofs (or their sketch in some cases) for some interesting properties of the two main logics proposed, including validity, soundness and completeness (see chapter 5). The properties shown by $MLTL_I$ and $NLTL_I$ transfer into the ACLs specified by them. As such, issues of evaluation are strictly related to the technical adequacy of the definitions and proofs presented. The above list of contributions points out in detail to where each of the thesis objectives is dealt with. Furthermore, we discuss in great detail at the end of each chapter the nature of each of the particular problems addressed in that chapter and the solution proposed.

The conversation protocols and policies presented in Prolog in chapter 6 were tested, with the aim of showing that our approach allows to deal with context and the perlocutionary effects in agent communication in a fairly strait forward manner. In that sense, we were interested in justifying some of the theoretical and formal claims made previously.

To complete the picture, it will avoid some misunderstandings to enumerate some points on which this thesis is not about. This work focuses on the external aspect of the system, that is, the public rules guiding conversation. Therefore, it does not discuss how agents might be built to operate in such systems. The private reasoning an agent makes after receiving a message and the agent's strategy to choose a response are left open. This is the ideal of interoperability for open multi-agent systems, namely, the ability of sharing languages and knowing how to use them without human intervention.

Systematic and widespread research on agent communication started in the early 1990s. After 15 years of research, we still desperately need to *ground* our ideas and intuitions as to what properties and functions agent communication languages should display, and how those languages can be embedded in multi-agent systems for their use. Otherwise there is the risk that the agent-based paradigm may suffer the same fate as expert-systems, that were once hailed as the new paradigm that would change the way we saw computing.

This work seeks to advance the work on formal accounts of agent conversations. The main purpose is not to advance sociology, philosophy of language, or game theory, but to show how some of these disciplines can be used to solve the problem of intention recognition in multi-agent systems. Furthermore, its main aim is to provide a grounded and public language so agents can communicate effectively. This thesis is a a contribution towards the idea of an open multi-agent system described in this chapter, and further develops the work published in several papers (Agerri and Alonso, 2006, 2005a,b, 2002a,b).

Specifically, a general overview and early versions of $MLTL_I$ are published in Agerri and Alonso (2005b) and Agerri and Alonso (2006). A partial formalization of $NLTL_I$ and the ACL normative pragmatics is given in Agerri and Alonso (2005a). Agerri and Alonso (2002b) and Agerri and Alonso (2002a) introduced the main intuitions behind our approach.

1.6 Outline

The outline of the thesis is as follows:

- Next chapter describes the research on human linguistic communication which constitutes the origin of ACLs, such as Speech Acts theory (Searle, 1969) and Grice's theory of implicature (Grice, 1975), and STRIPS-based theories (Cohen and Perrault, 1979) in computational linguistics.
- Chapter 3 is a critical literature review of the state of the art of agent communication. We present the problems these ACLs pose and enumerate a number of steps to be taken in order to overcome them. The remaining chapters describe each component of our proposal.
- In chapter 4, we define a computational model in which our ACL is grounded. Note that many of the problems of previous ACLs were caused by the fact that their specification languages cannot be related to a computational model.

- In chapter 5, we provide a complete set of speech acts which are specified using $MLTL_I$. Appendix A. extends this chapter to offer a reformulation of the FIPA Communicative Acts Library (FIPA CAL) based on $MLTL_I$. Our reformulation inherits the main intuitions behind FIPA CAL but put into practice with a grounded specification language.
- Chapter 6 shows how normative pragmatic principles facilitates and improves agent communication. It also shows how to define different policies to use the ACL semantics given in chapter 5 in different scenarios. Besides, we provide examples of interaction protocols which are reformulated using a normative pragmatic approach.
- We then study and show examples of the applicability of our proposal and discuss various issues about its application and verification.
- Finally, some conclusions shall be discussed.

2

Conceptual Foundations of ACLs

According to Genesereth and Ketchpel (1997), a software agent is any system which uses an agent communication language to exchange information. Now, what is communication? In general, 'communication' would mean *communication in a common language.* Understanding a common language means understanding the vocabulary and, most importantly, knowing how to use the vocabulary to affect the environment, to perform tasks, and to achieve goals. Thus, effective agent communication involves two aspects: understanding the *common language*, and using the language to achieve tasks and goals. Understanding the vocabulary alone does not enable agents to communicate. The use of a language is goal-based. Agents use ACLs to communicate with each other (sharing information and/or knowledge). ACLs define the type of messages (and their meaning) that agents may exchange, and the interaction protocols state how to use them. Agent-to-agent communication is key to understand the potential of the agent paradigm, just as the evolution of natural language was crucial to the development of human societies.

ACLs should allow agents to achieve their goals cooperatively, to acknowledge receipt of messages, to help others perform their tasks, to refuse, ask and request. In other words, ACLs should allow an agent to perform the same essential communicative actions that people perform when using natural language. ACLs should thus offer agents some functions that correspond to those provided by natural language to humans.

The starting point for most ACLs has been to look at the attempts to analyze human communication. As in natural language research, the two main issues when designing an ACL are the meaning of the utterances (i.e., illocutionary actions), and the analysis of conversations. In this chapter we offer a critical review of some approaches to the analysis of human linguistic communication that are the conceptual base of most of the theories of agent communication. Specifically, the section 2.1 introduces Austin's notion of performative and Searle's theory of Speech Acts. In section 2.2, we study Grice's theory of meaning and conversation. Section 2.3 shows a social-based alternative to the analysis of conversation. In section 2.4 we will look at the theories coming from computational linguistics. In particular, we will analyze STRIPS-based approaches. Finally, we will discuss the main issues presented and their influence to agent communication.

2.1 Theory of Speech Acts

The theory of speech acts (Searle, 1969) is a major paradigm in the study of natural language. It relates the form of utterances to the acts that are performed by uttering them. People do not just utter propositions, they perform illocutionary acts such as requesting or commanding (Austin, 1962).

2.1.1 Performatives

Austin (1962) established the original framework from which speech acts theory would later be developed. He argued that some utterances perform actions by virtue of being said. The speech act performed by the following example is a declaration. By virtue of performing this speech act, the agent actually gives a name to a ship:

1. I hereby name this ship the Stalin.

Austin claimed that the idea of reducing the study of natural language to logical truth-conditions was mistaken. Austin argued that some declarative sentences can be used without asserting their truth value. He called these kind of utterances *performatives*. Their main characteristics are their ability to change the state of the world and that they explicitly state what the speaker intends to achieve. Instead of truth conditions, performatives have a set of satisfiability conditions.

Austin distinguished between explicit and implicit performatives. Explicit performatives perform some acts by convention and explicitly show the intended goal of the acts. Classic examples are, among others, *promising*, *naming* and *betting*. Implicit performatives are the rest of the utterances that do not have a explicit performative. Austin first thought that performatives marked a special kind of utterances, but later argued that all utterances have a performative aspect. He identified three distinct actions that are simultaneously performed in a speech act:

- Locution: The actual physical utterance.
- Illocution: The conveying of the speakers' intentions to the hearer.
- Perlocution: Actions that occur as a result of the illocution.

According to Austin, the locutionary and illocutionary aspects could be detached in the analysis of communication, but it would not be easy to clearly distinguish between the illocutionary and the perlocutionary aspects. Similar problems have arisen when specifying the satisfiability conditions of the communicative acts in both FIPA ACL and KQML.

Artificial intelligence researchers have focused on the illocutionary aspect of the speech act to propose artificial languages for agent communication. The communicative acts defined by KQML and FIPA ACL correspond to the explicit specification of their components, namely, the illocution, locution and perlocution. The taxonomy of performatives verbs provided by Austin was later extended and improved by Searle (Searle, 1979).

2.1.2 Taxonomy of Speech Acts

Searle extended Austin's work and placed it within a more general theory of meaning, in which the main purpose is to list a set of satisfiability conditions for a successful *performance* of a speech act. Searle's most comprehensive work on speech acts is on the act of *promising*. Its satisfiability conditions are as follows (Searle, 1969, p. 59-61):

1. Normal conditions must hold for input and output.
2. The promise must have some content.
3. The promise must refer to some action in the future.
4. The object of the promise must be something good for the addressee.
5. The promise action cannot be something that would happen anyway.
6. The speaker must be sincere, that is, he must intend to do the action.
7. The act of promising places the speaker under an obligation to fulfill the promise. It is a commitment.
8. The speaker intends the addressee to recognize that a promise is being made and to do so by recognizing the speaker's intention.
9. The semantic rules of the language spoken by both speaker and addressee is such that the utterance is a promise only if condition 1 to 8 are satisfied.

Searle pointed out that 1, 8 and 9 hold for all speech acts while the rest are specific to promising. In order to be able to recognize a particular speech act, Searle proposed twelve different dimensions of speech acts (see Table 2.1), of which only three are central for the taxonomy he proposes: The illocutionary point, direction of fit and mental state. Since many verbs can be used in different acts, Searle's taxonomy refers not to performative verbs, but to the illocutionary acts using the verbs.

Dimension	Example
Illocutionary point	Some acts try to get the hearer to do something, some try to state something, some try to promise something.
Direction of fit	A speech act may be trying to state a fact or to bring about a new state to the world.
Psychological state	Beliefs (assert), intentions (promise), etc.
Force	Suggest or insist.
Status of participant	If a captain requests a soldier, it is an order.
Relation to interests	Boast or lament, congratulations or condolences.
Relation to discourse	Reply, object, deduce, conclude.
Allowed content	It relates the illocutionary force with the propositional content.
Speech required	Estimate can be done without speaking.
Institutional	Declaring war.
Implicit	insult, boast, threaten.
Performance style	Announce and confide can have same point and content.

Table 2.1. Dimensions of speech acts.

The first dimension, the illocutionary point, refers to the purpose of the type of act. The illocutionary point seems a bit too vague to be a criterion to classify acts, since a difference in the illocutionary point affects other dimensions such as the mental state and the direction of fit.

The direction of fit is not conclusive either, since some acts have *both ways* direction of fit, and both commissives and directives have the same, which for Searle was not very satisfactory. The direction of fit is a consequence of the illocutionary point.

As criterion to group illocutionary acts, the *mental state* is quite useful. The *mental state* expressed by the speech act refers to the propositional attitude of the speaker towards the propositional content of the utterance, and the illocutionary acts can be classified according to the mental state they refer to. Searle's theory of Speech Acts is embedded in a more general theory of mind (Searle, 1983).

Table 2.2 shows the taxonomy of speech acts. The taxonomy and its dimensions, like the mental state and the illocutionary point, have been widely used to specify communicative actions (speech acts) for ACLs like FIPA ACL and KQML.

Category	Mental State	Direction of fit	Illocutionary Point
Assertives	Belief in proposition	Words to world	Assertion
Directives	Want hearer to do action	World to words	Direction
Commissives	Intention to do action	World to words	Commision
Declaratives	None	Both ways	Declaration
Expressives	Many	None	Expression

Table 2.2. Searle's Taxonomy of speech acts.

2.1.3 Illocutionary Acts

Searle argued that the illocutionary act is the minimal unit of linguistic communication. Every illocutionary act consists of an illocutionary force F applied to a proposition p, $F(p)$. The illocutionary force establishes what type of speech act an utterance is. Searle identifies seven different components of the illocutionary force:

1. Illocutionary point: The illocutionary point is achieved if the act is successful. For example, the illocutionary point of a promise to do act p (commissive), is that the speaker commits herself to do p. The act will be successful if the promise is kept in the future.
2. Degree of strength of the illocutionary point: It distinguishes between "pass me the salt!" and "could you please pass me the salt?" Both are directives, but the former is a command while the later is a plea.
3. Mode of achievement: Set of conditions under which the illocutionary point has to be achieved. For example, a command may require a position of authority on behalf of the speaker.

4. Propositional content: Imposes what can be in the propositional content for a specific force F. Declaring "I promise I will come tomorrow to the University" would not be considered a promise for an academic; it is their *obligation* to come.
5. Preparatory conditions: The conditions that should hold for the success of the speech act.
6. Sincerity conditions: Conditions on the mental state of an agent. The propositional content of the illocutionary act should be identical to the propositional content of its associated mental state. This condition would guarantee that an agent does not say something it believes is false.
7. Degree of strength of the sincerity conditions: For instance, 'requesting' and 'begging' do not suggest the same level of desire.

These components take into account how the same propositional content can be expressed in many ways, that is, how applying different illocutionary forces will produce different illocutionary acts. For example, the same propositional content, that the speaker wants salt, can be expressed by uttering 3 different sentences:

2. (a) Can you pass me the salt?
   (b) I order you to pass me the salt.
   (c) I wish you would pass me the salt!

Some illocutionary acts are encoded by some specific grammatical structures (questions, orders, etc.), some are encoded by particular verbs (promise, declare, etc.), and others are performed by asserting that someone is performing them. That is, the speaker is *asserting* that the utterance is a promise, an order, etc.

It is assumed that the sender performs a speech act in order to achieve a goal; the accomplishment of the goal will depend on the satisfaction of the speech act, that is, on its *communicative meaning.* An utterance is seen as an action that achieves (or at least tries to achieve) some goals for the speaker. As Bach and Harnish (1979) stated,

> "An illocutionary act is successful if the speaker's illocutionary intention is recognized by the hearer. These intentions are essentially communicative because the fulfilment of illocutionary intentions consists in hearer understanding."
>
> (Bach and Harnish, 1979, p.15)

The illocutionary point of a performative utterance is given by the performative verb. Using again the example 1.:

1. I hereby name this ship the Stalin.

The verb *Name* refers to the illocutionary point. It is important to note that often it is not easy to identify the illocutionary point, since a utterance usually encodes more than one illocutionary act. An indirect illocutionary act occurs when in a single utterance more than one illocutionary acts are performed simultaneously. For example, one can perform an illocutionary act explicitly conveying another illocutionary act implicitly. In this case, the speaker often relies on the addressee recognizing the communicative intention. For example, if a person is asked:

3. Do you know the way to the Palace Hotel?

The speaker certainly is not expecting a yes-no answer; that would be inappropriate since the speaker is not only *asking* a question about the knowledge of the addressee but also *requesting* to tell them the way to the Palace Hotel. Searle and Vanderveken (1985) also provide the following example. If someone says:

4. Sir, you are standing on my foot!

The speaker is not only informing the addressee about their location, but requesting to that person to get off her foot. Thus, the non-literal primary speech act, which is the indirect speech act, is performed implicitly by means of performing a literal secondary speech act. Similarly, humans do distinguish between the actual explicitly expressed meaning and the inferred meaning (request), that is, quite often we are able to identify the illocutionary force of the speech act performed. Any theory of communication that does not account for this process is not complete. Agent communication theories have tried to solve this issue by explicitly specifying the (illocutionary) meaning of a set of speech acts available to agents (as in FIPA ACL and KQML). The communicative actions (or performatives) of these ACLs intend to represent the explicit illocutionary force of every type of speech act.

The conceptual foundations of agent communication lie firmly on the theory of speech acts. For example, KQML defines *performatives* as the communicative actions available for agents to communicative. The Communicative Acts Library of FIPA (FIPA CAL), corresponds to a library of speech acts, and the

Feasibility Preconditions have a strong reminiscence of the preparatory conditions of Searle's theory. Besides, like speech acts, ACLs do not offer an account for agent linguistic interactions; there are only some hints about how the identification of an illocutionary force may depend on the context set up by the previous communicative actions performed. Several authors (Holmback et al., 1999) have tried to fill this gap by adopting Grice's studies on conversation in natural language pragmatics to account for interactions in agent communication. In so doing, they implicitly acknowledged the necessity of developing an agent communication pragmatics, an issue which, in our opinion, has been largely overlooked by the agent communication community. In the next section, we study Grice's contribution to natural language pragmatics.

2.2 Grice's Theory of Communication

Grice's work on linguistic pragmatics is not limited to the study of conversation, but he also developed a theory of speaker's meaning, which is crucial to understand his work on conversation. Thus, first we will introduce the notion of speaker's meaning and then we will show how this is applied to the study of conversation.

2.2.1 Utterance and Speaker's Meaning

Grice (1957) introduced an idea of intentional meaning whose satisfaction lies in the recognition of that intention, namely, the Meaning-intention (M-intention).

> A meant something by uttering x is (roughly) equivalent to "A intended the utterance of x to produce some effect in an audience by means of the recognition of *this intention*".(Grice, 1957, p.220)

According to this definition, communication is an intentional action which is achieved when the intention is recognized. This allows to distinguish between speaker's and sentence meaning. Take for example the utterance of:

5. Your fitness level is excellent.

Although at first glance it seems to be a compliment, it could, in an appropriate context, be interpreted as an irony. The additional inference necessary to understand the implicit meaning (irony) is an implicature (Grice, 1975).

As we saw when discussing cases of indirect illocutionary acts in the previous section, utterances mean more than their explicit interpretation. Along with

M-intentions, Grice distinguished between *what is said* and *what is implicated* in order to work out *what is communicated* (Grice, 1975); *what is said* consists of the *semantic content* of the sentence plus two pragmatic aspects: reference fixing and resolution of any grammatical ambiguities; *what is implicated* represents those aspects of meaning that cannot be derived from *what is said* or, in other words, *what* the speaker *conversationally implicates* by using a sentence. Although Grice distinguished between several types of implicatures (see Figure 2.1), in the next section we limit our review to the conversational implicatures, which are directly related to the purposes of this thesis.

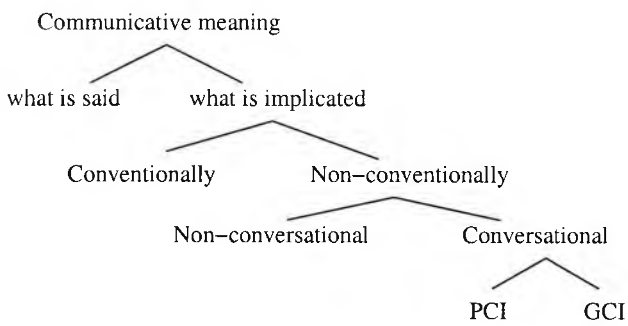

Fig. 2.1. Gricean schema of communication

2.2.2 Conversational Implicatures

Grice considered two types of conversational implicatures: those in which an implicature is carried in a particular context are particularized conversational implicatures, whereas those in which an implicature arises in absence of a specific context are called generalized conversational implicatures. As an example of particularized conversational implicature, consider

6. A: How is X doing in the bank?
   B: He is fine, not in jail yet.

where detailed contextual information is needed in order to work out what B *meant*, for instance, that X may be a compulsive thief, or that their colleagues are plotting to get him into trouble, etc. On the other hand, Grice's most famous example of generalized conversational implicature is:

7. X is meeting a woman this evening.

Unlike particularized conversational implicatures, the generalized ones are cases where the implicature is normally carried by saying p, and not by saying p at a particular moment and in a specific context.

> "The use of certain form of words in an utterance would normally (in the absence of special circumstances) carry such-and-such an implicature or type of an implicature." (Grice, 1975, p. 37)

In the case of 7., the implicature conveyed would be something like "X is meeting a woman who is not his wife, mother or sister." In order to explain how (conversational) implicatures arise, Grice proposed a general cooperative principle of communication, which is based on the conventional use that we (humans) do of natural language. This principle represents a set of conventions which all parties are aware of.

2.2.3 Cooperative Principle

As a complement to the cooperative principle, Grice formulates four conversational maxims which if observed will fulfil the cooperative principle (Grice, 1975, p.26-27):

Quantity:	Make your contribution as informative as is required.
	Do not make your contribution more informative than is required.
Quality:	Do not say what you believe to be false.
	Do not say that for which you lack adequate evidence.
Relation:	Be relevant.
Manner:	Avoid obscurity of expression.
	Avoid ambiguity.
	Be brief.
	Be orderly.

The general cooperative principle of communication is defined by Grice as follows:

> "Make your conversational contribution such as is required, at the stage at which it occurs, by the accepted purpose or direction of the talk exchange in which you are engaged." (Grice, 1975, p.26)

These maxims represent conventions which the participants are aware of. In human communication it is possible to flout these maxims in order to achieve a particular goal. Compare, for example, the following two utterances (Grice, 1975, p.37):

8. (a) Miss X sang "Home Sweet Home."
   (b) Miss X produced a series of sounds that corresponded closely with the score of "Home Sweet Home."

It is important that the audience are aware of some social guidelines in order to understand the implicature conveyed by the utterance of 8.(b). In this case, the maxim of manner is flouted since it seems apparent that the speaker was intentionally being obscure. However, the maxims are not always fulfilled. For those cases, Grice pointed out that even when an utterance apparently fails to follow them, the hearer still assumes the cooperative principle is being followed. Such assumption would motivate the additional inference that is needed to retrieve the conversational implicature.

Flouting the maxims is not the only way of generating conversational implicatures. In fact, Grice identified three different ways. First, implicatures can arise because the maxims are observed. Second, implicatures are conveyed because the maxims appear to be unintentionally flouted. The third one is the intentional flouting of the maxims as in 8.(b). As an example of implicatures in a conversation where the maxims are being followed, consider example 9.:

9. A: I am out of petrol.
   B: There is a garage round the corner.

In this example (Grice, 1975, p.32), B would be flouting the maxim of Relation unless B thinks that the garage is open and that it sells petrol. The implicature that the garage is open and it is possible to buy petrol seems to be the best interpretation available. Therefore, the implicature was generated with no clear evidence that any maxim is not observed. The following example (Grice, 1975, p.32) shows how implicatures can arise by the unintentional flouting of the maxims.

10. A: Where does C live?
    B: Somewhere in the South of France.

B's answer is less informative than required, which is an infringement of the first maxim of Quantity. However, this can be explained if A assumes that B does not know the precise location where C lives. B could not be more informative since they will be saying something that would flout the second maxim of Quality. Grice characterized the notion of conversational implicature as follows:

"By saying p, the speaker U conversationally implicates q iff

(i) he has said that p,
(ii) there is no reason to suppose that he is not observing the maxims, or at least the Cooperative Principle,
(iii) he could not be doing this unless he thought that q,
(iv) he knows (and knows that I know that he knows) that I can see that the supposition that he thinks q is required,
(v) he has done nothing to stop me thinking that q,
(vi) he intends me to think, or is at least willing to allow me to think, that q,
(vii) and so he has implicated that q."

That is, according to Grice, it is possible to explain and predict the generation of an implicature. In this sense, implicatures are *calculable*, which is one of the properties Grice proposed to help identifying them. These properties are listed below:

1. **Cancellability:** Implicatures can be explicitly or contextually cancelled. In the first case, a clause is added to correct the first utterance. In the second one, the form of the utterance that carries the implicature is used in a context that manifests that the speaker is cancelling it.
2. **Calculability:** The schema of conversational implicature defined above states how an implicature can be calculated.
3. **Non-detachability:** Implicatures are non-detachable in the sense it is possible to generate the same implicature by rephrasing the utterance. It is not possible to find another way of saying the same thing which does not convey the same implicature (except those generated by exploiting the maxims of Manner).
4. **Non-truth conditional:** Implicatures are not generated by *what is said*, but only by the saying of *what is said*. In other words, implicatures do not depend on the truth value of *what is said*.
5. **Non-uniqueness:** Utterances often generate more than one implicature. In this case, the implicature will be the disjunction of all the implicatures generated by the utterance.

Non-uniqueness and calculability illustrate the main weakness in Grice's theory. There is an inferential gap in the calculability of the implicature. In particular in clauses (iv) and (vi). It is not clear how the addressee is able to reach an interpretation that matches the speaker's intention. Consider the following example:

11. Kasparov is a machine.

Several interpretations can be attributed to this utterance. First, that Kasparov, the chess champion, is systematic and tireless; second, that his character is not very warm; third, that there is a machine called 'Kasparov'; fourth, that he lacks imagination, etc. According to Grice, examples of metaphor would be understood as an implicature caused by the flouting of the first maxim of Quality. What is not explained is how from concluding that this utterance is false, that Kasparov is a person, the addressee reaches the interpretation that he has a very cold character and not any of the other interpretations.

Paraphrasing Grice, the calculation of the implicature could be as follows: By saying 'Kasparov is a machine', the speaker U conversationally implicates that "Kasparov is tireless and systematic" iff

(i) he has said that 'Kasparov is a machine',
(ii) there is no reason to suppose that he is not observing the maxims, or at least the Cooperative Principle,
(iii) he could not be doing this unless he thought that 'Kasparov is systematic and tireless',
(iv) he knows (and knows that I know that he knows) that I can see that the supposition that he thinks 'Kasparov is systematic and tireless' is required
(v) he has done nothing to stop me thinking that 'Kasparov is systematic and tireless',
(vi) he intends me to think, or is at least willing to allow me to think, that 'Kasparov is systematic and tireless',
(vii) and so he has implicated that 'Kasparov is systematic and tireless'.

Clauses (iv) and (vi) state that the addressee assumes that the utterance is not true, so the speaker intended to communicate an additional meaning. This meaning has to be recognized and, therefore, the addressee concludes that some of Kasparov's attributes resemble a machine, that is, that Kasparov is systematic and tireless. The problem is that this utterance generates several implicatures and it is not explained how the hearer chooses exactly the implicature the speaker intended to communicate. There is a gap to bridge between the different interpretations conveyed by the utterance and the addressee's *correct* interpretation[1]. Therefore, Grice's theory does not fully explain the interpretation process.

[1] 'Correct' in the sense that it corresponds to the speaker's intention.

The review of Grice's theory of conversation is (conversationally) relevant because some authors have tried to adapt the cooperative principle and its maxims as a principle to guide agent communication (Holmback et al., 1999). This idea fits well with cooperative agents in closed environments, but they have proved inadequate for its application in open environments. In the review of ACLs offered in section 3, it will be argued that the intention recognition problem is present in agent communication. It will also explain how a pragmatics for ACLs based on a set of conversation policies can guide communication and bridge the gap between the hearer's interpretation and the speaker's intention.

As an alternative to the mentalistic pragmatics represented in the theories of Searle and Grice, Sacks (1972) proposes a social approach to the study of conversation. Recently (see Singh (2000); Fornara and Colombetti (2004)), several agent communication theories focus on the social consequences that the performance of communicative actions entail. In our view, Sacks's work constitutes a good starting point to develop a social-based pragmatic theory for agent communication.

2.3 Social Pragmatics

Sacks's work focused at how the conventional use of language constrains the posterior responses at certain points of conversation. By studying conversations, several patterns in the linguistic interaction were identified. Given the complexity of natural language, it seems reasonable to think that certain conventions are regularly repeated in order to automatize and simplify its everyday use. As argued with respect to the example "It is cold in here", there are situations in which human communication works in an automated way, as it happens in casual conversations about the weather, greetings, etc. In those cases, social constrains limit the interaction available in conversation.

When an agent performs an illocutionary act, the utterance can trigger several interpretations which in turn yield a number of different responses. In mentalistic pragmatics, successful interaction depends on the decision to choose the better utterance to express the speaker's meaning and the ability of the addressee to recognize the speaker's intention. Sacks' approach tries to identify patterns of communication that simplify the task of choosing the appropriate utterance. One of these patterns of communication are the *adjacency pairs.* Adjacency pairs have an ordering such that the first pair determines the possibilities for the second. Typical examples include greetings and question-answer pairs, which admit only one type of answer. Other examples can trigger a set of possibilities, such as a request or a rejection.

13. A: Let's drive to Edinburgh this weekend.
    B: The car is broken.

In this case, the meaning of B's utterance is a rejection simply because, according to the adjacency pairs approach, the responses have to be either a rejection or an acceptance. Not only the set of available responses have been reduced but the meaning of the answer is determined too. The addressee needs to know what utterances are related to the previous ones. This view can additionally be used to deal with tasks like initiating and terminating conversations. The meaning of an utterance is not only determined by the speech act, but also by the speech context. The context refers to all the external factors that have an influence on the meaning of the illocutionary act. Three aspects of the context can be identified:

- The conventions that rule the conversation in the situation in which it occurs.
- The status of the participants or the roles they play in the conversation will also affect the meaning of the speech act.
- The relation between the speech act and the rest of the discourse.

If the conversation follows a protocol, the protocol is not considered to be a separate aspect of the context. Conversely, the protocol is a useful way of collecting many aspects of the context such as a set of available illocutionary acts and the roles of the participants. Thus, in social pragmatics meaning is not an assignment of communicative intentions to a speaker, but a system of conventions which helps agents in the interpretation process. When following a protocol, agents exploit a system of conventions in order to help the hearer to interpret an illocutionary act.

Several examples provide evidence of the existence of patterns which conversations follow. A classic situation is given in phone conversations, where there are a set of obligations for the person answering the call, such as taking messages, etc. Considering the interactions as protocols has two benefits: It affects the meaning of utterances and it constrains the responses available to an utterance. For example, in a customer-based environment, when a seller utters the sentence "Can I help you?", it constrains the responses available to either a request or a refuse.

The concept of *convention* has appeared throughout this chapter. The conventional meaning of illocutionary acts should play an important role in the

the specification of ACLs, because it captures the public and social aspect of communication. In open systems, mentalistic and subjective meanings can not be verified if they are private.

The study of conventional protocols is useful to specify interactions between artificial agents, as it will be shown in the next chapter when discussing interaction protocols in agent communication. A social-based pragmatics constitutes a good starting approach, as it is the adoption of a computational approach to the analysis of utterances. The later is used to specify a set of speech acts (ACL semantics) and the former inspires the development of a social-normative policies for agent communication (ACL pragmatics). Next section reviews the computational approach to the theory of speech acts, which has been highly influential in the specification of communicative actions in both FIPA ACL and KQML.

2.4 Speech Acts in Artificial Intelligence

The theory of Speech Acts has been so influential to research in Artificial Intelligence because it suggests that language can be generated and understood using techniques for plan generation and recognition (see Lee (1998) for a comprehensive discussion on the use of Speech Acts theory in computational linguistics). STRIPS-based formalisms are the base of the semantic specification of KQML and FIPA ACL.

From a planning perspective, to understand an utterance the addressee has to recognize the plans behind the utterer's communicative action:

> "Successful communication will only normally take place if the addressee is able to recognize the plans behind the speaker's utterances. Only then will the addressee be able to give genuinely cooperative responses and sensibly anticipate future utterances. The process of plan recognition is unfortunately relatively poorly understood. Nonetheless, it is a task of quite general usefulness." (Gazdar and Mellish, 1989, p.398)

The work of Cohen and Perrault (1979) and Allen and Perrault (1980), has influenced much of the later work in agent communication. They showed that speech acts can be seen as plan operators. In this view, understanding an utterance consists of recognizing the underlying plan of the agent. Cohen and Perrault (1979) discuss a planning system for generating speech acts. The system models the user's beliefs and intentions and matches them with illocutionary acts.

Allen and Perrault (1980) extends this work by defining speech acts in terms of STRIPS plan operators (Fikes and Nilsson, 1971). The operators consist of a header with constraints on instantiation of its parameters, a set of preconditions, an effect list and a body which lists actions and goals which must be achieved for the action to be performed. For example, the definition of the illocutionary act of informing is as follows:

$INFORM(Speaker, Hearer, P)$
Preconditions: *Speaker knows P (P is true and Speaker believes P)*
Effects: *Hearer knows P*
Body: *Hearer believe Speaker want(hearer know P)*

The definition of *inform* states that the act of a speaker informing a hearer of some proposition P requires:

1. That P is true and that the speaker believes P.
2. That the effect of informing is that the hearer will also know P.
3. That informing consists of getting the hearer to believe that the speaker wants the hearer to know P.

Cohen et al. (1982) make use of such definitions to develop a question answering system. They argue that question answering is a special type of natural language dialogue and describe an analysis of transcripts of users questioning a real question answering system (PLANES) and an expert posing as an ideal system. They found out that:

- Users expect the system to respond to the users unstated implicit goals.
- Users often make complex requests in several utterances, each one providing more detail than the previous utterances. Such later utterances are shaped by the system's replies.
- Users expect the system to be sensitive to and correct any incorrect assumptions that they might hold.

Cohen et al. (1982) believed that a system must be able to recognize the intention and underlying plan behind any utterance. They distinguish between keyhole plan recognition and intended plan recognition. The former is done by simply observing the user's actions while the latter requires that the plan recognized is the one the user intended to be recognized. Intended plan recognition relies upon mutual beliefs and is conducted by a set of inference rules. As an example, suppose that a user makes some utterance which communicates that he wants the system to perform some action A:

(i) The system would believe that the user wants to perform A.
(ii) Besides, having assumed that the act is intentional, the system would believe that the user wants the system to believe that the user wants the system to do A.

Allen and Perrault (1980) argue that there are many inferences available from any given utterance. In order to control the number of inferences made, they state that the intended plan recognition should terminate once a line of inference matches an expected goal of the user. This theory allows the use of plan recognition techniques to understand the intentions behind a given utterance. Such techniques involve ascribing plausible goals to the speaker and then trying to formulate a plan involving the speech act which achieves the ascribed goals.

The planning view has been widely used in agent communication to specify illocutionary acts as STRIPS operators. The form of the *communicative acts* of FIPA ACL, and of the *performatives* defined in KQML are significant examples of this trend.

2.5 Discussion

Summarizing, the analysis of communication is a two-fold activity: We need an account of the individual meaning of utterances and of the study of how sequences of utterances are formed to structure conversations. In this chapter we have reviewed the most influential theories of natural language pragmatics to agent communication: The theory of speech acts tries to provide a definition of the meaning of single utterances. Regarding the interpretation of utterances in conversation, two different accounts were analyzed:

- Grice's cooperative principle and the "STRIPS and speech acts" approach represent a mentalistic point of view, in which the key issue of the interpretation is to recognize the speaker's intention.
- Sacks' account of conversation in terms of conventional meaning and conversation patterns offered an alternative to mentalistic theories. In this view, the interpretation process is guided by protocols which structure the conversation.

Most ACLs developed to date are based on the analysis of utterances as speech acts. Thus, the semantics of the ACL usually consists of a set of communicative actions defined as planning operators, that is, in terms of preconditions and effects (as in FIPA ACL and KQML). However, the study of conversation has not been so productive. In the next chapter we shall see attempts to use a version of Grice's cooperative principle for agent communication, and several protocol-based ACLs. The main problem of these approaches is that none of them consider the semantics and pragmatics in the same framework, in a way that the communicative meaning of an speech act does not depend only on the ACL semantics but also in the context in which is used. It shall also shown the shortcomings that these proposals present and discuss possible ways of tackle them. As in natural language pragmatics, the interoperability problem is still the *holy grail* of agent communication.

3

Agent Communication Languages

In the previous chapter we discussed the research on natural language pragmatics that has influenced the development of agent communication languages. Speech Acts theory (section 2.1) provided an analysis of utterances which has been largely used to specify speech acts in agent communication. In this view, an utterance consists of two parts: a proposition which is the object of the speech act and a function indicating the type of speech act that is performed. Section 2.2 introduced Grice's mentalistic account of conversation in which the interpretation process is successful if the communicative intention is recognized. Section 2.4 showed how speech acts and Artificial Intelligence (AI) techniques are combined to facilitate the interpretation process which is seen as a process of planning recognition. Sacks's (2.3) theory is based on a social view conversation, which is seen as a protocol which agents can follow. We have also shown how the ultimate goal of natural language pragmatics, explaining how an agent understands the speaker's intention and responds accordingly, remains to be solved.

The aim of this chapter is to discuss a set of requirements for ACLs. In order to do so, we will show that, as with natural language pragmatics, ACLs in the form proposed so far present several shortcomings that prevent them to be an effective tool of communication in open multi-agent systems. We will start with some conceptual preliminaries to clarify the terminology to be used in the rest of the thesis. Then, we shall offer a critical review of existing approaches to the specification of ACLs. Section 3.2 analyzes three types of approaches to the specification of ACL semantics: mentalistic, procedural and social. Sections 3.2.3 and 3.2.4 discuss the procedural and social approaches, whereas sections 3.2.1 and 3.2.2 present the most influential ACLs: FIPA ACL (Foundation of Intelligent and Physical Agents) and KQML (Knowledge Query Manipulation

Language). In section 3.3 we study various attempts to regulate conversations in agent communication. Sections 3.2.5 and 3.3.7 discuss the problems posed by the the semantics and interaction protocols, respectively, of the ACLs proposed so far. We shall finish the chapter analyzing the desirable requirements that ACLs should meet for their use in open multi-agent systems.

3.1 Terminological Disquisitions

Chapter 2 introduced most of the central concepts that will be used in agent communication: Performatives, speech acts, illocutionary acts, and perlocutionary act. However, the use of some of these concepts from natural language pragmatics differs from their use in agent communication. This section aims to define the concepts used for the development of ACLs.

Section 2.1 introduced the original concept of *performative*. According to Austin (1962), a *performative* is an utterance which is not true or false. In other words, a performative is a sentence in which the employment of a particular illocutionary force is made explicit by naming the force in the sentence itself. Performative utterances can be recognized by the presence of a *performative verb* such as promise, request, apologize and name. In agent communication, a *performative* can be an explicit description of the illocutionary force of the speech act or, as in KQML, a performative denotes both the illocutionary force and the speech act.

In the literature on the theory of speech acts, a speech act is an act that a speaker performs when making an utterance. Most of the work on speech acts is actually on illocutionary acts. Illocutionary acts are speech acts which consists of a propositional content and an illocutionary force whereby the speaker promises, apologizes, asserts, and so on. When performing a speech act, the speaker may intend to produce a particular effect in the addressee. This is called the *perlocutionary act.* In agent communication, the concept of speech acts can describe several situations, like a message being sent, the effect of message passing, the interpretation of the receiver, or the interpretation of the performance of the act in its context. FIPA ACL also uses the concept of *communicative act* to refer to speech acts. Thus, in its STRIPS-like form, a communicative act consists of a set of preconditions (illocutionary act) and a set of rational effects (perlocutionary act). Note that *communicative act* can refer to both the performative verb (inform) and the speech act.

Summarizing, there seems to be, in agent communication, a propension for using the same term to denote different concepts. As it has been explained, this is the case with terms such as *performative*, *speech act*, and *communicative act*. Homonymous terms create confusion and ambiguity. In order to avoid misunderstanding, we will use these terms in a traditional way (Austin, 1962; Searle, 1969) using both *speech acts* and *communicative acts* as equivalent terms to refer to the *action of sending a message*. We restrict the meaning of *performative* to the predicate that expresses the illocutionary force of the speech act. For example, a communicative act of FIPA ACL, *inform*, consists of a performative and a content such as:

$$inform(x, y, \phi)$$

In this example, *inform* is the performative, x and y denote the sender and the receiver respectively, and ϕ is the content of the message. In the rest of the chapter, the terms *speech act*, *communicative action*, *performative* and *perlocutionary act* will be used with the meaning defined above, not the meanings given in FIPA, KQML or some other ACL.

3.2 Communicative Acts

The main components of an ACL, as in every other language, are syntax, semantics and pragmatics. The syntax of an ACL corresponds to the format of a message or communicative action: a performative, a sender, a receiver and the content. The semantics of communicative actions define their meaning, which is a function of the illocutionary force of the performative and of its content. There are two different levels of semantics: On the one hand, the specification of the meaning of the performatives; on the other hand, the specification of the language in which the content is expressed. The semantics of ACLs refers to the meaning of the performatives which define the illocutionary and perlocutionary aspects of the communicative action. When specifying ACLs, it is generally assumed that the content language can be any existing language such as PROLOG or KIF. This thesis will also take this approach. The last component, the ACL pragmatics, is concerned with the use of the language and how it is affected by the context. ACL pragmatics will be discussed in section 3.3.

3.2.1 FIPA ACL

The aim of the FIPA consortium is to promote the success of emerging agent-based applications, services and equipment. In order to do so, it provides specifications that maximize interoperability across agent-based applications (Agentcities, 2002). The FIPA repository consists of:

- Applications: Documents related to the typical application scenarios of agent-based technology.
- Abstract Architecture: Abstract specification of agent platforms.
- Agent Communication: Specification of the communication language for agent interaction: FIPA ACL. This includes the structure of the language, the content representation languages, ontologies, etc.
- Agent Management: It deals with the normative context in which agents interact. That includes control of agents across different platforms.
- Agent Message Transport: Specifications of transport services for the transport and representation of messages across different network transport protocols.

Agent Communication and Agent Management are the most developed components of FIPA. The goal of standardization of FIPA refers mainly to these two issues. FIPA ACL specifications are based on a library of communicative acts, a library of conversation protocols, and a library of content languages. FIPA has been implemented in JADE (2002), although its implementation does not include the formal semantics of the ACL.

The syntax of FIPA messages is similar to KQML. A set of parameters characterize the messages of the language (see Table 3.1).

(*inform*
:sender (agent-identifier :name i)
:receiver (set (agent-identifier :name j))
:content (news p)
:language FIPA-SL
:ontology breaking-news)

Table 3.1. Description of inform.

The form of FIPA messages is reminiscent of an speech act, that is, they consist of an illocutionary force and a propositional content: $F(p)$. The illocutionary force is represented by a set of performatives which conform the FIPA CAL (Communicative Acts Library). Table 3.1 shows the structure of the communicative act *inform*. The performative denotes the kind of speech act performed. The rest of the parameters describe the participants, the propositional content, the language used for the propositional content and the lexicon.

The semantics is defined in terms of STRIPS-like preconditions and postconditions. The FIPA ACL is based on a language developed for the ARCOL system (ARtimis COmmunication Language). Since ARCOL was originally designed for human-computer dialogues, some assumptions made then may no longer be adequate. In particular, the assumption of *sincerity* of agents has been widely criticized.

FIPA CAL

FIPA ACL does not make any commitment to a particular content language; however, agents need to have some understanding of the Semantic Language (SL) to work out the meaning of FIPA ACL communicative actions. SL is a modal logic based on the logic of intention of Cohen and Levesque (1990), with operators for beliefs, desires, uncertain beliefs (U) and intentions (PG, Persistent Goals).

ARCOL uses the SL language for the semantics of communicative acts. SL uses a KD45 model. A description of the most important elements of SL follows:

- Variables: i and j denote agents; e, $e'\ldots$,e_1, denote event sequences.
- Action expressions: $a_1|a_2$ denotes a sequential action; $a_1;a_2$ refers to a non-deterministic choice, that is, a sequence of the type a_1 or a_2.

A set of modal operators represent some propositional attitudes:

- $B_i\phi$, i believes that the proposition ϕ is true.
- $U_i\phi$, i is uncertain about ϕ but thinks that ϕ is more likely than $\neg\phi$.
- $Feasible(a, \phi)$, action a can be performed and proposition ϕ will be true.
- $Done(a, \phi)$ an action a has been performed and proposition ϕ was true before the execution.
- $C_i\phi$, the proposition ϕ is part of the choices of i.
- $Agent(i, a)$, means that i denotes the only agent appearing in action a.

Some abbreviations are also defined:

- $Possible(\phi)$ is an abbreviation for $(\exists a)Feasible(a, \phi)$.

- $B_i if$ abbreviates $B_i\phi \vee B_i\neg\phi$.
- $U_i if$ corresponds to $U_i\phi \vee U_i\neg\phi$.

By using the primitives, it is possible to define other operators such as intention, $I_i\phi$, to express that an agent i intends to achieve that ϕ be true. As in the theory of Cohen and Levesque (1990), intention is defined as a persistent goal with commitment.

The meaning of FIPA performatives is defined in terms of feasibility preconditions (FP) and rational effects (RE) of the communicative actions. The FPs define the conditions that have to be true before an agent may perform a communicative action. The RE is the perlocutionary aspect of the speech act, that is, the expected effect that an agent wants to produce by performing an action. For example, the FP of *inform* is that the sender should actually *believe* the truth of the propositional content expressed by its communicative act. The definition of *inform* in FIPA ACL is as follows:

$$< i,\ inform\ (j,\ \phi) >$$
$$\text{FP: } B_i\phi \wedge \neg B_i(B_j if\phi \vee U_j if\phi)$$
$$\text{RE: } B_j\phi$$

That is, the sender holds that the proposition ϕ is true; it (the sender) does not believe that the receiver has any knowledge about the truth of the proposition $\neg B_i(B_j if\phi \vee U_j if\phi)$; and it intends that the receiver should also come to believe that the proposition is true (rational effect RE: $B_j\phi$).

There are several properties that model the performance of actions:

Property 1

$$\models I_i p \Rightarrow I_i Done(a_1 | \ldots | a_n, true)$$

where $a_1, \ldots, a_n$ are all the actions of type a_k, where the actions a_k satisfy:

1. $(\exists x) B_i a_k = x$ i.e., actions of type a_k exist
2. p is the RE of a_k
3. $\neg C_i \neg Possible(Done(a_k, true))$ i.e. does not desire that a_k be impossible.

This property establishes that if an agent i intends to bring about a fact p, and it believes that p is a RE that the agent i intends to achieve, then i intends to do some action a. This property gives the agent the ability to plan an action to achieve its RE.

Property 2

$$\models I_i Done(a) \Rightarrow B_i Feasible(a) \vee I_i B_i Feasible(a)$$

The second property establishes that if an agent i intends to perform an action a then either it believes it is feasible, or it intends to believe it is feasible to perform an action a.

Property 3

$$\models I_i Done(a) \Rightarrow I_i RE(a)$$

If an agent i intends to perform a communicative act a then it intends to bring about the RE of action a.

Property 4

$$\models B_i(Done(a) \wedge Agent(j, a) \Rightarrow I_j B_i I_j RE(a))$$

This property is called *intentional effect.* It establishes that when agent i observes agent j performing an action a, i should believe that j intends i to believe that j intends to achieve RE of action a. The nesting of attitudes aims to make public its intention of achieving $RE(a)$.

Property 5

$$\models B_i(Done(a) \Rightarrow FP(a))$$

If an agent believes that an action has been performed then the agent believed the FPs before the action was performed. Moreover, the FPs would persist after that. Thus, when an agent observes the execution of an action a, it believes that the persistent FPs of a hold.

FIPA CAL is composed by those communicative acts that are considered essential to the development of agents. There are two types of speech acts in the FIPA CAL: A set of primitives composed by inform, request, confirm and disconfirm (see Table 3.2), and the rest of the speech acts defined in terms of these primitives.

Request is a directive communicative act whereas inform, confirm and disconfirm are assertives. FIPA CAL does not include any other type of speech acts. The REs of inform, confirm and disconfirm are to modify the set of beliefs of the addressee. All these three speech acts require sincerity: $B_i\phi$ in the

$< i, inform(j, \phi) >$ $FP : B_i\phi \wedge \neg B_i(B_j if\phi \vee U_j if\phi)$ $RE : B_j\phi$	$< i, request(j, a) >$ $FP : FP(a)[i/j] \wedge B_i Agent(j, a) \wedge \neg B_i PG_j Done(a)$ $RE : Done(a)$
$< i, confirm(j, \phi) >$ $FP : B_i\phi \wedge B_i U_j\phi$ $RE : B_j\phi$	$< i, disconfirm(j, \phi) >$ $FP : B_i\neg\phi \wedge B_i(B_j\phi \vee U_j\phi)$ $RE : B_j\neg\phi$

Table 3.2. Primitive speech acts in FIPA.

case of *inform* and *confirm*, and $B_i\neg\phi$ for *disconfirm*. The difference between these speech acts is given in the description of the context in which they are applicable, that is, according to the *feasibility preconditions* (FPs).

Wooldridge (1999) proposed a multi modal logic to reason about agents' knowledge and time to give the semantics of some FIPA speech acts. Wooldridge aims to verify the ACL and to do so he uses a logic where he can attribute knowledge to agents in terms of their state. The precondition of an *inform* message is:

$$do_i(inform_{i,j}(\phi)) \rightarrow K_i(\phi)$$

That is, agent i knows ϕ before sending an inform message. He also includes two communicative acts from the commissive category, absent in both FIPA ACL and KQML: *commit* and *refrain*. The preconditions to send a *commit* message state that the agent knows that it will perform α before ϕ is true. $\alpha W\phi$ means α is true until ϕ becomes true.

$$do_i(commit_{i,j}(\alpha, \phi)) \rightarrow K_i(\neg(\neg do_i(\alpha)W\phi))$$

However, by attempting to enable verification, Wooldridge reduces agent communication to a non-intentional or low-level communication language.

3.2.2 KQML

One of the first ACLs was developed within the Knowledge Sharing Effort project (KSE) that Finin and Weber (1993) initiated in the early 90s. Its general purpose was to design techniques, methodologies and software tools for *knowledge sharing and knowledge reuse*. Knowledge sharing requires communication, which in turn, requires a common language. The 'common language' consists of three different components:

- Knowledge Interchange Format (KIF): The content language.

- Ontolingua: A language to define the lexicon, that is, sharable ontologies.
- KQML: A high-level interaction language (Knowledge Query Manipulation Language).

KIF is based on first-order logic and its implementation is independent from the ACL semantics. An ontology is a common lexicon to describe a subject domain. In particular, Ontolingua is a language for building, publishing and sharing ontologies. Finally, KQML represents the set of speech acts that agents can use to interact with each other.

KQML (Finin et al., 1994) is an extensible language designed to facilitate cooperation and interaction among agents. The language is described as a standard for agent communication (Labrou and Finin, 1994a), consisting of a set of performatives whose meaning is independent of their propositional content. KQML is also independent of the transport mechanism, e.g., HTTP, CORBA, email, TCP/IP, etc, and of other high-level protocols such as auctions and negotiation protocols.

The performatives defined in KQML are all assertives and directives. KQML establishes some requirements that are part of its model: Symbolic names to identify other agents, a facilitator agent (facilitator coordinates the interactions of other agents), the ability to send messages to access the knowledge of the agent, and the ability to interact asynchronously with more than one agent at the same time. One of the main criticism against KQML was its lack of semantics, which motivated two mentalistic semantic specifications by Cohen and Levesque (1997) and by Labrou and Finin (1994b). Section 3.2.2.1 describes Cohen and Levesque's specification based on their concept of *intention* as choice plus commitment, whereas Labrou and Finin's proposal, based on a cognitive approach which uses planning operators *à la STRIPS*, is analyzed in section 3.2.2.2.

3.2.2.1 Cohen and Levesque

KQML specifies three different types of performatives:

- **Discourse**: ask-if, ask-all, ask-one, stream-all, eos, tell, untell, deny, insert, uninsert, delete-one, delete-all, undelete, achieve, unachieve, advertise, subscribe.
- **Intervention and Mechanics**: error, sorry, standby, ready, next, rest, discard.

- **Facilitation and Networking**: register, un-register, forward, broadcast, transport-address, recommend-one, recommend-all, broker-one, broker-all, recruit-one, recruit-all.

Cohen and Levesque (1997) present a semantics for KQML based on a logic for intentions within a theory of rational action. They use a multi-modal logic and define primitives like $HAPPENS$ and $P-GOAL$, that is, occurrence and persistent goal respectively. The definition of commitment as persistent goal is as follows (Cohen and Levesque, 1990):

$$\begin{aligned}&(P - GOAL\ x\ p\ q) \equiv\\&(BEL\ x\neg p) \wedge (GOAL\ x\ (LATER\ p))\wedge\\&[KNOW\ x\ (PRIOR\ [(BEL\ x\ p) \vee (BEL\ x\ \Box\neg p)\vee\\&(BEL\ x\ \neg q)]\neg[GOAL\ x\ (LATER\ p)])]\end{aligned}$$

That is, agent x has p as a persistent goal if x has p as a goal and is self-committed toward this goal until x comes to believe that the goal is achieved or unachievable. Literally, x believes p is currently false, x has a goal that p be true later, and x knows that prior to dropping the goal, a belief in p, or the impossibility of p, or a belief in q being false will come before x drops the goal.

Besides, intention is a kind of persistent goal (choice and commitment) in which an agent commits itself to execute an action:

$$\begin{aligned}&(INTEND_1\ x\ a\ q) \equiv\\&(P - GOAL\ x\ [DONE\ x\ (BEL\ x\ (HAPPENS\ a))?; a]q)\end{aligned}$$

Intending to do an action a or achieve a proposition p is a special kind of commitment (i.e., persistent goal) to having done the action a or having achieved p. Literally, agent x intends to do a (escape clause q) if: x has a persistent goal to have done the action a, which occurs after x believes it will happen next. The persistent goal has a escape clause q.

In this approach, *having the intention to do a* commits the agent to reach a state in which it is about to do the intended action. Consequently, an agent cannot commit to do something by accident. For example, an agent cannot "intend for the sun to rise tomorrow"; the specification of such intention does not depend on any action it might do.

Using persistent goals and intentions, Cohen and Levesque (1997) define all communicative acts as attempts. For example, Searle argued that an essential condition of a request is that the speaker *attempts* to get the addressee to perform the requested action:

$$
\begin{aligned}
&[ATTEMPT\ x\ e\ a\ \phi] \equiv \\
&[(GOAL\ x\ (LATER\ a))\wedge \\
&(INTEND_1\ x\ e; \phi? \\
&(GOAL\ x\ (LATER\ a)))]?; e]
\end{aligned}
$$

This definition states that the intended action may succeed and may still fail to achieve the goal, in which case the action does not need to be retried, but if it fails, the agent can try again. Literally, x has a goal that a be true later, and x intends that e will make ϕ true, with the escape clause that x still has a goal that a be true later, and (if false, x can drop the goal) e holds when all of this is true.

The semantics of communicative acts such as request and inform are specified as follows. $REQUEST\ a$ means that an agent requests an action to be performed for some value of an event e:

$$
\begin{aligned}
&REQUEST\ spkr\ addr\ e\ a \equiv \\
&[ATTEMPT\ spkr\ e\ \exists e'\ (DONE\ addr\ a) \\
&[BMB\ addr\ spkr \\
&(GOAL\ spkr\ \exists e' \\
&[\Diamond(DONE\ addr\ a) \wedge \\
&(INTEND_1\ addr\ \alpha \\
&(GOAL\ spkr \\
&[\Diamond(DONE\ addr\ a) \wedge (HELPFUL\ addr\ spkr)]))])]]
\end{aligned}
$$

That is, event e is a request if it is an attempt at that time to get the addressee to do a, while being committed to making public that the speaker wants: 1) that a is done, and 2) that the addressed part should intend to achieve it relative to the speaker's wanting it and relative to the addressee's being helpfully disposed towards the speaker. So the meaning of the performative request is an attempt to get the addressee to respond with a message informing that p is true or p is false. BMB is the operator for mutual belief.

Regarding inform, Cohen and Levesque argue that the communicative act of informing can be defined as an attempt to get the addressee to know that some proposition is true:

$INFORM$ *spkr addr e p* $\equiv$
[$ATTEMPT$ *spkr addr e*
($KNOW$ *addr p*)
[BMB *addr spkr*
($P - GOAL$*spkr*
($KNOW$ *addr* ($KNOW$ *spkr p*)))]]

In other words, *inform* is defined as an attempt in which the speaker is committed to make public that it is committed to the addressee's knowing that it (the speaker) knows *p*.

The semantics of KQML uniforms the context so that agents can "view" each other capabilities. Agents act as if they manage a knowledge base, the Virtual Knowledge Base (VKB). An agent does actually communicate with a VKB since every question, assertion, etc., is always addressed to the agent's VKB. The statements of a VKB are classified into beliefs and goals. Beliefs refer to the information the agent will encode about its external environment and about other agents' VKBs. Goals are states of the world that an agent intends to achieve. The semantic definitions of the performatives refer to these two different mental states of the agent. Representing beliefs and goals is independent of KQML; different languages can be used instead because the performative will just refer to that particular mental state of the agent.

3.2.2.2 Labrou and Finin

Labrou and Finin (1994b) proposed a semantic specification using preconditions, post-conditions and completions based on Speech Acts theory. *Preconditions* indicate the necessary mental state to send a performative and for the receiver to accept it and process it. If these preconditions do not hold, then an *error* or a *sorry* message is generated. *Postconditions* describe the states of both interlocutors after the message has been uttered and after receiving and processing (but before replying) the message. Postconditions hold unless an error or sorry message is sent. The completion conditions define the final state, which usually corresponds to the fulfilment of the initial communicative intention. The general idea is that the *preconditions* describe what can be assumed to be the state of the agents involved in the conversation. Similarly, the *postconditions* describe the states of the agents assuming the successful performance of the communicative act. Thus, the semantics of a performative such as *ask-if* is as follows:

- $ASK - IF(A, B, x)$:
  1. A wants to know what B believes regarding the truth status of the content x.
  2. $WANT(A, KNOW(A, S))$ where S may be any of $BEL(B, x)$, or $\neg(BEL(B, x))$.
  3. **Pre(A)**: $WANT(A, KNOW(A, S)) \wedge$
     $KNOW(A, INT(B, PROC(B, M)))$ where M is **ask-if**(A, B, x)
     Pre(B): $INT(B, PROC(B, M))$
  4. **Post(A)**: $INT(A, KNOW(A, S))$
     Post(B): $KNOW(B, WANT(A, KNOW(A, S)))$
  5. **Completion**: $KNOW(A, S')$ where S' is either $\neg(BEL(B, x))$ or $BEL(B, x)$, but not necessarily the same instantiation of S that appears in $Post(A)$, for example.
  6. Not believing something is not necessarily the same as believing its negation (assuming the language of B provides logical negation), although this may be the case for certain systems. The **Pre(A)** and **Pre(B)** suggest that a proper advertisement is needed to establish them.

In this account, a KQML message represents a single speech act with its associated semantics and protocol, and a list of attribute/value pairs. The meaning of the ask-if parameters is specified in Table 3.3.

Keyword	Meaning
:content	the information about which the performative expresses an attitude
:language	the name of the representation language of the :content
:ontology	the name of the ontology assumed in the content parameter
:reply-with	the expected label in a *response* to the current message
:sender	the actual sender of the performative
:receiver	the actual receiver of the performative

Table 3.3. Meaning of the ask-if parameters.

The meaning of *ask-if* is that the sender wants to know whether the sentence is in the receiver's virtual knowledge base (VKB). Note that sincerity is imposed on agents.

3.2.3 Procedural Semantics

So far, the semantics for ACLs reviewed is based on the mentalistic approach to communication. In fact, this view has been the most popular to develop agent communication languages. However, there are at least two more significant views to specify the semantics of ACLs: The procedural and the social approaches.

Procedural semantics for ACLs have been proposed by Bradshaw et al. (1997), Greaves et al. (2000) and Akkermans et al. (1998) among others. In this approach, agents interchange procedural directives where a message determines the subsequent response available to the addressee. Patterns of conversations are identified (negotiation, auctions, etc.) and protocols are defined to determine the use of the message in those scenarios. The procedural meaning of the speech acts constrains the type and number of responses that are allowed. The interaction protocols are structured in *conversation policies.*

The meaning of the performative will specify the communicative behaviour of the agent. For example, Pitt and Mamdani (1999) give the meaning of a speech act as the intention to reply. A reply function provides the set of speech acts from which the reply will be selected.

$$sa =< s, perf(r, (C, L, O, p, i, ts)) >$$

The performative of a speech act is given by a function f, such that a function $f(sa)$ returns $perf$ in the above example. In order to do this, the function $f(sa)$ is required to be a performative used in the reply and an element of the set of possible replies given by the function. Thus, the semantics of a speech act is as follows:

$$\| < s, perf(r, (C, L, O, p, i, ts)) > \| = L_r < r, sa > \text{ such that } f(sa) \in reply(perf, p, conv_r(i))$$

A performative, a protocol state and a protocol name are the input for the reply function. The input of the function $conv_r(i)$ is the conversation's identifier i and it gives the current protocol state of the conversation. The protocols are defined in terms of finite state diagrams, and the reply function provides the set of available performatives.

3.2.4 Social Semantics

In the specification of a social semantics for ACLs (Singh, 2000; Fornara and Colombetti, 2004), the satisfiability preconditions of speech acts are based on the commitments of the speaker. A commitment guarantees that the addressee will trust the speaker. If an assertive act is performed, the speaker commits itself to the truth of the propositional content. Singh's semantics is based on three validity assumptions which are given using Computation Tree Logic (CTL) extended with modalities for commitments, beliefs and intentions.

- Subjective: the sender is committed to be sincere.
- Objective: the sender is committed to send something it is true.
- Practical: the sender is justified to send the message.

Singh gives a formal social semantics to six types of speech acts: assertives, directives, commissives, permissives, prohibitives and declaratives. The semantics for each speech act are given in terms of the three validity assumptions. The meaning of each communicative action is public since it expresses the commitment of the speaker to a group. Thus, for an *inform* message, the sender commits that it believes its content (subjective) and that the message is true (objective). The practical aspect states that the sender commits that it has a reason to know the content. Singh inaugurated a recent trend of commitment-based approaches. For example, Fornara and Colombetti (2004) specify the speech acts using a operational semantics within an institutional-based approach. This approach restrict the analysis of every speech act to the expression of commitments.

3.2.5 Discussion

This section discusses some problems of the mentalistic, procedural and social semantic approaches to ACLs reviewed above. We first (3.2.5.1) discuss the mentalistic approaches FIPA ACL and KQML. In section 3.2.5.2 ,we shall show several shortcomings of both the social and procedural ACLs.

3.2.5.1 Mentalistic Semantics

We have reviewed three different mentalistic proposals: Regarding KQML, Labrou and Finin's (1994a) semantics, and the intentional approach from Cohen and Levesque (1997); FIPA SL is the specification language for the FIPA ACL semantics.

First, it has been argued (Cohen and Levesque, 1997) that Labrou and Finin (1994a) do not provide a formal semantic analysis of the operators used to define the meaning of the performatives. Second, the meaning of the standard performatives is unclear. For example, the definition of *deny* says that "the embedded performative does not apply to the speaker"; so by using *denying tell*, agents would deny the performative rather than their beliefs. If this is the case, how can it be explained that the performatives perform actions by virtue of being sent?

Third, there are misidentified performatives. There are performatives, such as *achieve*, that should be an argument of a directive performative (e.g. a *request*). Fourth, some classes, such as the class of commissives, are missing. That means that agents are not able to execute essential tasks such as making a *request* or committing themselves to execute an action.

Regarding FIPA ACL, the requirements of its mentalistic specification are too restrictive. There are scenarios such as e-commerce and adversarial negotiation, where is not possible to use these ACLs because agents are required to be *sincere*. In open MAS, agents may not rely in trusting other agents in order to achieve their goals. ACL semantics should be flexible enough to be used in all situations but the conditions specified are so abstract that they are not adequate to guarantee interoperability. Moreover, FIPA ACL semantics specifies only the propositional attitudes of the sender; they do not offer any explanation on how the interpretation process of the receiver works, and they do not account for the perlocutionary effects of the speech acts. Summarizing, mentalistic approaches do not comply with the requirements for ACLs specified in section 1.4.

- Autonomous: Agents' autonomy is limited. For example, agents are required to be sincere.
- Complete: Agents cannot perform basic actions such as committing. The set of communicative actions is not complete.
- Contextual: The context of mentalistic ACLs is defined by the sender.
- Formal: Labrou and Finin do not offer a formal definition of the language used to specify the meaning of the communicative actions. FIPA ACL refers to the author of FIPA SL but does not include its semantics in the specification.
- Grounded: Mentalistic ACLs are not grounded on a computational model, which makes it difficult to verify that the use of an ACL complies with the semantic specification.
- Public: Communication in mentalistic semantics is not public, because it is based on the private internal states of agents. This impedes to use the language in different contexts. Ideally, the meaning of a speech act should depend on the context in which is performed.
- Perlocutionary: Perlocutionary effects are not guaranteed by the semantic specification.

This list builds on the set of requirements we listed in section 1.4. The requirements are: Autonomous, Complete, Contextual, Declarative, Formal, Grounded, Public and Perlocutionary. Table 3.4 shows how mentalistic approaches perform regarding those requirements.

Requirements	**ACL Semantics**		
	FIPA SL	Labrou-Finin	Cohen-Levesque
Autonomous	?	?	✓
Complete	-	-	✓
Contextual	-	-	-
Declarative	✓	✓	✓
Formal	✓	-	✓
Grounded	-	-	-
Public	-	-	-
Perlocutionary	-	-	-

Table 3.4. Mentalistic Semantics and ACL Requirements.

Next section analyzes the properties of procedural and social ACL semantics in terms of the requirements listed in section 1.4.

3.2.5.2 Procedural and Social Semantics

The procedural approach to agent communication reviewed in section 3.2.3 is not high-level enough. That is, it is not appropriate for autonomous and rational computational entities. This approach is valid for reactive agents, but a more complex semantics is needed to allow agents to cooperate, collaborate, etc. The autonomy of agents is too constrained because the decision of how to answer a message is predetermined by conversational templates. Communication becomes a fixed exchange of meaningless tokens. This also affects heterogeneity, since ACLs become redundant under these theories.

Requirements	**ACL semantics**		
	Procedural	Singh (2000)	Fornara and Colombetti (2004)
Autonomous	-	✓	✓
Complete	✓	✓	✓
Contextual	-	-	-
Declarative	-	✓	-
Formal	✓	✓	-
Grounded	✓	?	✓
Public	-	✓	✓
Perlocutionary	-	-	-

Table 3.5. Procedural and Social Semantics.

In table 3.5 we can see that the procedural semantics: (i) constrains too much the autonomy of agents; (ii) defines a complete set of communicative actions, that is, agents can perform at least the relevant type of communicative actions defined by the theory of Speech Acts: assertive, directive and commissive; (iii) it takes into account the context to define the meaning of the messages; (iv) the semantics is not declarative; procedural semantics does not provide the illocutionary and perlocutionary aspects of communication, but states instead the order in which the messages can be sent. This causes agent communication to be meaningless; (v) they do not provide a formal definition of the language being used to specify the semantics; (vi) although the semantics is based on a computational model (finite state systems, for example), the meaning of the performatives is not grounded into that computational model; (vii) the meaning of a message is public, because it is defined in terms of its available responses; (viii) in most cases, there is not a clear semantics defining the perlocutionary effects. When this is not the case, the perlocutionary effects are achieved at the cost eliminating agents' autonomy.

Regarding the social semantics proposed by Singh (2000) and Fornara and Colombetti (2004), a more detailed characterization of concepts such as convention, power, obligation, commitment is needed. These concepts are implicitly included in the agents' behaviour, and therefore different systems present incompatibilities. Social semantics defines every communicative act in terms of commitments. For example, the definition of *request* states that "the sender commits that the receiver has committed to accepting a request from him." Thus, in order to be able to make a *request*, the receiver has to previously agree with the sender that it will accept the *request*. Besides, this approach does not account for the perlocutionary effects. In Table 3.5 is shown that: (i) Social semantics provides a flexible specification that does not constrain too much agents' behaviour; (ii) they provide a complete set of communicative actions; (iii) they do not take into account the context to define the meaning of the performative; (iv) social semantics is declarative, and it offers a high level characterization of meaning for communicative acts; this is not the case in Fornara and Colombetti (2004) though, where they present an operational semantics for a set of communicative acts; (v) they formally define the language to be used to specify the semantics; (vi) the temporal operators can be grounded in a computational model, but it is not shown how beliefs and intentions are; (vii) the part of the meaning of the communicative actions which depend on commitments is public, but that is not the case with the notion of intention and belief defined by Singh

(2000), which depends on agents' private mental states; and (viii), ACLs based on a social approach do not offer a procedure to for the achievement of the perlocutionary effects of performing a communicative action.

The social semantics approach looks the most promising one. However, several requirements remain unfulfilled. These requirements should ensure that the ACL can be applied in different scenarios by taking into account the context. This point is further discussed in the following section.

3.3 Conversation Analysis in Agent Communication

After discussing the semantics of ACLs, in this section we review several accounts to regulate conversations in agent communication. Conversation analysis is crucial to agent communication because it structures the sequences in which messages can be organized. We argue that most of the problems of ACL semantics listed in the previous section are caused by the lack of a pragmatic theory to guide the use of the communicative actions (semantics). Most the ACLs defined to date have been defined its semantics as a stand-alone component. We claim that the full meaning of a communicative act should not be encoded by the semantics alone, not if we want to comply with the requirements that are being discussed anyway, and that every speech act occurs in a context in which it acquires its full meaning. That extra information provided by the context is what ACL pragmatics should (also) be about.

For communication to be successful, agents must be able to select and generate ACL messages that best achieve their goals. At the same time, the receiver has to interpret correctly the message so it can choose the adequate response. Moreover, interaction protocols should specify how to initiate and terminate conversations. In this sense, a remarkable consequence of ACL pragmatics is that taking into consideration more contextual aspects does actually help to reduce the search space of possible responses.

Another positive consequence of defining a pragmatic theory for ACLs is that ACL pragmatics would facilitate the applicability of the ACL because they provide interaction patterns (as the social approach to conversation of Sacks (1972) of section 2.3) which contextually modify and regulate agents' behaviour. Section 3.3.1, look at the interaction protocols library specified for the FIPA ACL. Then, in section 3.3.2 Kumar et al. (2002) propose conversation protocols based on joint intentions. Section 3.3.3 uses a version of Grice's cooperative principle (see 2.2) to guide the interpretation process. Sections 3.3.4 and 3.3.5

analyze two procedural proposals based on Coloured Petri Nets and Finite State Machines. Section 3.3.6 defines protocols using social commitments. Finally, in section 3.3.7 we shall discuss the problems of the proposals reviewed and state the requirements for an ACL for open multi-agent systems.

3.3.1 FIPA IPL

The patterns of conversation specified under the FIPA interaction protocol library (IPL), uses an extension of UML sequence diagrams to model conversations (see Figure 3.1).

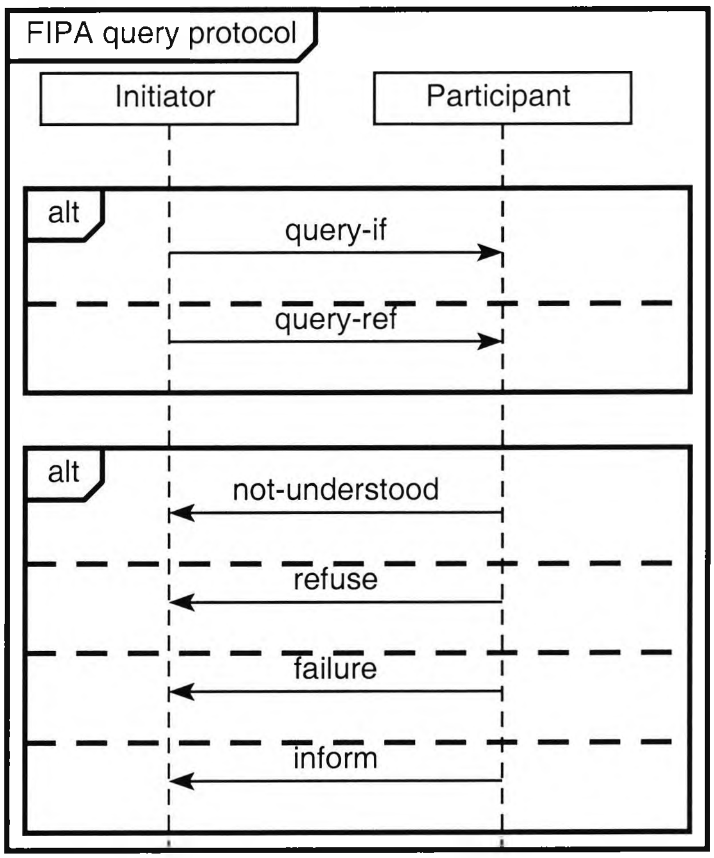

Fig. 3.1. FIPA Query Interaction Protocol

UML sequence diagrams is a technique to model interaction patterns between instances of a system. They model the flow of logic within a system in a visual manner, and are commonly used for both analysis and design purposes. Sequence diagrams are the most popular UML artifact for dynamic modelling, which focuses on identifying the behaviour within the system.

The interaction protocols of the FIPA IPL allow ambiguous interpretations. For example, in the English Auction interaction protocol, the initial call for proposals is sent to all bidders (though this is not explicit in the diagram). The acceptance of the proposal is presumably sent only to one bidder, although again, it is not clearly shown by the diagram. This suggests that an explicit representation of the roles of agents in the conversation is needed.

3.3.2 Joint Intentions

Kumar et al. (2002) developed a formalism to specify conversation policies based on joint intentions. Conversation policies are understood as joint actions caused by joint intentions. Their conception of policies is goal-based; policies are used to achieve certain tasks. The originality of the proposal comes from the identification of state of affairs (landmarks) that must be brought when executing a policy in order to achieve its goal. Families of conversation policies are formally defined as partially ordered landmarks. Landmarks are characterized by propositions that are true in the state represented by that landmark. Then, policies are treated as joint action expressions and joint intention theory is used to define the conversation policies.

The logic of joint intention to specify the conversation policies builds on Cohen and Levesque (1991). Kumar et al. (2002) proposal is strongly based on the semantic specification for KQML of Cohen and Levesque (1997) reviewed in section 3.2.2. The specification language is based on mentalistic concepts like *joint intention* and *mutual belief* and it is not grounded on a computational model. In short, it inherits many of the problems of the mentalistic semantics for ACLs (see section 3.2.5).

3.3.3 Cooperative Principle

Holmback et al. (1999) define a Strengthened Agent Cooperative Principle (SACP) to govern agent communication, which is based on Grice's cooperative principle (see section 2.2). The principle reads as follows:

> "An agent should use a speech act which has the strongest conditions of satisfaction consistent with its intended communicative purpose at that stage of the communication." (Holmback et al., 1999, p.13)

That is, the receiver would assume that the sender has used the performative that best expresses her intended speech act. This assumption would guide further conversation.

Leaving aside more serious issues such as the difficulty to specify policies following such principle, the SACP negatively affects a main feature of ACLs: ACLs define their primitives independently of the content language. However, as it is pointed out in Agerri and Alonso (2002a), the SACP requires compatibility between the performative and the propositional content. Consequently, by limiting the extensibility of the language, it negatively affects agents' heterogeneity.

3.3.4 Coloured Petri Nets

Cost et al. (1999b) have applied Coloured Petri Nets for conversation protocols in KQML. A Petri net is a graphical and mathematical modelling tool. It consists of places, transitions, and arcs that connect them. Input arcs connect places with transitions, while output arcs start at a transition and end at a place. Coloured nets are extended Petri nets in which tokens are differentiated by colours. Transition eligibility depends then on the availability of an appropriately coloured token in all the input places of this transition. Similarly, the output of a transition is not just a token but a specifically coloured token.

Petri Nets conversation protocols view ACL pragmatics from a procedural point of view. As we argued in section 3.2.5 procedural approaches constrain too much agents' autonomy. Since the meaning of the messages are limited to the order in which they can be performed, agent communication is reduced to a meaningless exchange of ordered tokens.

3.3.5 KAoS

This procedural approach defines ACLs in terms of message sequences (Greaves et al., 2000). The Knowledgeable Agent-oriented System (KAoS) architecture (Bradshaw et al., 1997) define conversation policies as finite state diagrams, labels and numbers on each arrow denoting the sender and receiver for a message to cause that state transition (see Figure 3.2).

In this proposal, the formal semantics of policies are given using joint intention theory. Operators for mutual belief (MB), attempt (ATT), request (REQ), weak mutual goal (WMG), joint persistent goal (JPG) and persistent weak achievement goal (PWAG) are defined. KAoS policies are based on finite state

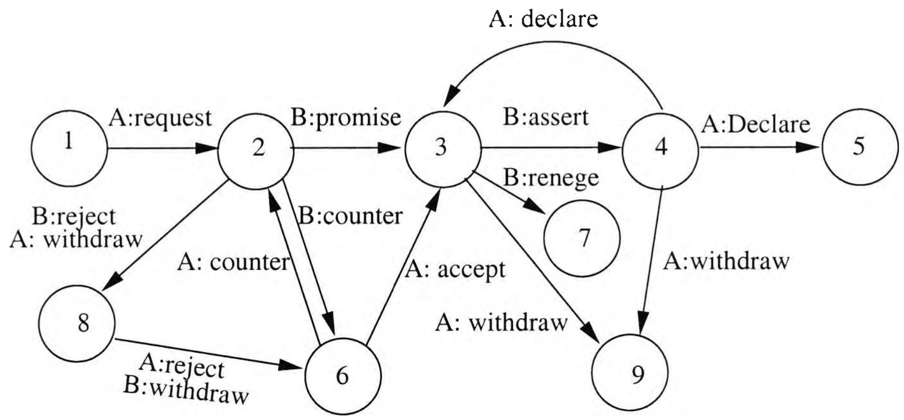

Fig. 3.2. Conversation for Action (Winograd and Flores, 1987)

diagrams with speech acts. Thus, the meaning of the performative is precisely defined using propositional attitudes that the sender must hold. Finite state diagrams have also been used for KQML protocol specification and applied to complex protocols such as English auctions.

The first drawback of using finite state diagrams to represent all possible states reached in a conversation is that it quickly becomes impractical. Even for simple protocols, trying to draw all transitions of all agents involved seems to be unrealistic. Some of the policies include redundancies in their representation. That is because every time a transition takes place, a new state has to be created, although many transitions of the previous state are still valid in the new one. In a complex negotiation protocol many states will have to be created and many of them will have the same transitions. Every state will have a transition for A to ask a question, B to ask a question, A to withdraw, B to withdraw, etc.

3.3.6 Social Commitments

Flores and Kremer (2002) design an ACL with its semantics defined in terms of social commitments to facilitate their verification. Interestingly, they offer a distinction between conversation policies and conversation protocols. Protocols are seen as mere sequences of performatives while policies are understood as inference rules. It is also claimed that policies offer more flexibility since they do not fully predetermine the communicative behaviour of agents. A basic set of speech acts to negotiate social commitments and to specify a number of policies to guide the interaction is offered. The Z notation is used to specify conversations.

Pure social approaches do not take into account the crucial role of the motivational aspects of communication, because they reject the use of goals and intentions to specify ACLs. However, although social aspects of communication must be taken into account, communication still remains a goal-based activity: the main reason of agents to communicate is to achieve a particular goal or is motivated by a communicative intention.

3.3.7 Discussion

In section 3.2.5 we have introduced a set of requirements that both the semantics and pragmatics of ACLs should try to comply with in order to be used in open environments. Some of them were proposed by Singh (2000) and by Mayfield et al. (1996). In the remainder of the thesis, the ACL developed will aim to comply with the following requirements:

1. Autonomous: Our ACL respects agents' autonomy.
2. Complete: It provides a complete set of communicative actions identified by the theory of speech acts (this point only applies to the ACL semantics).
3. Contextual: The meaning of the communicative actions is affected by the context.
4. Declarative: The semantics describes what a message means rather than how to use it.
5. Formal: For the sake of clarity, the specification should be formal to maintain the heterogeneity in its eventual implementations.
6. Grounded: By grounding the semantics in a computational model we can relate the meaning of the properties of the agents to program properties.
7. Public: The semantics should be public if we want to be able to monitoring agents' behaviour. Thus, it should be possible to determine whether agents act following the semantics.
8. Perlocutionary: A unified ACL ensures that the semantics defines the meaning of the messages, which will be affected by the different scenarios in which the ACL may be used. ACL pragmatics is concerned with the specification of rules to regulate the use of the messages to facilitate the achievement of the perlocutionary effects.

Table 3.7 shows the performance of the pragmatic approaches to agent communication considered in this section with respect to the features listed above. We do not discuss whether they present a complete set of speech acts for the obvious reason that it is an exclusive feature of the ACL semantics.

Requirements	Protocols					
	FIPA IPL	Joint Int	Coop Pr.	Petri N.	KAoS	Soc. Commit
Autonomous	-	✓	✓	-	-	✓
Contextual	-	-	✓	-	-	-
Declarative	-	✓	-	-	-	-
Formal	-	✓	-	✓	✓	✓
Grounded	-	-	-	✓	✓	✓
Public	✓	-	-	✓	✓	✓
Perlocutionary	-	-	-	-	-	-

Table 3.6. Requirements for ACL Pragmatics.

Many of the social semantic proposals have merged the semantic aspects with the social ones (i.e, obligations, commitments), with the purpose of making agents cooperate. As it will be explained in the next section, this is the source of some of the problems of current ACLs. Communication is necessary to coordinate cooperation, but that does not mean that communication is needed for agents to cooperate. The semantics of an ACL should concern only on the definition of the meaning of a set of communicative actions when performed in a particular context. We argue that the social aspects must be confined to a separate but complementary level to the semantics, that of interaction protocols and conversation policies (pragmatics). Besides, commitment-based approaches fail to capture the intentional aspect of agent communication, in which agents perform speech acts in order to achieve a particular goal.

A common feature of the conversation protocols proposed for agent communication is the lack of attached semantics. Usually, produces languages in which the only meaning of the messages is the order in which they have to be sent. The motivation for developing ACLs is to support agent conversations in multi-agent systems. To guarantee interoperability, two basic requirements should be met: The meaning of communicative actions must be public and the sequencing of messages must be guided by conversation principles which do not constrain the autonomy and heterogeneity of agents. This is our challenge.

4

A Unified ACL

The previous chapter looked at the semantic and pragmatic (interaction protocols) specifications of various ACLs. We discussed the problems associated with those specifications and listed a number of desiderata for ACLs. In order to comply with those requirements, this chapter presents a unified agent communication language which takes into account both the intentional and social aspects of agent communication. The semantics of ACLs consists of a set of communicative acts whereas the pragmatics is a set of conversation policies and interaction protocols which regulate the use of the communicative actions to facilitate the achievement of their perlocutionary effects. The semantics and pragmatics of agent communication have traditionally been addressed separately. In this chapter we propose and show how to integrate both aspects of agent communication in the same framework. Besides, in our approach the ACL pragmatics is no longer limited to protocols that specify the order in which messages are to be performed. Additionally, it enriches the semantic meaning according to contextual information. In this sense, our approach to ACLs is *pragmatic* because we believe that the communicative meaning cannot be specified by means of the semantics only, but it needs the pragmatics to provide some vital information about agents' roles, conventions, background knowledge, etc.

In order to fully comply with the requirements we established in the previous chapter, we need to ground our ACL in a computational model. Thus, the semantic and pragmatic specification languages ($MLTL_I$ and $NLTL_I$) are defined upon that computational model.

More specifically, section 4.1 argues for the need of a pragmatic theory with an attached semantics for agent communication languages or, in other words, a unified ACL. We then present in section 4.2 the four components of our unified ACL. The following section (4.3) discusses why grounding our theory in a computational model is important, and section 4.4 defines a computational model of which the unified ACL is a component. Finally, we present a general ACL framework defined in terms of the computational model (section 4.5) and some related issues are discussed (section 4.6).

4.1 Pragmatics and Agent Communication

As in Semantics and Pragmatics of natural language, we believe that semantics and pragmatics are complementary disciplines in the study of agent communication. On the one hand, *Semantics* is a discipline which studies the meaning of linguistic expressions. In particular, it studies the meaning of linguistic expressions without consideration of the effect that pragmatic factors, such as features of the context, conventions of language use, and the goals of the speaker, have on the meaning of language in use. On the other hand, *Pragmatics* is defined as the the study of those aspects of meaning and language use that are dependent on the sender, the receiver and other features of the context of utterance, such as the following:

- The effect that the following have on the sender's choice of expression and the receiver's interpretation of an utterance:
  - Context of utterance.
  - Generally observed principles of communication.
  - The goals of the sender.
  - The treatment of given versus new information.
  - The relations of meaning or function between portions of discourse or turns of conversation.

The last point of the list refers to what is known as 'conversation analysis', which is the study of the relations of utterances to other parts of the conversation, especially with a view to determining the following:

- Participants' methods of
  - Turn-taking.
  - Constructing sequences of messages across turns.
  - How conversation works in different conventional settings.

Following Sack's ideas on the study of communication (see section 2.3), agent conversation analysis is concerned with the study of linguistic interaction in conventional settings such as English and Dutch auctions, Contract-Net protocol, and those scenarios in which the role of the agents influence their communicative behaviour. Thus, our theory of ACL pragmatics should account for the achievements of the sender's goals, the fulfilment of the speech act (including the perlocutionary effects), it should also contextually enrich the standard meaning of the messages so that our ACL can be used in different scenarios, and it should co-ordinate turn-taking when agents interact. Therefore, ACLs should capture both the cognitive and social aspects of communication. Next section elaborates on this two-fold character of agent communication.

4.1.1 The Semantic and Pragmatic Interface

The semantics of an ACL defines a specification for each communicative act which must be satisfied by the system of agents using that language. It is important to note that there is a distinction between program semantics for communication statements in an agent's program and ACL semantics, which are specifications for communicative acts. That agents are ACL compliant means that their program semantics will satisfy the semantics defined by the ACL specification.

Agents operate in particular contexts, which include their private mental states and the public observable state of the society. As we argued in the previous chapter, ACLs based on mental states usually specify semantics in terms of preconditions and postconditions which must be true before or after the communicative action is performed. On the other hand, ACLs based on social aspects specify social facts created or modified by the performance of a communicative action. This allows to verify whether agents act according to their specification. For example, the specification for the semantics of communicative actions may refer to future actions and it is necessary to verify that agents will execute them. Typically, in a promise an agent holds an intention as a precondition of sending the *promise* message.

The perlocutionary (postconditions) effects of the communicative actions have proved difficult to specify (Dastani et al., 2003). Traditionally, the semantics of agent communication languages has been enriched to account for several of these pragmatic issues, such as the fulfilment of illocutionary and perlocutionary acts, the relations of meaning between the different parts of the discourse, and the context of the utterance. Thus, the study of conversation in agent communication has been limited to establish the order in which the speech acts can be performed. The previous chapter showed that if the ACL semantics is specified to guarantee the perlocutionary effects, then agents' behaviour is constrained too much. However, leaving the fulfilment of the perlocutionary effects up to the receiver endanger the success of the communicative interchange. Therefore, a crucial question remains to be answered: How can we reach an equilibrium between (agents') autonomy and (communicative) efficiency?

Our unified ACL is composed by a semantic and a pragmatic theories in which the pragmatics of the language contextually enriches the minimal meaning of the speech acts to facilitate the achievement of the perlocutionary effects. Next section formally defines a unified ACL following this idea. In the remaining sections of this chapter, a computational model is formally introduced and we define our ACL as one of its components. The remaining chapters of the thesis use the framework defined here: In chapter 5 we formally specify the semantics of our unified ACL, and a complete Speech Acts Library (SAL) is provided. Then, chapter 6 introduces the Normative Pragmatics (NPRAG) theory, and we show how our proposal achieves the requirements for ACLs discussed so far.

We will try to show how the semantics specified in chapter 5 respect agents' autonomy, are complete, declarative, formal and meaningful. For the ACL semantics to be used in different scenarios, the pragmatic theory defined in chapter 6 contextually enrich the meaning of the communicative actions (semantics), it makes communication public and it facilitates the achievement of the perlocutionary effects.

4.2 A unified ACL framework

The various recent developments in ACL semantics have not been followed by a theory of agent communication pragmatics. The agent communication community has investigated procedural theories rather than theories that regulate the use of the communicative acts depending on their meaning. Paradoxically, this community seems to be more interested in the agents' ability of having conversations according to agents' goals rather than in their capabilities in "following-the-rule" conversation schemes. This contradiction is yet another good reason to work on agent communication pragmatics, in an attempt to bridge the gap between the semantic specification of communicative actions and their use to engage in goal-based interactions.

Traditionally, ACL specifications would just consist of a set of communicative actions and several interaction protocols would then separately define conversational templates for specific scenarios (e.g., auctions). Social protocols include the social component in order to facilitate verification (Wooldridge, 2000b), but fail to capture the general intuition about communication, namely, that agents perform speech acts in order to achieve a certain goal(s). Conversely, our framework places the social aspect of communication at the pragmatic level, maintaining the intentional aspect as a central feature of the semantics (communicative actions). In our approach, ACL pragmatics is represented by conversation policies which take into account the social effects of performing a communicative action and thereby facilitate the achievement of perlocutionary effects. In other words, conversation policies guide and constrain the use of the speech acts. The concept of "right" plays a central role in our definition of conversation policies, allowing us to express the social consequences of performing a particular communicative action. Other approaches (notably Rovatsos et al. (2003)), simply consider the perlocutionary component of communication which is embedded in the semantics of the language (they do not consider a separate pragmatic component). We believe that this approach does not allow to decide whether or not the satisfaction of the perlocutionary action is intentional, which means that it is not possible to assign responsibility, sanctions, etc., in a normative system.

Our resulting unified ACL is composed by a Speech Acts Library (SAL), that is, the semantic meaning of the performatives; a set of conversation policies regulating its use, NPRAG (Normative PRAGmatics), and the specification languages $MLTL_I$ (Motivational Linear Temporal Logic) and $NLTL_I$ (Normative Linear Temporal Logic) for SAL and NPRAG respectively (our framework builds on Wooldridge (2000b)). Note that the interaction protocols would also be defined using NPRAG. In short, the set of communicative acts defined by SAL represents the semantics of the ACL, whereas the set of policies defined by NPRAG represents the normative pragmatics. The semantics captures the intentional character of communication between autonomous agents. The pragmatic level regulates the use of the semantics and facilitates the achievement of perlocutionary effects. Thus, an unified ACL is the tuple:

$$UACL = \langle SAL, MLTL_I, NPRAG, NLTL_I \rangle$$

The next two sections (4.2.1 and 4.2.2) provide a more detailed description of the ACL semantics SAL and pragmatics NPRAG and their respective specification languages $MLTL_I$ and $NLTL_I$. The subscripts of both logics refer to the computational model in which they are grounded. "I" refers to 'interpreted' from the Interpreted Systems model (Fagin et al., 1995).

4.2.1 Speech Acts Library

As in FIPA ACL, the communication language SAL is a STRIPS-like language with preconditions and effects. The Preconditions establish the conditions to be true for the agent to send a message (including the goal the sender intends to achieve by sending that message). On the other hand, the Rational Effects state what the sender intends to cause by performing the communicative action. As it has been already discussed, autonomous agents, by definition, cannot be forced to guarantee the effects. The semantics of SAL is given by a function

$$[\![-]\!]_{SAL} : \mathit{wff}(SAL) \rightarrow \mathit{wff}(MLTL_I)$$

A message of SAL, φ, is mapped to a formula $[\![\varphi]\!]_{SAL}$ of $MLTL_I$. This formula defines the semantics of φ, which represents the preconditions to be true for the agent to send a message of SAL. In FIPA ACL, these preconditions are represented by the Feasibility Preconditions (FPs). We do define a new syntax for our communication language SAL. Instead, we use the syntax of FIPA ACL, which allows to identify a set of well-formed formulae of $\mathcal{L}_c$, wff($\mathcal{L}_c$), which correspond to messages, φ. For example, using a FIPA ACL message such as *request*, the resultant picture is as follows:

$$[\![\langle i, request(j,a)\rangle]\!]_c = FP\ (a)\ [i\ j] \wedge B_i Agent\ (j,a) \wedge \neg B_i I_j\ Done(a)$$

One of the first questions relevant to our task is what the nature of the specification language should be. From the several languages that have been proposed to use in specifying multi-agent systems, we choose that of formal logic. Logic is a well developed and stable technique with a long and rich tradition in different disciplines such as philosophy, computer science, economics, linguistics and in its own sake. Furthermore, logic is particularly useful to reason about agents' beliefs, goals, intentions, knowledge, desires, etc. Temporal logic has been successful for the specification and verification of systems (Manna and Pnueli, 1993). In this thesis we will use a multimodal temporal logic as specification language for the ACL semantics and pragmatics.

Thus, the semantics of our specification language $MLTL_I$ mainly consists of an extension of Propositional Linear Temporal Logic (PLTL) with operators for cognitive states such as beliefs, goals and intentions to model the intentional character of communication. However, we do not model them in the traditional way as private mental states of agents. In fact, operators for cognitive states are defined in our framework with respect to the runs related to the beliefs, goals and intentions of agents $(\mathcal{B}_i, \mathcal{G}_i, \mathcal{I}_i)$. Thus, $\mathcal{B}_i$ refers to those points in the system considered possible from the agents' viewpoint. $\mathcal{G}_i$ are runs that the agent wants to hold, and $\mathcal{I}_i$ denote the computing paths with the actions the agent will perform. The interpretation of cognitive states with respect to runs in the system is inspired by the *interpreted systems* approach which we discuss below.

The semantics of $MLTL_I$ will be defined by means of the $\models$ satisfaction defined as

$$\models \subseteq \ wff(MLTL_I) \ \times \ mod(MLTL_I)$$

Informally, the meaning of a formula φ of $MLTL_I$ corresponds to the set of models in which φ is satisfied. The following function expresses that if φ is a *wff* of $MLTL_I$, then $[\![\varphi]\!]_s$ is the set of models in which φ is satisfied.

$$[\![-]\!]_{MLTL_I} : wff(MLTL_I) \rightarrow \wp(mod(MLTL_I))$$

Given that $MLTL_I$ grounds the cognitive states in a computational model, the semantics of the ACL can be given by the models of the system. Most of the ACLs reviewed in the previous chapter do not define their specification languages related to a computational model, which means that it is not possible to verify whether the agents' behaviour conforms to the ACL semantics.

4.2.2 NPRAG

Linear Temporal Logic combined with a deontic operator is used to define the language $NLTL_I$. Using this language we specify several components needed for the design of normative conversation policies and interaction protocols, that is, to specify pragmatic rules to guide agents communicative behaviour. These rules are built according to the meaning of the speech acts they regulate. We define rights, roles and violations to construct the policies. Following the procedure of the previous section, the semantics of the pragmatic language NPRAG is given by the function:

$$[\![-]\!]_{NPRAG} : wff(NPRAG) \rightarrow wff(NLTL_I)$$

The normative policies will be given a semantics in terms of the models where the formulae are satisfied:

$$[\![-]\!]_{NLTL_I} : wff(NLTL_I) \rightarrow \wp(mod(NLTL_I))$$

The definition of the syntax for the conversation policies and protocols is beyond the scope of this thesis. We believe that the role can be filled by a declarative language like Prolog, or any other declarative programming language.

Grounding our specification languages $MLTL_I$ and $NLTL_I$, in a computational model is the first step to be able to verify the compliance of the system to the semantics of the unified ACL. Thus, in the next section we explain in more detail why we think this is so important to any agent theory. In section 4.4 we formally define the components of a multi-agent system. The unified ACL and the multi-agent system (as a set of agents within a organizational structure) form the necessary components for an agent communication framework to comply with the desirable requirements. Finally, the general picture of our agent communication framework is given in section 4.5.

4.3 Computational Models

Traditionally, the role of formal logic in artificial intelligence and distributed computing is to provide clear formal tools to specify complex systems. However, the logic-based specifications have been criticized on the grounds that they do not provide real methodologies for building distributed systems. The situation has been somewhat worsened in the age of the Internet, in which the agent paradigm has grown significantly. In order to cope with the increasing complexity of the capabilities required by agents, researchers have been using complex multimodal logics for their specification which are generally ignored by programmers that do not see a clear relation between the specification in formal logic and computational systems.

Thus, we need new methodologies for the specification of multi-agent systems and agent communication languages which can be taken into account by programmers. Several authors (Fagin et al., 1995; Wooldridge, 2000a) have argued that to bridge the gap between theory and practice, the multimodal logics used in the specification of multi-agent systems must be grounded in a computational model. This is directly related to the satisfaction of the "Grounded" and "Public"requirements for ACLs (see section 1.4). Following this point, we define a computational model upon which our unified ACL specification languages, $MLTL_I$ and $NLTL_I$ are grounded.

Typically, the semantics of the logics used for the specification of multi-agent systems is based on Kripke models or possible world semantics (Kripke, 1963).

Definition 1 (Kripke model).

A Kripke model $M = \langle W, R_1, \ldots, R_n, L \rangle$ *is a tuple composed by a non-empty set of worlds* W*,* n *(one for each agent) accessibility relations* $R_i \subseteq W \times W$ *between possible worlds, and an interpretation* $L : AP \rightarrow 2^W$ *which assigns to every atomic proposition the set of possible worlds where it is true. The basic concept is that the truth of a formula* φ *at a possible world* w *of a Kripke model* $M = \langle W, R_1, \ldots, R_n, \pi \rangle$*, denoted by* $M, x \models \varphi$*, is recursively defined as follows:*

$M, x \models p$ *iff* $x \in AP(p)$

$M, x \not\models \perp$

$M, x \models \varphi_1 \rightarrow \varphi_2$ *if* $M, x \models \varphi_1$ *implies* $M, x \models \varphi_2$

$M, x \models \Box\varphi$ *if* $M, y \models \varphi$ *for every* $y \in W$ *such that* xRy

Although Kripke semantics is conceptually useful to understand many concepts in multi-agent systems, the problem is that there is no clear correspondence between the possible world semantics of Kripke models and a multi-agent system. In particular, it is not very clear what could it be the correspondents of the accessibility relations between possible worlds in computational system. That is why it is important to make available a computationally grounded formalism that aims at clearly relating the semantics descriptions with states in computing systems. In this thesis, we follow the work on *knowledge* by Fagin et al. (1995) to propose a grounded modal logic which models agents' cognitive states as states of a multi-agent system; by doing this, the accessibility relations that characterize the propositional attitudes (beliefs, goals, intentions, etc.) relate system's states. Furthermore, although we do not use possible world semantics to characterize motivations and norms in our model, we still keep most of the machinery of modal logic (such as Kripke frames).

Communicating agents operate in specific contexts which usually consist of agent internal states, public observable states which keep the current state of the system, and the programs of agents. Ideally, an ACL should include information about the system and how agents affect and are affected by it. Thus, if an agent accepts a request, it is understood that the agent's program will, at some point, execute an action towards the satisfaction of that request. In other words, our unified ACL must be built upon a computational model.

Several agent communication languages have been put forward in an attempt to standardize agent communication. However, as it was discussed in sections 3.2 and 3.3, most of them are not grounded in a computational model (Guerin and Pitt, 2002; Wooldridge, 2000b). The semantics given to these ACLs are based on modal logic with possible world semantics. Consequently, it is not possible to verify whether the agents are compliant with the semantics of the ACL (van der Hoek and Wooldridge, 2003).

For our purposes, being computationally grounded means that, for a set of programs, $\Pi = \{\pi_1, \pi_2, \ldots\}$, we can show that one of these programs corresponds to one theory. The theory can be a formula φ of the multimodal language used to specify FIPA ACL, for example. As we argued in the previous section, the semantics of the specification language will be given in terms of the set of models where the formula is satisfied. Both the semantics and pragmatics are specified using $MLTL_I$ and $NLTL_I$ formulae will be used to express and reason about the properties and behaviour of agents in a system. Furthermore, questions can be raised about the cognitive and deontic operators of our specification languages, since traditionally the logics including these kind of operators do not relate to computational models. We solve the issue by formalizing both cognitive and deontic operators à la Fagin et al. (1995).

4.3.1 Basic Properties of the Computational Model

Usually, the properties of computations that are expressed with temporal logics are *safety* properties, *liveness* properties and *fairness* properties (Manna and Pnueli, 1995). Safety properties state what should not happen in the computation. Some examples of safety properties are partial correctness properties of the kind of "If a precondition φ holds at the input of a program, then when it terminates (if it does), a postcondition ψ will hold at the output." They also state that "a resource should never be used by two or more processes simultaneously."

Conversely, *liveness* properties state what is desired to happen in the computation. These can be formulated as "If a precondition φ holds at the input of a program, then it will terminate and a postcondition ψ will hold at the output." These are total correctness properties; for instance, "if a message is sent, it will be delivered."

The purpose of *fairness* properties is trying to guarantee that all processes will be treated fairly. They express important requirements of concurrent systems, that is, systems whereby several processes sharing resources are run concurrently by an operating system which is to schedule their execution in a fair way. A fair schedule would mean that if the process is persistent enough then eventually its request will be granted.

4.3.2 Modelling Multi-Agent Systems

A multi-agent system can be seen as a collection of interacting agents. Part of the knowledge representation literature employs modal languages defined on semantic structures called *Interpreted Systems* (Fagin et al., 1995). The main idea is to describe a system by specifying the states of every agent of the system and of the environment. We choose interpreted systems to be our computational model since it is relatively straightforward to define other logics upon it. This is what we do: We adapt the interpreted systems approach to model the multi-agent system in which the (communicative) interactions take place, and show how this model is suitable to describe the macroscopic properties of agent communication in multi-agent systems.

For our purposes, examples of multi-agent systems are programs or processes which run on the same computing device, message-passing systems, processes in a computer network, etc. Although possible world semantics is not used to describe the macroscopic properties of the system, we can still benefit from most of its technical apparatus. In fact, an interpreted system can be defined using Kripke frames (or structures). The main technical apparatus from which our model is adapted can be found in Fagin et al. (1995).

We have already discussed that our specification languages will include a temporal component. The key intuition behind this is that every agent in the multi-agent system is in some state at any point in time. This state is the agent's local state which consists of all the information about other agents and about the environment to which agents have access. For our purposes the local state could include the messages performed so far in the ongoing conversation, which will in fact affect the future utterances of the agents participating in the conversation.

Fagin et al. (1995) argue that since we see agents as being in some state at any point in time, we can also think of the whole system as being in some state. In this sense, the notion of *environment* refers to everything else in the system that is not an agent. In agent communication, the environment's state would consist of the current messages that are still being sent, or some properties about the social aspects of the system, such as contextual features that define a specific scenario in which conversation is taking place. Both the agent's local state and the environment's state conform the global state of a system. We can preliminary define a global state of a multi-agent system as a tuple $(s_e, s_1, \ldots, s_n)$ where s_e is the environment's state and s_i is the local state of agent i.

Given a set of agents $Ag = (1, \ldots, n)$, consider a set of states for the environment and n sets of local states, one for every agent, of the multi-agent system. We write L_i to refer to the non-empty set of local states for agent i, and L_e denotes the set of possible states for the environment. Thus, elements of L_e are denoted by $l_e, m_e, \ldots$ and members of L_i by $l_i, m_i, \ldots$.

Definition 2 (Global states).

A global state describes the system at a given point in time. A set of global states consists of the set of environment's states and the set of agents' local states: $GS = L_e \times L_1 \times \cdots \times L_n$.

A global state captures the situation of all the agents of the system and of the environment at a particular point of time. An interpreted system of global states is then defined as follows.

Definition 3 (Interpreted System of global states).

An interpreted system of global states is the tuple (GS, L), *where* L *is an interpretation for the atomic propositions and* GS *is a set of global states as defined above.*

We want to integrate time in our model so we can describe how the system evolves over time. In order to do that, we define the notion of *run* (Fagin et al., 1995). A run is a function from time to global states that can be seen as a complete description of what happens over time in one possible execution of the system. We assume that runs are infinite and that time is discrete and ranges over the natural numbers. This means that $r(0)$ refers to the initial global state in a possible run r, where $r(1)$ is the next global state of that run. It is quite common to use the notion of 'run' as a synonym of 'computation'. In our framework the term 'run' refers to the description of computational paths of *global states* (see below). If a run is finite, we call it a *terminating* run. We restrict the use of 'computation' to the object of the description given at some state s by means of an atomic proposition ϕ.

Definition 4 (Run).

A run is a function from the time domain to sets of global states and it gives an infinite sequence of global states $(r(0), r(1), \ldots, r(n))$ *which completely describes a possible execution of the system.*

There can be many runs in multi-agent system, since the system's global states can evolve in many possible ways. In our case, the time is modelled as a sequence of states extending infinitely into the future. This sequence of states is called a computation path and in the Interpreted Systems literature, it is called a run. In general future is not determined, so we consider several computational paths, representing different possible futures. Intuitively, a system consists of a nonempty set of runs. This means that we do not model the system directly, but instead we describe the possible behaviours of the system by means of runs. In a communication system, the runs describe all the possible communicative acts that can be performed by the agents. The notion of point relates runs with time.

Definition 5 (Point).

A point consists of a pair (r, m) *where* r *is a run and* m *is a given time. If* $r(m) = (s_e, s_1, \ldots, s_n)$ *is the global state at point* (r, m)*, then we say that* $r_e(m) = s_e$ *and* $r_i(m) = s_i$ *where* $i = 1, \ldots, n$*. This means that* $r_e(m)$ *and* $r_i(m)$ *are the environment's and agents' local state respectively at the point* (r, m)*.*

A system can be seen as a Kripke structure except that we do not have any labelling or interpretation function to assign truth values to the atomic propositions.

Definition 6 (System (with time)).

A system T *over* GS *is a set of runs over global states in* GS*. A point* (r, m) *is a point in system* T *if the run* r *is in the system* T *such that* $r \in T$*. A round* m *in run* r *takes place between* $m - 1$ *and time* m*.*

Since the temporal component of our model is linear time as defined in the literature (Manna and Pnueli, 1992), we can see the system T as a type of computation tree where the set of runs correspond to the set of all infinite runs of a computation tree. In the next section we combine the concepts defined in this section to structure the computational model in which to ground our agent communication framework.

4.4 A fair Interpreted System

In this section we specify a multi-agent system in terms of Interpreted Systems. The type of multi-agent system defined in this section is the system we will consider when specifying the ACL semantics and pragmatics. Having a system of interacting agents make the system extremely complex to specify. It is usually difficult enough to reason about the actions of one agent, so when we consider a multi-agent system the list of parameters we have to consider (agent's actions, details of the environment, how the environment affect some agents, etc.) can grow *ad infinitum*.

One way of reducing the complexity is to focus only on some specific aspects of the system, those that are relevant for our purposes (e.g., communication). In this sense, the formal model we present here is general enough to capture the crucial features of multi-agent systems. In particular, we show how to adapt the Interpreted Systems structure from Fagin et al. (1995) to model agent communication in multi-agent systems (including the communication history and how to model the organizational structure of the system).

The description of the internals of the agent programs is beyond the purpose of this thesis. We believe that a different variety of programming languages can be used to describe systems in fair interpreted systems. As we argued in the introduction, for completeness reasons we will show how Simple Programming Language (Manna and Pnueli, 1995) can be used to describe agent programs in multi-agent systems.

As we discussed in the previous section, the semantics of the multi-agent system can be seen as Kripke structures. Suppose that we have a set of atomic propositions AP which describe basic facts about the system, such as "agent i sends message p in round 4 of this run". We can define a very simple interpreted system as a structure $\mathcal{I} = \langle T, L \rangle$, where T is a system of runs over sets of global states GS and L is a labelling function which assign truth values to the propositions at the global states. Thus for every $\phi \in AP$ and global state s of T, then we have that $L(s)(\phi) \in \{true, false\}$. Finally, we define a temporal modality $\Box$ (it is true at all later points) for which a dual is defined $\Diamond\phi = \neg\Box\neg\phi$.

Definition 7 (Basic Interpreted System: Syntax).
The syntax is defined as follows:

1. *If ϕ is an atomic proposition of AP then ϕ is a formula of $\mathcal{L}$*
2. *If $\phi \in \mathcal{L}$ then $\neg\phi \in \mathcal{L}$*
3. *If $\phi \in \mathcal{L}$ and $\psi \in \mathcal{L}$ then $(\phi \rightarrow \psi) \in \mathcal{L}$*
4. *If $\phi \in \mathcal{L}$ then $\Box\phi \in \mathcal{L}$*

The models are interpretation systems and we use labelling instead of valuations, as it is traditional in temporal logic. The interpreted system is then associated to a Kripke structure $M_I = (S, L, R_1, \ldots, R_n)$ by taking S to be the global states in I, and we define the relation R_i which corresponds to the weakest normal modal logic. $(I, r, m) \models \phi$ iff $(M_I, (r, m)) \models \phi$ means that a formula ϕ is true at a point (r, m) in an interpreted system I if and only if ϕ is true at a point (r, m) in the associated interpreted Kripke structure M_I.

Definition 8 (Basic Interpreted System: Semantics).
$(I, r, m) \models \phi$ iff $(M_I, (r, m)) \models \phi$ means that a formula ϕ is true at a point (r, m) in an interpreted system I if and only if ϕ is true at a point (r, m) in the associated interpreted Kripke structure M_I.

$(I, r, m) \models \phi$ iff $\phi \in L(s)$
$(I, r, m) \not\models \phi$ iff it is not the case that $(I, r, m) \models \phi$.
$(I, r, m) \models \phi \rightarrow \psi$ if $(I, r, m) \models \phi$ implies $(I, r, m) \models \psi$
$(I, r, m) \models \Box\phi$ if $(I, r, m') \models \phi$ $\forall m' \geq m$

The basic modal logic K provides the axiomatics:

$$K : \Box(\phi \rightarrow \psi) \rightarrow (\Box\phi \rightarrow \Box\psi)$$

K is complemented by the necessitation rule NEC : if $\vdash \phi$ then $\vdash \Box\phi$. This very basic logic constitutes the basis upon which other temporal or multimodal logics, including $MLTL_I$ and $NLTL_I$, are built.

If we assume the $\Box$ operator to mean "in every point in the future" and the $\Diamond$ "at some point in the future", we are able to express in table 4.1 the properties of computations listed in the previous section.

Safety	$\Box\neg(bad)$
Partial Correctness	$\phi \rightarrow \Box(terminal \rightarrow \psi)$
Liveness	$\Diamond(good)$
Total Correctness	$\phi \rightarrow \Diamond(terminal \wedge \psi)$
Weak Fairness	$\Box\alpha \rightarrow \Diamond\beta$
Justice	$\Diamond\Box\alpha \rightarrow \Diamond\beta$
Strong Fairness	$\Box\Diamond\alpha \rightarrow \Diamond\beta$

Table 4.1. Computational Properties in basic interpreted systems.

In verifying systems, we can be interested only in correctness along fair execution sequences. When dealing with communication protocols of any kind, we may wish to restrict the set of computation sequences. In this case, unfair execution sequences are those in which a sender continuously sends messages without being able to reach the receiver. Fairness conditions assert that requests for services are eventually granted.

Starting from the basic system defined in this section, we can now define a multi-agent system modelled as a fair Interpreted System for our agent communication framework.

Definition 9 (Fair Interpreted System).

A multi-agent system is described as a (fair) Interpreted System, $IS = \langle \Sigma, \mathcal{T}, GS_0, L, \mathcal{J}, \mathcal{C} \rangle$.

- $\mathcal{T}$ *is a set of runs over sets of global states* GS. *Each run* $r \in \mathcal{T}$ *is an infinite sequence of global states* $(g_1, g_2, \ldots)$ *where* $g_i \in GS$ *and it describes everything that happens over time in a possible execution of the system.*
- GS_0 *is the initial condition.* GS_0 *characterizes initial states, i.e. if a state* g *of the system satisfies the assertion* GS_0*, then it is a state from which the system can start running.*
- L *is a labelling function* $L(g)(\phi) \in \{true, false\}$ *where* $g \in GS$ *and* $\phi \in AP$.
- $\mathcal{J} \subseteq \mathcal{T}$ *is a set of fair runs. The requirement of justice for* $\tau \in \mathcal{J}$ *disallows a computation in which* τ *is only enabled beyond a certain point but taken only finitely many times. A fair run cannot be continually executed without being taken. Justice allows us to ensure that each parallel process is executed fairly. Without justice one process could be waiting forever while a parallel process continues to execute.*
- *For an alphabet* Σ, *we use* Σ^* *and* Σ^ω *to refer to the sets of all finite and infinite words over* Σ, *respectively. For two words* $\rho_1 \in \Sigma^*$ *and* $\rho_2 \in \Sigma^* \cup \Sigma^\omega$, *we use* $\rho_1 \cdot \rho_2$ *to express concatenation of* ρ_1 *and* ρ_2. *We state that each run* $r = g_0, g_1, g_2, \ldots$ *induces the word* $L(r) = L(g_0) \cdot L(g_1) \cdot L(g_2) \ldots \in \Sigma^\omega$. *To express that a computation is fair, we refer to the set* $Inf(r)$ *of states that* r *visits infinitely often:*

  $$Inf(r) = \{g \in GS : \textit{for infinitely many } i \geq 0, \textit{ we have } g_i = g\}$$

  We have seen three different types of fairness. Thus, the way we refer to $Inf(r)$ *depends upon the fairness condition of* T. *For justice, where* $\tau \subseteq 2^S \times 2^S$, *and* r *is fair iff for every pair* $\langle B, G \rangle \in \tau$, *we have that* $Inf(r) \cap (S\ B) = \emptyset$ *implies* $Inf(r) \cap G = \emptyset$.
- $\mathcal{C} \subseteq \mathcal{T}$ *is a set of strong fairness runs. A strong fairness computation cannot be enabled infinitely many times but taken only a finite number of times. A process could get stuck at a just computation if it is not continuously*

enabled, but only periodically enabled. As in the previous kind of fairness, we can express this as follows:

Where $\tau \subseteq 2^S \times 2^S$, and r is fair iff for every pair $\langle B, G\rangle \in \tau$, we have that $Inf(r) \cap B = \emptyset$ implies $Inf(r) \cap G = \emptyset$.

The fair interpreted system (interpreted system for short) offers a convenient representation of the observable social context which is encoded in the environment's state. In our system IS, a set of global states GS such that $GS = g_1, \ldots, g_n$ represents the idea that, at a any point in time, each of the agents is in some state g. In order to take into account the environment, we distinguish between agent's local state from a global state (Fagin et al., 1995).

Thus, we distinguish agents from their environment. Of course, we can also think of the environment as being another agent (as in chapter 6 when when we define normative systems). Thus, a global state of a system with n agents is defined as an $(n+1)$ tuple of the form $(s_e, s_1, \ldots, s_n)$ where s_e is the state of the environment and s_i is agent i's local state. In the next section we elaborate on the role of global states to define our agent communication framework.

4.4.1 Global states in an Agent Communication Framework

The specific characterization of the environment depends on the system being analyzed. For example, if we are interested in agent communication, we can view a message buffer, in which messages to be delivered are stored, as part of the environment. For our purposes, the environment includes the global states created by interactions and the rules governing their creation. Additionally, the following aspects of the context may affect the meaning of a communicative act:

- The previous communicative acts in the ongoing interaction.
- The scenario in which the interaction takes place.
- The organizational structure in the system, for example, the status and authority of participants.

We treat all these aspects of the context as public knowledge and include them in the *environment*. The environment describes all publicly observable phenomena including propositions representing social facts (for example the rights of an agent), control variables (for example role variables), the rules governing interaction (conversation policies). The environment can be determined by observing communications: We assume that a message being added to an agent's channel can be observed, but a message being removed from the channel cannot.

The specification languages $MLTL_I$ and $NLTL_I$ for the unified ACL semantics and pragmatics are grounded in the interpreted system IS we have defined above. The interpreted system IS and the unified ACL form the *agent communication framework* (ACF). Being able to globally describe the multi-agent system (transition system) allows us to describe three important aspects of agent communication: The communication history, the communication rules and the social facts. These three aspects are described in the next section.

4.4.2 Description of the Social Structure

One of the reasons for using global states in our system is that we can record the history of the current run in the environment's state. Besides, the communication history h_i acts as a conversational record for each agent's local state. The type of each history variable is a list of messages. For some agent i, h_i is a chronologically ordered list of all the messages sent on some agent's input and output channels.

Normative facts are described with the language $NLTL_I$ and includes agent obligations, rights, and role relationships. The type of normative facts variable is a mapping from well formed formulae of the normative facts language $NLTL_I$ to true or false values.

$$nf_i : \mathit{wff}(NLTL_I) \rightarrow Tr$$

where $Tr = \{true, false\}$. For some agent i, nf_i is an interpretation of all the normative fact propositions which arise from the initial facts and the communications the agent i has observed.

Finally, communication rules describe how the environment changes when a communication happens. This includes updating the history and also encodes all the rules governing the creation and modification of normative facts from the point of view of agent i. These rules define how runs modify social facts and are what Searle calls constitutive rules (Searle, 1983). In most real systems there are also non-communicative events (such as moving an object in a world) which do not explicitly communicate any information but may still alter the social facts in a system. Since we are only concerned with communication we assume here that normative facts are solely a function of the communications occurring in the system. The governing rules also capture contextual aspects, for example the aspects specific to the application domain, that is, these conventions will be different in different application domains. The ACL pragmatic component (described in chapter 6) defines the communicative rules.

4.5 An Agent Communication Framework

Now we have available all the components to define our Agent Communication Framework (ACF). The communication framework consists of those necessary elements to model communication in agent communication. It therefore includes a model of the multi-agent system and a definition of the agent communication language.

Definition 10. *The agent communication framework is a tuple*

$$ACF = \langle \mathcal{IS}, UACL \rangle$$

where

- $\mathcal{IS}$ *denotes the interpreted system. The computational model is the basic structure upon which the specification languages* $MLTL_I$ *and* $NLTL_I$ *are defined. It also provides a natural way of describing the norms of the system.*
- $UACL$ *is a unified agent communication language which defines its semantics and pragmatics in the same framework so that the semantic meaning of the speech acts is contextually enriched by the pragmatics. The unified ACL is represented by the tuple*

$$UACL = \langle SAL, MLTL_I, NPRAG, NLTL_I \rangle.$$

In the remaining of this thesis, we focus on the last element of the framework: specifying a unified ACL by means of grounded specification languages and a new normative theory of agent communication pragmatics (NPRAG), which aims to satisfy the requirements of a good ACL for open multi-agent systems.

Describing the internals of the programs of the multi-agent system is not strictly necessary for our purposes, but in doing so we present a more complete picture of the multi-agent system. By describing the macro component of the multi-agent systems, that is, agent's behaviours, is sufficient. We use SPL (Manna and Pnueli, 1995) to describe the semantics of the agent programs so we can show the full picture of the system.

4.5.1 Describing Agent Programs

Manna and Pnueli (1992, 1995) specify the Simple Programming Language to describe programs. We can apply SPL to describe the programs of agents in our multi-agent system. Using SPL, the entire multi-agent system, consisting of two agents, can be described by a program P.

$$P :: [M_1 || M_2]$$

where M_1 and M_2 are seen as modules which represent the programs of two agents. The transitions associated with a program that contains a module statement are obtained by treating the module statements as the processes of the program.

Each module M_i has the following form:

module M_i :: [**module** ; *interface specification*; *body*]

The keyword **module** identifies this as a module statement, *module-declaration* declares variables and communication channels that will be used by the module and the *body* is a statement (the agent's program) that may contain additional local declarations. In our framework, all agent programs are modules. Note that the program P does not need a declaration statement of its own since all declaration statements are moved from a program's heading to the module headings when a program is viewed in this way. The *interface specification* is a list of declaration statements, each of the form

modes $x_1, \ldots, x_m$: type **where** φ

where *modes* is a list of one or more modes, which may be **local**, **in**, **out**, or **external**. The list $x_1, \ldots, x_m$ consists of names of variables or channels that are declared by this statement. *Type* specifies the type of declared variables and *body* is a statement. Statements within the body of module M refer only to variables and channels that are declared within the interface specification of M.

Figure 4.1 shows a simple multi-agent system which consists of only two agents whose programs are described by the modules M_1 and M_2 (Manna and Pnueli, 1995). For simplicity, the messages exchanged by our agents are simple integers. The interface specification of module M_1 declares x and z as writable integer variables, whose initial value is 0. Besides, by not declaring them as external, it forbids these variables to be written by the environment. Variable y is declared as an integer variable that may be read by M_1 and modified externally. Similarly, M_2 identifies x as an integer that is writable by the environment but read-only for M_2. Finally, variable y is identified as writable by M_2 and read-only for the environment. The program starts with the values of x, y, and z set to 0. Module M_1 starts a communication protocol between the two modules by setting x to 1. M_2 answers by setting y to 1, and M_1 answers back by setting z to 1.

$$
M_1 :: \left[\begin{array}{l}
\textbf{module} \\
\textbf{external in} \;\; y \quad : \textbf{integer} \\
\textbf{out} \qquad\qquad x, z : \textbf{integer where } x = 0,\; z = 0 \\
\qquad l_0 : \left[\begin{array}{l} l_1 : x := 1 \\ l_2 : \textbf{await } (y = 1) \\ l_3 : z := 1 \end{array}\right] : \widehat{l_0}
\end{array}\right]
$$

$$\|$$

$$
M_2 :: \left[\begin{array}{l}
\textbf{module} \\
\textbf{external in} \;\; x : \textbf{integer} \\
\textbf{out} \qquad\qquad y : \textbf{integer where } \; y = 0 \\
\qquad m_0 : \left[\begin{array}{l} m_1 : \textbf{await } (x = 1) \\ m_2 : y := 1 \end{array}\right] : \widehat{m_0}
\end{array}\right]
$$

Fig. 4.1. A system of two Agents.

4.6 Discussion

Most of the ACLs proposed do not ground their specification languages in a computational model. Among the mentalistic approaches, the most notorious cases are FIPA ACL and KQML. Since these ACLs do not take into account the social (public) aspects of agent communication it is not possible to check that agents are using the ACL according to its semantics. These problems are basically caused by the absence of a computational model in which to ground the semantics of the ACL.

The procedural approaches (Greaves et al., 2000; Pitt and Mamdani, 1999) often do not provide a declarative and meaningful set of messages, so we cannot consider them to provide an adequate (high-level enough) ACL for open multi-agent systems.

Finally, social approaches, which are traditionally based on the notion of commitment (Singh, 2000; Fornara and Colombetti, 2004; Flores and Kremer, 2002), differ in the treatment of the ACL semantics. All of them limit their pragmatics to a set of interaction protocols which merely dictate the order in which the messages can be uttered. All of them try to encode the social aspects of communication in the semantics, which means that messages are not defined in terms of communicative intentions anymore, but in relation to the commitments that are created by the utterance of an speech act.

Singh offers a semantics in terms of Computation Tree Logic with operators for commitments, beliefs and intentions. Although Singh does not explicitly define a computational model, the use of CTL as specification language implicitly means that the specification language is defined in terms of a transition system. However, the the mental operators are not given a grounded semantics (incidentally, Singh does not seem to use the restricted notation of CTL by which every temporal operators has to be used together with a path quantifier). Fornara and Colombetti (2004) give an operational semantics for an institutional-based approach. The decision to avoid any cognitive aspect produces a verifiable ACL, but an ACL in which the intentional character of communication is missing. Flores and Kremer (2002) use the Z notation for the specification of ACLs, which is not a very familiar notation to give the semantics of programming languages.

Taking our agent communication framework as a starting point, we can extend the interpreted system to define the specification languages of the unified ACL. On the one hand, the specification language $MLTL_I$ provides a high-level declarative meaning to a set of speech acts which constitute the semantics (SAL) of the speech acts. On the other hand, $NLTL_I$ offers a formal definition of the normative pragmatic component (NPRAG). Finally, it should be noted that although we choose to extend the interpreted system defining motivational and normative operators, this is by no means the only way to define an ACL taking the computational model IS as a starting point.

We are aware of the standardization effort led, among others, by FIPA. Thus, proposing yet another different paradigm for the specification of ACLs would not contribute to the standardization of agent communication. Therefore, our unified ACL takes as a starting point the FIPA ACL in order to define two different but complementary components: SAL and NPRAG. The semantics is concerned with the communicative acts whereas the pragmatics refers to normative rules that regulate their use. This allows us to preserve both the cognitive and the social aspects of agent communication. Specifically, the semantics of the unified ACL defines all the FIPA communicative acts using the tools developed in this chapter. This is the main task of the next chapter.

5

ACL semantics: $MLTL_I$ and Speech Acts Library

In chapters 2 and 3 we discussed the claim that agent communication is based on social conventions and how the performance of speech acts is affected by the context, including the role of the participants. The recognition of the communicative intention of the sender is a pragmatic process that should lead to the success of the linguistic interchange. It has also been argued that pragmatic processes are, more often than not, required to enrich the underdetermination of the semantic meaning. In fact, we claim that the pure semantic meaning of an utterance is not enough to work out the full communicative meaning or speech act which the speaker intended to perform. This relies on the intuition that every speech act is performed in a context and with respect to some background information.

One of the commonly agreed requirements for ACLs in open multi-agent systems is that ACL semantics be clearly and explicitly defined. It will be unrealistic that such reasoning about the sender's intentions takes place in open multi-agent systems applications (while agents are supposed to be using their resources in performing some tasks). Besides, we argued that providing a well-defined semantics (set of speech acts or communicative acts) of an ACL does not free the designer of the task of developing a pragmatic theory to regulate agents' understanding process. To put it plainly, a procedure to constrain agents' communicative behaviour in dialogue is needed.

In our approach, the ACL semantics (speech acts library) defines the meaning of the communicative acts, and the ACL pragmatics consists of a set of policies and protocols that model agents' communicative behaviour in terms of the context and scenario in which the dialogue is taking place. At the same time, we stress the importance of grounding the unified ACL in a computational model so we relate the ACL specification languages $MLTL_I$ and $NLTL_I$ to a interpreted system. Agent communication is an intentional activity. But relying on private mental states for the formalization of the ACL semantics had the undesirable consequence of our ACL not being public and, therefore, verifiable. we present an approach by which agents' beliefs, goals and intentions are defined *externally.*

It has also been shown that no ACL proposed so far presents both semantic and pragmatic components in the same framework. In this sense, we enumerated a number of shortcomings of the majority of ACLs proposed to date, which we classified according to the mentalistic, procedural and social approaches discussed in chapter 3. The problems are summarized in table 5.1.

Requirements	**ACLs**		
	FIPA ACL	Procedural	Singh (2000)
Autonomous	?	-	✓
Complete	-	✓	✓
Contextual	-	-	-
Declarative	✓	-	✓
Formal	✓	-	✓
Grounded	-	-	?
Public	-	✓	✓
Perlocutionary	-	-	-

Table 5.1. Requirements for ACL semantics.

Having formalized in the previous chapter the basic concepts of an agent communication framework, providing the definition of an interpreted system upon which the UACL specification languages, $MLTL_I$ and $NLTL_I$, are built, we will devote the next two chapters to each of the two major components of our agent communication framework:

- ACL semantics or Speech Acts Library (SAL): We define in section 5.1 $MLTL_I$ as our semantic specification language; that means that the meaning of speech acts such as 'inform' will be given in terms of $MLTL_I$ formulae. $MLTL_I$ include cognitive operators for beliefs, goals, intentions and temporal operators to express the properties of system states. The second part of the chapter establishes a taxonomy of speech acts in terms of the preconditions (section 5.2) and defines a complete set of speech acts covering every category of the taxonomy.
- ACL pragmatics or Normative Pragmatics (NPRAG): $NLTL_I$ includes deontic and temporal notions to express the meaning of the pragmatic policies

and protocols which constrain agents' communicative behaviour. We provide examples of scenarios in which our pragmatic theory can be used.

Taking as a starting point the computational model presented in 4.4, sections 5.1.6 and 5.1.7 define the syntax and semantics for our semantic specification language $MLTL_I$. Besides, we discuss (section 5.1.1) and justify the properties assigned to the operators for beliefs, goals and intentions.

Then, a complete set of communicative acts or speech acts are defined in section 5.1.7. We present a definition for each of the speech acts defined in FIPA ACL (2002). Of course, since our semantic specification is different from the Semantic Language used for FIPA ACL, the result will be quite different. However, we believe that the resultant speech acts library facilitates the verification of the ACL without losing any expressive power. Furthermore, we believe that temporal logic is suitable to model agent communication because it captures how agents' states change over time. Table 5.2 states the list of requirements that the semantics presented in this chapter aims to fulfil.

Requirements	**ACLs**
	$\mathcal{SAL}$
Autonomous	✓
Complete	✓
Contextual	-
Declarative	✓
Formal	✓
Grounded	✓
Public	✓
Perlocutionary	-

Table 5.2. Aims of SAL.

Note that, unlike many of other ACL semantics approaches discussed, we do not aim to satisfy the contextual and perlocutionary requirements. That this is a task for the ACL pragmatics.

5.1 $MLTL_I$

In this section, we first informally present the main properties of the temporal component of $MLTL_I$ to focus then on the operators for beliefs, goals and intentions. A syntax, semantics and axiomatics for $MLTL_I$ are provided. The main contribution of this section is that we define a multimodal linear time logic of belief, goals and intentions with respect to a computational model defined as an interpreted system in the previous chapter. An other way to express it is that $MLTL_I$ provides a logic in the spirit of the BDI logics (with goals instead of desires and with linear time instead of branching) of Rao and Georgeff (1991a) but with the mental operators defined with respect to runs. Furthermore, $MLTL_I$ closest relation is with respect to the logic of knowledge of Halpern and Vardi (1988). We discuss these issues in section 5.3.

LTL is a type of temporal logic with connectives that allows us to refer to the future. The general idea is that while the past is determined and cannot be changed, the future is non-deterministic and can take different courses from the present moment. Thus, assertions about future events are relative to that course of future events which is considered actual at the current moment. Since the future is not determined, we consider several computation paths, representing different possible futures, any of which might be the actual path that is realized.

LTL are evaluated on paths, that is, we define that a state of a system satisfies an LTL formula if all paths from the given state satisfy it. In short, LTL describes properties of individual executions and its semantics is defined as a set of executions. There have been many debates about the advantages of using LTL over CTL or viceversa. Nowadays it is generally agreed that the expressiveness of both logics is not comparable. CTL allows explicit quantification over paths, but it does not allow one to select a range of paths by describing them with a formula, as LTL does. For example, in LTL it is possible to say "all paths which have a ϕ along them have a ψ along them" by writing $F\phi \rightarrow F\psi$. This cannot be written in CTL because of the constraint that every temporal operator F has to be used with an associated path quantifier E or A. Ultimately, the choice usually depends on the purposes of the specification and the personal preferences of the designer. LTL is more widely used because CTL formulae are sometimes harder to understand.

Temporal model checking using linear temporal logic is one of the most used techniques for system verification. This feature is important to investigate different methods of verification for the ACL pragmatics. The other component of our specification languages $MLTL_I$ and $NLTL_I$ consists of the cognitive operators for beliefs, goals and intentions and a motivational operator for obligation, respectively. The purposes of this section are twofold: On the one hand, we describe the properties of beliefs, goals and intentions of $MLTL_I$ and discuss how to ground them in the interpreted system IS. On the other hand, we set up the general semantic framework for $MLTL_I$ which is used to define $NLTL_I$ in the next chapter.

5.1.1 Formalizing Mental Attitudes

According to WordNet's definition, *motivation* is "the psychological or mental feature that arouses an organism to action toward a desired goal, or that which gives purpose and direction to behaviour". Propositional or mental attitudes such as beliefs, goals, desires, and intentions are key elements of agents' mental states that serve as the basis for its decisions about when and how to act in the world. Besides, goals and intentions are crucial in order to model communicative actions (such as *promise*) in which agents express the *intention* to execute an action. Conceptually, beliefs are considered statements of properties of its world that an agent takes to be true; goals are states that an agent wants to bring about, and intentions are those actions or plans that an agent is committed to perform.

Traditionally, agents beliefs, desires and intentions in the BDI paradigm can be seen as capturing agents' informational, motivational and deliberative states, respectively. The BDI paradigm has its roots in the work by Dennett (1981) and Bratman (1987) presented in section 1.1, which explored the value of mentalistic descriptions and the critical role of intentions in tractable, practical reasoning, creating a new field of study: BDI theory. The intuition behind the ascription of mental states to machines was argued by McCarthy as follows:

> "To ascribe beliefs, free will, intentions, consciousness, abilities, or wants to a machine is legitimate when such ascription expresses the same information about the machine that expresses about a person. It is useful when the ascription helps us understand the structure of the machine, its past or future behaviour, or how to repair or improve it. [...] Ascription of mental qualities is most straightforward for machines of known structure such as thermostats and computer systems, but it is most useful when applied to entities whose structure is incompletely known." (McCarthy, 1990)

The main problem for the design of ACLs (as argued in section 3.2.1) is that the formalization of the BDI attitudes depends on agents' private mental states, effectively making the theory unverifiable. In the following, we explore the distinguishing properties of beliefs, goals and intentions for the specification of the unified ACL. In order to tackle this problem, we build on Halpern and Vardi (1989), where the notion of knowledge is defined *externally.*

Roughly, there are two main semantic approaches to formalizing agent systems via modal logics. The traditional model is based on the work of Kripke (1963) and Hintikka (1962) on possible-world models, which it has been described in the previous chapter (section 4.2). It has been mentioned that possible-world models cannot be related to a computational model, so we proposed to adapt the interpreted systems approach to model agent communication in multi-agent systems. In order to do this, we extend the interpreted system to formalize beliefs, goals and intentions. The possible-world approach includes the theory of intention (Cohen and Levesque, 1990) and the belief-desire-intention (BDI) paradigm of (Rao and Georgeff, 1991a). Some of these logics can be reduced to standard concurrent logics of programs such as mu-calculus (Schild, 2000), but it it is still not clear how to get concrete agent models with the belief, desire and intention accessibility relations from specific programs. In other words, no clear correspondence can be drawn between the notion of world in the possible worlds model and that of state of a computational system. Appropriate grounded semantics ensures that a clear correspondence can be found between states in the computing system and configurations in the logical description (see van der Hoek and Wooldridge (2003) for a good discussion on these issues).

In our approach it is possible to associate the system with a computer program, and logic formulae can be understood as properties of program computations. This is what we meant by saying that (see section 4.3) these kind of models are computationally grounded (Wooldridge, 1999). There are very few computational grounded models for formalizing general agents which present propositional attitudes (beliefs, goals, intentions, desires). Executable agent languages such as AgentSpeak(L) (Rao, 1996) and 3APL (Dastani et al., 2004) have a relatively simple semantics (van der Hoek and Wooldridge, 2003).

5.1.2 Kripke Structures

The semantics of both $MLTL_I$ and $NLTL_I$ is given by associating the interpreted system IS to a Kripke structure M_I. The basic concepts of the computational model IS were given in section 4.3 so here we just introduce some central concepts related to Kripke structures that are needed for the semantics of our specification languages.

Kripke (1963) developed the idea that the notions of possibility and necessity could be captured in terms of *possible worlds.* The point is that we live in a world which in fact is a possible world. A fiction work (a film, a fiction novel) is a description of a possible world different from the actual world. If someone says "I saw a green unicorn", then this is true if it is the case that there is a possible world in which there are green unicorns. Thus, in the possible worlds logic, a statement is *true in a world* instead of just *true*. Statements are necessarily true if they are true in every possible world. The logics of knowledge (epistemic) and belief (doxastic) were developed by Hintikka (1962) and later extended by Moore (1980). A modal doxastic logic for n agents is obtained by joining together n modal logics, one for each agent. For the sake of simplicity, it is usually assumed that the agents are homogeneous, that is, that they can be described with the same logic. Such a system is denoted by the same name as the modal system, but with the subscript n, such as B_n, which is the logic consisting of n copies of the logic K. A Kripke structure can be defined as follows:

Definition 11 (Kripke structure).

A Kripke structure M for n agents over ϕ is a tuple $M = \langle S, \mathcal{R}_1, \ldots, \mathcal{R}_n, L \rangle$ composed by a non-empty set of states S, a labelling function *L that associates with each state S in a truth assignment to the primitive propositions in ϕ, such as, $L(s) : \phi \rightarrow$ **true,false** for each state $s \in S$; that is, an interpretation $L : \phi \rightarrow 2^S$ assigns to every atomic proposition the set of states or possible worlds where it is true. Finally, $R_i \subseteq S \times S$ is an accessibility relation between sets states or possible worlds, that is, is a set of pairs of elements of S.*

The labelling $L(s)$ tells us whether ϕ is true or false in state s. For example, if ϕ denotes the fact that "It is windy and rainy in Scotland", then $L(s)(\phi) = \textbf{true}$ captures the situation in which it is both windy and raining in Scotland at a state s of the structure M. The accessibility relation $\mathcal{R}_i$ captures the possibility relation according to agent $i : (s,t) \in \mathcal{R}_i$ if agent i considers state t possible, given its information in state s.

Definition 12 (Satisfaction in Kripke structures).

The basic concept is that the truth of a formula ϕ at an state s of a model $M = \langle S, \mathcal{R}_1, \ldots, \mathcal{R}_n, L \rangle$. This can be read as "$\phi$ is true at (M,s)" or "ϕ holds at (M,s)". This is denoted by $M, s \models \phi$, and it is recursively defined as follows:

$M, s \models \phi$ *iff* $\pi(s)(\phi)$ *for* $\phi \in AP$
$M, s \models \neg\phi$ *iff* $(M, s) \not\models \phi$
$M, s \models \phi \rightarrow \psi$ *if* $M, s \models \phi$ *implies* $M, s \models \psi$
$M, s \models \Box\phi$ *if* $M, t \models \phi$ *for every* $t \in S$ *such that* $(s,t) \in \mathcal{R}_i$

Validity on Kripke structures is defined as follows:

Definition 13 (Validity in Kripke structures).

Given a structure $M = (S, \mathcal{R}_1, \ldots, \mathcal{R}_n, L)$ we say that ϕ is valid in M, and we write $M \models \psi$, if $(M, s) \models \phi$ for every state $s \in S$. Besides, we say that ϕ is valid, and write $\models \phi$, if ϕ is valid in all structures.

The accessibility relations $\mathcal{R}_i$, (where $i = 1, \ldots, n$), give the properties of the structures. For example, if the accessibility relation is defined as an *equivalence* relation on S, then the properties of the structure are:

- Reflexive: $\forall s \in S$, we have $(s, s) \in \mathcal{R}$
- Symmetric: $\forall s, t \in S$, we have $(s, t) \in \mathcal{R}$ if and only if $(t, s) \in \mathcal{R}$.
- Transitive: $\forall s, t, u \in S$, we have that if $(s, t) \in \mathcal{R}$ and $(t, u) \in \mathcal{R}$, then $(s, u) \in \mathcal{R}$.

Informally, by considering $\mathcal{R}_i$ to be an equivalence relation we can capture the intuition that agent i considers t possible in state s if in both s and t agent i has the same information about the world, that is, the two different states are indistinguishable to the agent i. This is the way that Halpern and Vardi (1988) characterize knowledge. For the structures of $MLTL_I$ we need to consider which properties may be more appropriate with respect to beliefs, goals and intentions.

In addition to the three defined above, the Euclidean, Serial and Linear relations are quite common:

- Euclidean: $\forall s, t, u \in S$ with $(s, t) \in \mathcal{R}$ and $(s, u) \in \mathcal{R}$ we have $(t, u) \in \mathcal{R}$.
- Serial: $\forall s \in \mathcal{R}$ there is a $t \in \mathcal{R}$ such that $(s, t) \in \mathcal{R}$.
- Linear: $\forall s, t, u \in S$ we have that $(s, t) \in \mathcal{R}$ and $(s, u) \in \mathcal{R}$ together imply that $(t, u) \in \mathcal{R}$, or $t = u$, or $(u, t) \in \mathcal{R}$.

After introducing some central definitions related to Kripke structures, we give a definition for each of the cognitive operators to be used in $MLTL_I$: Beliefs, goals and intentions.

5.1.3 Properties of Beliefs

As a general idea, beliefs can be considered propositions held by an agent to be true. Traditionally, the concept of belief has been distinguished from the concept of knowledge. An agent can believe a proposition but at the same time believe that can it be mistaken. However, knowing a proposition *means* that that proposition is true. Following this, Cohen and Levesque (1985) analyzed knowledge as *justified true belief.* This and other distinctions will not be considered since in open multi-agent systems every propositional attitude may eventually be revised.

Hintikka (1962) characterizes the belief accessibility relations as serial, transitive and euclidean. In the possible worlds approach, that two states are related by the accessibility relation informally means that "the states s, t are possible according to agent's beliefs". For a relation to be *serial* consider that an agent in s believes that t is possible, then if the agent would not believe t to be possible in s that would mean that the beliefs in s are not consistent, which is contrary to our intuition of consistency. Regarding *transitivity,* suppose that an agent in s considers t possible, and that t consider possible u. Then, if the agent did not consider possible u when in s, the agent would believe that u it is not possible. If this is the case, then it would also believe this at t (because it is believed possible at s) and therefore u would not be believed possible from t. With respect to *euclideanness,* suppose that an agent in s believes that t is possible, and that u is possible. Then, if agent does not consider possible u while in t that would mean that it does not consider possible u while in s (because it is believed to be possible at s) and therefore t would not be possible while in s.

Recall the definition of a system of global states in the previous chapter and how we associated Kripke structures to the interpreted system. Let us consider the following global states $g = (l_e, l_i, \ldots, l_n)$, $g' = (l'_e, l'_i)$, and $g'' = (l''_e, l''_i, \ldots, l''_n)$ where l_e, l'_e, l''_e are the environment's state and $(l_i, \ldots, l_n)$, $(l'_i, \ldots, l'_n)$, $(l''_i, \ldots, l''_n)$ the local states of the agents. Then,

Definition 14 (Belief accessibility relation).

We say that the Kripke structure M_I associated to the interpreted system IS where the accessibility relation $\mathcal{B}_i$ that models belief is serial, transitive and euclidean.

Proof.

That M_I is serial follows from the supposition that there is always a global state g such that $g \in GS$, that is that GS is a nonempty set of global states. M_I transitivity: suppose that $\mathcal{B}_i(g, g')$ and that $\mathcal{B}_i(g', g'')$, for a some agents $i \in Ag$. By definition, it must be that $g' \in GS$ and $l'_i \in GS$. We also have that $g'' \in GS$ and $l''_i \in GS$. Therefore, we must also have that $\mathcal{B}_i(g, g'')$.
M_I is euclidean: if we assume that $\mathcal{B}_i(g, g')$ and $\mathcal{B}_i(g, g'')$, then we have that $g'' \in GS$ and $l''_i \in GS$. Therefore, $\mathcal{B}_i(g', g'')$.

An alternative and perhaps more intuitive method to define the properties of the cognitive operators for beliefs is to characterize the formulae that are always true in our interpretation with respect to each propositional attitude. In other words, we characterize the accessibility relation $\mathcal{B}_i$ in terms of formula schemes that we can expect to be valid. The schemes are the classic axioms of modal logic (see table 5.1.3).

Name	**Axiom**
PC	All propositional tautologies
Modus Ponens (MP) :	From ϕ and $\phi \rightarrow \psi$ to infer ψ.
Necessitation :	From ϕ to infer $\Box\phi$.
K Kripke Axiom:	$\Box(\phi \rightarrow \psi) \rightarrow (\Box\phi \rightarrow \Box\psi)$.
T Knowledge Axiom:	$\Box\phi \rightarrow \phi$.
D Consistency Axiom:	$\Box\phi \rightarrow \neg\Box\neg\phi$.
4 Positive Introspection:	$\Box\phi \rightarrow \Box\Box\phi$.
5 Negative Introspection:	$\neg\Box\neg\phi \rightarrow \Box\neg\Box\neg\phi$.

Table 5.3. Some traditional Axioms.

The operator for belief we introduce in $MLTL_I$ has the standard form of $B_i\phi$, for which we will define, in section 5.1.7, the notion of satisfaction that determines whether a formula such as $B_i\phi$ is true at a given point of the system. A traditional reading of the belief operator can be "agent i beliefs proposition ϕ", and intuitively we can think of it as having such a reading. The commonly accepted axiomatization for a logic of belief consists of the system $KD45$ which in our case it is over a set of agents n in a multi-agent system; to denote this, we write $KD45_n$.

Definition 15 (Belief Axioms).

We list the properties of belief and provide a formal proof of their validity (usually available in the traditional modal logic literature, such as Chellas (1980)).

PC $\vdash_{KD45} \phi$ *where* ϕ *is any propositional tautology.*

K $\vdash_{KD45} B(\phi \rightarrow \psi) \rightarrow (B\phi \rightarrow B\psi)$

D $\vdash_{KD45} B\phi \rightarrow \neg B \neg \phi$

4 $\vdash_{KD45} B\phi \rightarrow BB\phi$

5 $\vdash_{KD45} \neg B\phi \rightarrow B\neg B\phi$

MP *If* $\vdash_{KD45} \phi$ *and* $\vdash_{KD45} \phi \rightarrow \psi$, *then* $\vdash_{KD45} \psi$

NEC *If* $\vdash_{KD45} \phi$, *then* $\vdash_{KD45} B\phi$

Axiom K states that agent i believes all the logical consequences of its beliefs. Thus, if an agent believes ϕ and believes that ϕ implies ψ, then both 'ϕ' and 'ϕ implies ψ' are true at all states the agent considers possible. Thus, ψ must be true at all states that the agent considers possible, so the agent must also believe ψ.

The Necessitation rule assumes that agent believe all the formulae that are valid in a given structure. Thus, if ϕ is true at all the possible states of structure M_I, then ϕ must be true at all the states that an agent considers possible at any given point in M_I, so it must be the case that $B_i\phi$ is true at all possible states of M. Note that the Necessitation (NEC) rule is not equivalent to the scheme $\phi \rightarrow B_i\phi$ which means that if ϕ is true, then agent i believes ϕ. What the NEC rule states is that if ϕ is valid for all structures M then $B_i\phi$ is also valid. The scheme $\phi \rightarrow B_i\phi$ would imply that agents would necessarily believe all things that are true.

Axiom D assumes that agents cannot both believe a proposition ϕ and its negation $\neg\phi$. In other words, if an agent believes that ϕ is the case, then it does not believe that $\neg\phi$ is the case.

Finally, the last two properties for beliefs we are interested in are those that allow agents to do introspection regarding their beliefs. In this sense, agents believe what they believe and believe what they do not believe. These two properties are represented by the Positive and Negative Introspection Axioms (4 and 5).

Therefore, our notion of belief is given by the conjunction of axioms $KD45$, which is the classical characterization of belief. However, there is an important difference in our approach. Although the axiomatics of beliefs, goals and intentions correspond to the classic axiomatic systems, we define their semantics with respect to runs and global states.

A well known theorem in the literature is the correspondence between the validity of, say, $B_i\phi \rightarrow B_i\phi$ and the property that the accessibility relation $\mathcal{B}_i$ is reflexive (Halpern and Moses, 1992). Both expressions, the Knowledge axiom and the property of reflexivity, are based on the general intuition that anything that an agent knows must be true. Similarly, it is also possible to see a correspondence between the Positive Introspection axiom (4) and the property of $\mathcal{B}_i$ of being transitive.

An axiom system $KD45_n$, for n agents where $i = (1, \ldots, n)$, is *sound* for a language $\mathcal{L}$ with respect to a class M_{ste} of structures where the accessibility relation is serial, transitive and euclidean, if every formula in $\mathcal{L}$ provable in the axiom system $KD45_n$ is valid with respect to M_{ste}. Besides, the axiom system $KD45_n$ is *complete* for a language $\mathcal{L}$ with respect to M_{ste} if every formula in $\mathcal{L}$ that is valid with respect to M_{ste} is provable in the system $KD45_n$. Thus, $KD45_n$ characterizes the class M_I if it provides a sound and complete axiomatization of that class, that is, if for any formula ϕ, $KD45 \vdash \phi$ iff $M_I \models \phi$. By defining soundness and completeness we are giving a tight connection between the syntactic notion of provability and the semantic notion of validity.

Theorem 1. *The axiom system $KD45_n$ is a*

(a) sound axiomatization of the language $\mathcal{L}$ with respect to the structure M_{ste}, where the accessibility relations $\mathcal{B}_i$ are serial, transitive and euclidean.
(b) complete axiomatization of $\mathcal{L}$ with respect to M_{ste} (Halpern and Moses, 1992). See Proof in Annex B.

Table 5.4 shows the relation between the Axioms that characterize the notion of belief and their corresponding properties of $\mathcal{B}_i$.

$\mathcal{R}$ properties	Axioms	Name
Serial	$B_i\phi \rightarrow \neg B_i \neg\phi$	D
Transitive	$B_i \rightarrow B_i B_i \phi$	4
Euclidean	$\neg B_i\phi \rightarrow B_i \neg B_i \phi$	5

Table 5.4. Axioms KD45 and $\mathcal{B}_i$.

For all formulae ϕ and ψ, all structures M_I where the possibility relation $\mathcal{B}_i$ is serial, transitive and euclidean, and all agents $i = 1, \ldots, n$, we have that:

K $\quad M \models B(\phi \rightarrow \psi) \rightarrow (B\phi \rightarrow B\psi)$.

D $\quad M \models B\phi \rightarrow \neg B \neg \phi$.

4 $\quad M \models B\phi \rightarrow BB\phi$.

5 $\quad M \models \neg B \neg \phi \rightarrow B \neg B \neg \phi$.

NEC if $M \models \phi$ then $M \models \Box\phi$.

And these schemes are respectively valid in the indicated classes of structures (i.e., serial, transitive and euclidean).

Proof: Validity of $KD45_n$. Except the proof for axioms 4 and 5 that are usually left to the reader, most of these proofs are well known (Fagin et al., 1995; Chellas, 1980) so we include them as an annex.

Although the notion of belief presented here has mainly followed the traditional axiomatics beliefs by means of a KD45 logic, we have seen that the belief accessibility relation relates global states because the Kripke structure characterized by the belief relation is associated to the interpreted system IS defined in the previous chapter. We will make explicit this point when defining the semantics of $MLTL_I$. Besides, defining the cognitive component of $MLTL_I$ instead of continuously referring to the literature will hopefully help the reader to understand better how we integrate $MLTL_I$ in the interpreted system IS.

We argued in section 3.2.5 that giving a mentalistic semantics to ACLs makes it impossible to verify the ACL. However, in the characterization of belief we have provided, we were using the intuitive notion of "an agent believing a proposition to be true", which has been traditionally given a possible world semantics. Obviously, this seems to be contradictory to our claim that the operators to express the cognitive states of agents were to be grounded in the computational model IS.

Furthermore, since we want to consider how the multi-agent system changes over time, having a notion of belief (or goal and intentions, respectively) is not enough. By placing the notion of belief defined here into the framework defined in section 4.5 to model ACLs in multi-agent systems, we can capture how agents' propositional attitudes change. Our framework therefore allows to talk about cognitive concepts and their interactions in the environment over time.

In section 4.5, we described a system in terms of runs and global states that changes over time. We assumed that each agent is in some local state at some point of time. *Local states* encode all the information available to the agent at that time. To be able to talk about other aspects of the system, such as the history of messages if we are interested in modelling communication, we defined also an *environment*, which encodes everything else of the system that is not part of agents' local states. Let us remind the global states and runs definitions:

Definition 16 (Global states).

A global state g of a system with n agents is the tuple $(l_e, l_1, \ldots, l_n)$, where l_e is the state of the environment and l_i is agent i's local state. As with local states, we assume that the system is in a unique global state at a given time. If the global state at some time is $g = (l_e, l_1, \ldots, l_n)$, then we define $g_e = l_e$ and $g_i = l_i$ for $i = 1, \ldots, n$, to denote the environment's local state and the agent's local state respectively.

Definition 17 (Runs).

A run is defined as a sequence of global states such that $g_0, g_1, \ldots, g_n$. In this sense, runs are seen as computational paths. Two runs g, g' are said to be *equivalent if* $(g_0, \ldots, g_n) = (g'_0, \ldots, g'_n)$. *A system for* n *agents consists of a set* T *of runs* r.

The idea is that "agent i believes ϕ" means that, "with respect to agent's beliefs, the system could be at a point where ϕ holds". In other words, beliefs refer to points in some run that an agent think possible. From now onwards, this is the notion we will use when talking about agents' beliefs. The accessibility relation for belief remains to be serial, transitive and euclidean; the different is placed in the accessibility relation characterizing a Kripke structure associated to the computational model IS. In $MLTL_I$ (5.1.7), beliefs are interpreted in terms of the possible, according to the agent, points in the interpreted system.

Unlike traditional mentalistic approaches to the semantics of ACLs, the notion of belief used in this thesis does not require that the belief be true. Therefore, an agent holding a belief does not automatically made the content of the belief to be true. This property is central for open MAS, where agents have available incomplete and defeasible information.

Next two sections introduce the notion of goal and intentions in a similar way to that of belief. However, most of the apparatus presented here is easily adapted for the definition of goal, so instead of repeating ourselves we will refer to the relevant parts of this section when necessary.

5.1.4 Properties of Goals

We strongly believe that, despite several efforts to avoid the motivational aspect of communication when modelling ACLs (Fornara and Colombetti, 2004; Flores and Kremer, 2002; Pitt and Mamdani, 1999) and to a lesser extent Singh (2000), agent communication is an intentional and goal-based activity. According to this view, which is coherent with most of the major pragmatic theories presented in chapter 2, agents perform communicative acts in order to achieve some specific goal, not only to express the commitment of doing something. When an agent i sends an inform message to j that ϕ, i wants to bring about a particular state, for example, that agent j believes that ϕ. We are aware that adding a new primitive makes the system more complex. Thus, a possible alternative would be to translate attitudes such as goals in terms of beliefs. For example, goals can be states that agents believe are desirable, meaning that an agent i believes it is good that an agent j will believe that ϕ.

However, we believe that this does not capture the full meaning of an agent having a goal. It is not a mere desire, but a state a particular agent really wants to bring about. So, if i wants j to believe that ϕ, it will send an *inform* message to j. The translation in terms of belief of the goal statement does not imply that i is actually going to do something to achieve them, but only that it would be good to have them. It therefore seems justified to introduce a new operator for a new propositional attitude that affects agents behaviour. As with beliefs, our view of goal is *external*, so having a goal in our system refers to those runs in which there are some global states that the agent wants to happen.

The story of formalization of goals in multi-agent system research is a curious one. A major paradigm under which goals have been heavily studied is BDI (Rao and Georgeff, 1991b). Traditional formal treatments of the BDI model (Rao and Georgeff, 1991a; Wooldridge, 2002) attempt to capture the static and dynamic properties of beliefs, desires and intentions. The singularity of the story is that the vast majority of these BDI-based models do not deal actually with desires, but with goals, that is, goals have been disguised as desires in most of the literature. In fact, Rao and Georgeff (1991b) require desires to be consistent which, as they acknowledge, means that they are actually formalizing a notion of *goal*. Goals are considered to be consistent as opposite to desires because you just cannot want to achieve a goal ϕ and at the same time want to bring about a state $\neg\phi$. However, one can easily think of desires as being potentially contradictory.

When a goal is adopted by an agent, that means that agent's goals serve as the starting point of a process which results in the agent starting a process to achieve specific goals by acting upon intentions. Goals are required to be achievable, so that the adoption of goals by agents will take into account the feasibility of achieving a particular goal, given the current global state of the system. For example, one may have the desire on Friday evening to go out for a few drinks and relax with friends and, at the same time, have the desire to finish the correction of a research paper by that same Friday evening, so he does not need to work over the weekend. However, if we consider the situation in which one has the goal to finish the paper by Friday evening – which means that the agent expresses the intention to execute a series of actions towards the achievement of that particular goal -, then it is not possible to have the goal to relax with your friends, since the achievement of that goal would require a series of steps that conflict with the goal of finishing the paper (van der Hoek and Wooldridge, 2003).

This arguably long comparison of goals and desires can be significantly shortened if we look at the properties we want the notion of goal to have in our system.

Quite simply, we require our notion of goal to be serial, that is, if the agent has the goal of bringing about ϕ, then it has not got the goal of achieving $\neg\phi$, that is, it respects the Consistency Axiom D, which is also the interpretation traditionally favoured when formalizing goals. The characterization of the properties of *goal* will follow the same steps of the characterization of belief. Thus, on the one hand, we define the notion of *goal* in terms of the properties of an accessibility relation $\mathcal{G}_i$. On the other hand, we specify the set of axioms that are to be valid. The definitions and formalizations of the previous section (5.1.3) apply entirely to the definition of goal, except those issues directly relevant to the transitive and euclidean relations and the axioms 4 and 5.

Thus, we have a Kripke structure M_I for n agents which is characterized by a serial accessibility relation $\mathcal{G}_i$. We introduce an operator for goal, $G_i\phi$, for which we define, in section 5.1.7, its truth, that determines whether a formula such as $G_i\phi$ is true at a particular point in the system. An intuitive reading of the goal operator may be "agent i has the goal of bringing about ϕ". However, as it has been made clear when defining belief, a more precise reading adapted to the interpreted system is considered when defining the semantics of $MLTL_I$ (5.1.7).

As with belief, the properties of the goal accessibility relation are described by establishing a number of properties and provide a formal proof of their validity. We will not repeat here the process so the reader is referred to the previous section.

Definition 18 (Goal Axioms).

The axiomatics for Goals are characterized by the system KD_n:

K $\quad \vdash_{KD} G(\phi \rightarrow \psi) \rightarrow (G\phi \rightarrow G\psi)$.

D $\quad \vdash_{KD} G\phi \rightarrow \neg G \neg\phi$.

MP *if* $\vdash_{KD} \phi$, *and* $\vdash_{KD} \phi \rightarrow \psi$, *then* $\vdash_{KD} \psi$

NEC *if* $\vdash_{KD} \phi$ *then* $\vdash_{KD} G\phi$.

Axiom K says that any formulae implied by the current goal of the agent are also goals of the agent. If an agent's goal is ϕ and having the goal of ϕ implies ψ, then both 'ϕ' and 'ϕ implies ψ' are true at all states the agent wants to bring about. Thus, ψ must be true at all states that the agent wants to bring about, so the agent must also have the goal of ψ.

$$\models (G_i\phi \wedge G_i(\phi \rightarrow \psi)) \rightarrow G_i\psi$$

The omniscience property is implied by the Necessitation (NEC) inference rule by which, if ϕ is true, then we can deduce that agent i has the goal of bringing about ϕ. The consequence that all formulae implied by the current goal of the agent are also goals of the agent seems to be quite counter-intuitive. In fact, in order to solve this problem the K axiom has been weakened to:

$$G_i\phi \wedge B_i(\phi \rightarrow \psi)) \rightarrow G_i\psi$$

However, this alternative is also problematic because it causes agents to assume as goals the side-effects (or collateral effects as it is fashionable nowadays) of its real goals. Thus, if an agent has the goal of bombing a bridge, and the agent believes that bombing a bridge will imply the destruction of a nearby school, then the agent assumes as a new goal the destruction of the nearby school. Another very good reason to maintain axiom K is that it constitutes the basis of the standard modal logic. Axiom D distinguishes goals from desires. We require agents' goals to be consistent, as expressed by the Consistency Axiom (D).

The accessibility relation for goals is serial; it maintains a consistent behaviour by which agents cannot have the goal of bring about one state and its contrary. The relation $\mathcal{G}_i$ for goal is also defined from an external point of view that is, in relation to global states, as opposed to the traditional notion defined using possible world semantics, like in FIPA ACL (2002); Singh (2000).

In the interpreted system IS an "agent i has the goal of ϕ" means that, "in regards to the agent's goals, the system could be at a point where ϕ holds". Goals can be seen as fact at a global state that an agent wants to bring about. This is the standard notion from now onwards to be used when talking about goals.

The states accessible through the accessibility relation $\mathcal{G}_i$ are a subset of those accessible through the accessibility relation for beliefs $\mathcal{B}_i$. It is quite common that the set of goal states is a subset of those believed possible, such that $G_i \subseteq B_i$. This responds to the common sense claim that there are global states which the agent does not want to bring about.

We finish the introduction of mental attitudes by discussing the properties of the notion of intention. We analyze the meaning of ACL messages in terms of the communicative intention when sending a particular message, inspired by the work of Grice (1975) and Searle (1969) (see section 2). The communicative intentions in turn corresponds to the illocution of the message, which is encoded in the preconditions of the speech act.

5.1.5 Properties of Intentions

In trying to formalize intentions, some authors have tried to reduce the concept of intention to some combination of belief and desire, and others to those of beliefs and goals. The content of beliefs and knowledge is considered to be propositions, whereas in the case of intention, its content has usually been considered to be an action (Tuomela and Miller, 1988).

Bratman (1987) distinguished between present-directed intentions and future-directed ones. Present-directed intentions *causally* produce behaviour, for example, "moving an arm". Future-directed intentions guide agents' planning and constrain their adoption of other intentions. An example of this type of intention may be "going to the cinema tomorrow". Bratman (1987) also argues that *intending to do something* is not the same as *doing something intentionally.* In particular, *intending to do something* is related with the coordination of an agent's plans. Intention it is understood in this thesis as a future-intention.

Following Bratman (1987) and Searle (1983), intentions are analyzed separately from beliefs and goals. The agent has to commit itself to one plan of action which will be re-valuated only in the case the environment significantly changes. Moreover, agents need to coordinate their plans to execute future actions. Once a future action is intended (committed to), an agent decides what other actions will be taken on the future along with the original action. This capacity requires that the agent will not simultaneously believe that it will *not* do α. Otherwise, the agent will not be able to plan a subsequent action to α since it believes it will not be done. Therefore, a sort of commitment by the agent is necessary in order to be able to decide what to do next.

According to Bratman, intentions should satisfy the following properties. If an agent intends to achieve ϕ by executing α, then:

1. The agent believes achieving ϕ is possible.
2. The agent does not believe it will not bring about ϕ.
3. Under certain conditions, the agent believes it will bring about ϕ.
4. Agents do not need to intend all the side-effects of their intentions.

Bratman also argues that what an agent intends is a subset of what an agent chooses. For example, an agent might build a plan of action which achieve some state of affairs. If the agent also believes that its action(s) will cause a set of side effects, then the agent has chosen to achieve the goal and the side effects. However, it has only *intended* to achieve the goal. This distinction is based in the fact that if the plan fails to achieve both the expected goal and side-effects, the agent will build a new plan to achieve the same goal but it will not to achieve any side-effects. These ideas have been adopted by Cohen and Levesque (1990), formally defining intentions as choices which the agent is committed to, as it was explained in section 3.2.1.

In this thesis we also ought to consider a special type of intention: the communicative intention. Searle claims that the content of an intention is a causally self-referential representation of its conditions of satisfaction. That is, for an agent to intend to go to a store, the conditions of satisfaction would be that the intention should cause the agent to go to the store.

Grice theory of conversation depends on the notion of intentional meaning. Although the Gricean reflexive definition of speaker's meaning (section 2.2, page 24), is central for linguists studying the communicative intention, Grice provided different definitions of speaker's intention. The one presented previously is reflexive, but few years later he proposed an iterative definition (Grice, 1969, p.92):
"U meant something by uttering x is true iff, for some audience A, U uttered x intending:

(i) to produce a particular response r
(ii) A to think (recognize) that U intends (i)
(iii) A to fulfil (i) on the basis of his fulfilment of (ii)."

The first clause has been criticized because it includes the hearer A in the object of the intention. It is not the main problem of this definition though, since it seems reasonable that the intention of the speaker is directed to an audience A. Actually, the main problem here is that the fulfilment of the perlocutionary effects is a requirement for the fulfilment of the communicative action.

If we do consider the perlocutionary effects as part of the communicative intention, then these effects are conditions for the communicative action to be satisfied. For example, if an agent sends an `inform` message of ϕ to agent j, then the conditions of satisfaction of this message consist of j believing that ϕ. This requirement is clearly too restrictive.

This problem is also present in some ACL semantic approaches. For example, FIPA ACL specification states that the receiving agent is not obliged to fulfil the perlocutionary effects of the speech act, but no alternative solution is provided. This is one of the shortcomings of trying to ensure the fulfilment of pragmatic aspects by means of semantic specifications (only).

Following this point, we propose that the fulfilment of the perlocutionary effects depend on the normative conversation policies specified in the ACL pragmatics. Thus, we simply use intentions to intuitively denote those runs with the choice of global states that agents want to achieve. We define a simple operator for intentions $I_i\phi$ which intuitively means that "i intends to bring about ϕ". In the $MLTL_I$ structure associated to an interpreted system IS, "an agent i having an intention to bring about ϕ", means that from the point of view of the agents' intentions, there is run in which i intends, along that run, to bring about ϕ. In order to avoid repetition, we define the properties of our notion of intention, referring the reader to the two previous sections (5.1.3 and 5.1.4) for the formal details.

Definition 19 (Intention Axioms).

The axiomatics for intentions are characterized by the system KD_n:

K $\quad \vdash_{KD} I(\phi \rightarrow \psi) \rightarrow (I\phi \rightarrow I\psi)$.

D $\quad \vdash_{KD} I\phi \rightarrow \neg I \neg \phi$.

MP *if* $\vdash_{KD} \phi$, *and* $\vdash_{KD} \phi \rightarrow \psi$, *then* $\vdash_{KD} \psi$.

NEC *if* $\vdash_{KD} \phi$, *then* $\vdash_{KD} I\phi$.

The intention accessibility relations $\mathcal{I}_i$ are serial, that is, if the agent intends to achieve ϕ, then it does not intend to achieve $\neg\phi$. Our agents' intentions are consistent: Similarly to goals, the properties of intentions are given by the system KD_n. All formulas that are implied by the current intentions of the agent are also intentions of the agent. If an agent intends to achieve ϕ and achieving ϕ implies ψ, then both 'ϕ' and 'ϕ implies ψ' are true at all states the agent intends to achieve. Thus, ψ must be true at all states that the agent intends to achieve, so the agent must also intend ψ.

When considering together the accessibility relations $\mathcal{I}_i$, $\mathcal{G}_i$ and $\mathcal{B}_i$, different relations between the accessible states have been considered. Rao and Georgeff (1991b) describe three points of view with respect to the relations between the accessibility relations: Strong Realism, Realism and Weak Realism. In strong realism, the set of belief accessible states is a subset of goal-accessible states, which in turn is a subset of the intention-accessible states. This means that if an agent has the goal of achieving ϕ, then it also believes ϕ. Moreover, if an agent intends to achieve ϕ, then it also has the goal to achieve ϕ. That is,

$$\mathcal{B}_i \subseteq \mathcal{G}_i \subseteq \mathcal{I}_i$$

The problem of the strong realism point of view is that agents constrained by this conditions are over-cautious. Agents only have the goal of bringing about propositions which are believed and only intend to achieve a proposition that are part of their goals. Agents cannot have beliefs about propositions that they do not want to bring about.

An alternative is provided by the realism conception, which is expressed by the relation

$$\mathcal{I}_i \subseteq \mathcal{G}_i \subseteq \mathcal{B}_i$$

which states that agents are over-enthusiastic. It has the unwanted effect that if an agent believes ϕ then it will intend to achieve ϕ. It also states that if the agent intends to achieve ϕ, then it has the goal of ϕ, which we agree with. The problem then is considering intentions a subset of goals and beliefs, because it is possible that an agent's intention may fail to achieve ϕ, and the real computing path may not correspond to the agent's belief.

Rao and Georgeff (1991b) proposed a weak realism as an alternative, which states that if agents intend ϕ then they do not have the goal of $\neg\phi$; if agents intend ϕ then they do not believe that $\neg\phi$ and if agents have the goal of ϕ then they do not believe that $\neg\phi$. In our view, this creates the difficulty of not requiring agents to believe the object of their goals. However, the first part of the weak realism is interesting, since it constrains agents to not having goals that are inconsistent to their intentions.

We therefore consider an intersection relation between intention-accessible states and goal-accessible states such that

$$\mathcal{G}_i \cap \mathcal{I}_i \neq \emptyset$$

which means that the intersection between intention-accessible states and goal-accessible states will be non-empty.

We finish here the definition of the properties of the mental attitudes to be included in $MLTL_I$. We have defined one accessibility relation for each of the attitudes, and justified why the properties are appropriate for the purposes of defining a semantic specification language for ACLs which captures the motivational character of agent communication. In the following, we formally present the language $MLTL_I$. First, the vocabulary and syntax are defined, leaving the semantics of $MLTL_I$ for section 5.1.7. After that, in the second part of the chapter, $MLTL_I$ is used to define a Speech Act Library.

5.1.6 $MLTL_I$ Syntax

The syntax of $MLTL_I$ associated to the interpreted system IS consists of the vocabulary of the interpreted system IS introduced in 4.4 which will be extended with a temporal component (which we define below), and the accessibility relations for beliefs, goals and intentions. $MLTL_I$ structures are actually the result of the combination of IS with the accessibility relations $\mathcal{B}_i$, $\mathcal{G}_i$ and $\mathcal{I}_i$ defined for the structure M_I.

The following symbols and abbreviations will be used: = for definitions. To start to construct a formal language, a set of atomic propositions (where each proposition corresponds to a variable in the model) and the usual Boolean connectives are introduced: negation $\neg$, disjunction $\vee$, conjunction $\wedge$, conditional $\rightarrow$, and material equivalence $\leftrightarrow$. Atomic formulae will be denoted by ϕ, ϕ_0, ϕ_1, $\psi \ldots$

The operators X, F, G, U are called the temporal operators. All the temporal operators are interpreted relative to a *current global state*. There are many runs (sequences of global states) of the system starting at the current state. The temporal operators describe the ordering of events in time along a run and have the following intuitive meaning:

- $F\phi$ (reads "ϕ holds sometime in the future") is true of run if there exists a global state in the run where formula ϕ is true.
- $G\phi$ (reads "ϕ holds globally") is true of a if ϕ is true at every global state in the run.
- $X\phi$ (reads "ϕ holds in the next state") is true of a path if ϕ is true in the state reached immediately after the current state in the run.
- $\phi\ U\psi$ (reads "ϕ holds until ψ holds", is true of a run if ψ is true in some state in the run, and ϕ holds in all preceding states. In other words, ψ does eventually hold and that ϕ will hold everywhere until ψ holds.

Definition 20 **(*$MLTL_I$* Syntax).**
Our specification language $MLTL_I$ consists the following (consider n agents):

C1 If ϕ is an atomic proposition of AP then ϕ is a $MLTL_I$ formula.
C2 If ϕ and ψ are $MLTL_I$ formulae, then so are $\neg\phi$ and $\phi \wedge \psi$.
C3 If ϕ is a $MLTL_I$ formula then $B_i(\phi), G_i(\phi), I_i(\phi)$ are also $MLTL_I$ formulae.
C4 If ϕ is a $MLTL_I$ formula then so are $X\phi$, $F\phi$, $G\phi$, $\phi\ U\ \phi$.

We use $True$ and $False$ as shorthands for $\phi \vee \neg\phi$ and $\neg True$ respectively. Although we have include in the syntax every temporal operator, we can define X, F and G as abbreviations:

$X\phi \equiv False\ U\ \phi$

$F\phi \equiv True\ U\ \phi$
$G\phi \equiv \neg F \neg \phi$

The next operator X is true at some state s whenever ϕ is true at some future point t and there are no other states between s and t. F holds if a formula is true at some point in the future and G is true always in the future, that is, there is not a future global state in which ϕ is not true.

We can conventionally establish several binding priorities for $MLTL_I$ connectives. The unary connectives ($\neg$, the temporal connectives G, F, X, and the mental attitudes operators B_i, G_i and I_i) bind most tightly. Next in priority are $\wedge$ and $\vee$, and finally $\rightarrow$ and U.

The basic intuition to describe and to ascribe cognitive states to the agents in the system from an external point of view is to understand that $MLTL_I$ is a structure which is generated by associating the interpreted system IS with the serial, transitive and euclidean structures M_I, so that beliefs, goals and intentions refer to *runs* of the multi-agent system. The fundamental notion in this approach is the one of *local state*. If we look the system at any point in time, every agent is in some unique *state*. The only assumptions we need to make about local states is that all the information that agents' possess of the system is encoded in their local state. Now, given that we are interested in having an ACL semantic specification language which can be used to describe the unique state of a multi-agent system at each point in time so we do not rely on the agents' internal states to evaluate and verify their communicative behaviour, we need not only to describe the local state of the agents but also the rest of the multi-agent system, which is called the *environment*. For example, when analyzing a system where agents send messages along some communication channel, it is quite useful to keep a record or history (see section 4.4) of the messages that have been sent. Thus, when describing a multi-agent system as a whole (agents and environment), we use the notion of *global state*.

These ideas are formalized in the following section, in which we provide a semantics for $MLTL_I$.

5.1.7 $MLTL_I$ Semantics

At this stage, IS as defined in section 4.4 is a minimal system which we will extend here with time operators. We have defined $MLTL_I$ structures by associating IS with the M_I structures for beliefs, goals and intentions, so that the semantics for the cognitive operators refers to the interpreted system. In other words, we ground the semantics of our specification language $MLTL_I$ in the computational model IS.

Let us remind that a multi-agent system consisted in a nonempty set T of runs, where each run r is a function from time to global states g such that $g \in GS$. We assume that time is discrete and ranges over the natural numbers. Let us remind the definition of run and interpreted system.

Definition 21 (Runs (revisited)).

A run r over nonempty sets of global states GS is a sequence of global states in GS that gives a complete description of an execution. A point consists of a tuple (r, m) where r is a run and m is the time. If $r(m) = (s_e, s_1, \ldots, s_n)$ is the global state at point (r, m), then we say that $r_e(m) = s_e$ and $r_i(m) = s_i$, for $i = 1, \ldots, n$.

Definition 22 (Interpreted System (revisited)).

A system T over a set of global states GS is a set of runs over GS. An interpreted system is a pair (T, L) where T is a system of runs over global states and L is a labelling function for the atomic propositions AP over GS, which assigns truth values to the atomic propositions at the global states. For every $\phi \in AP$ and $g \in GS$, $L(g)(\phi) \in \{true, false\}$. A point is in the interpreted system IS if $r \in T$. Formally, an interpreted system IS is defined by the tuple $(\Sigma, T, GS_0, L, \mathcal{J}, \mathcal{C})$.

For simplicity, when defining the $MLTL_I$ structures and their semantics we will not be using the complete definition of IS. Instead a reduced version consisting of the set of runs T and the labelling function will be used. These two elements are the minimum necessary to define the semantics, so we will refer to the fair runs and initial states only when it is relevant to the discussion.

We have seen in section 4.3 how to associate Kripke structures with a basic system of runs which is based on the technique of Fagin et al. (1995) to ascribe knowledge to agents. We generate a Kripke structure $MLTL_I$ from an interpreted system IS, so that the accessibility relations for beliefs, goals and intentions in a Kripke structure M are serial, transitive and euclidean relations.

Definition 23 ($MLTL_I$ structure).

Given a system of runs T, the structure $MLTL_I$ is generated by associating the interpreted system $IS = (T, L)$ with the serial, transitive and euclidean Kripke structures $M_I = (S, \mathcal{B}_i, \mathcal{G}_i, \mathcal{I}_i, L)$, such that $MLTL_I = (GS, \mathcal{B}_i, \mathcal{G}_i, \mathcal{I}_i, L)$ where:

- *GS corresponds to the sets of global states in IS.*
- *L is a labelling function $L : S \rightarrow 2^{AP}$ from global states to truth values, where AP is a set of atomic propositions. This function assign truth values to the primitive propositions AP at each global state in GS.*
- *$\mathcal{B}_i$ where $i = (1, \ldots, n)$ is a set of agents, gives the accessibility relation on global states, which is serial, transitive and euclidean. Thus, we have that $(l_e, l_1, \ldots, l_n)$ $\mathcal{B}_i$ $(l'_e, l'_1, \ldots, l'_n)$ if $l'_i \in GS_i$. If $g = (l_e, l_1, \ldots, l_n)$, $g' = (l'_e, l'_1, \ldots, l'_n)$, and l_i $\mathcal{B}_i$ l'_i, then we say that g and g' are $\mathcal{B}_i$-accessible to agent i. The formula $B_i\phi$ is defined to be true at g exactly if ϕ is true at all the global states that are $\mathcal{B}_i$-accessible from g.*
- *The accessibility relations for goals $\mathcal{G}_i$ and intentions $\mathcal{I}_i$ are defined in the same manner.*

Both the relations for goals and intentions are serial, so we simply adopt their definition to say that the accessibility relations that characterized goals and intentions between two global states, g $\mathcal{G}_i$ g', and $g\mathcal{I}_i$ g' respectively, are serial. Just to remind that given that $g = (s_e, s_1, \ldots, s_n)$ is the global state, we say that $g_e = s_e$ and $g_i = s_i$ for $i = 1, \ldots, n$; this means that g_i is the local state of agent i at the point at a given time. Agents' beliefs, goals and intentions are defined with respect to their local states and can be induced to relate points. We will use sometimes the simplified notation for global states g for convenience.

We can now apply the previous definition to define truth for a formula ϕ at a global state $r(m)$ of the interpreted system IS.

Definition 24 (Satisfaction in IS with respect to $MLTL_I$).

In this framework, to say that a formula ϕ is true at a point at a global state $r(m)$ in an interpreted system IS if it is true in the related $MLTL_I$. Formally,

$$(IS, r, m) \models \phi \text{ if } (MLTL_I, s \models \phi).$$

We would like to remark that the semantics of the accessibility relations presented here relates global states and not points. We choose global states to stress the intuitions coming from the computational model IS. Moreover, it allows us to give a natural definition to the time operators.

Definition 25 ($MLTL_I$ semantics).

The semantics of $MLTL_I$ is inductively defined as follows:

$(IS, r, m) \models \phi$ *iff* $L(r, m)(\phi) = true$
$(IS, r, m) \models \phi \wedge \psi$ *iff* $(IS, r, m) \models \phi$ *and* $(IS, r, m) \models \psi$

$(IS, r, m) \models \neg\phi$ *iff it is not the case that* $(IS, r, m) \models \phi$

$(IS, r, m) \models B_i\phi$ *iff* $\forall(r', m')$ *such that* $(r, m)\ \mathcal{B}_i\ (r', m')$, *then* $(IS, r', m') \models \phi$

$(IS, r, m) \models G_i(\phi)$ *iff for all* (r', m') *such that* $(r, m)\ \mathcal{G}_i\ (r', m')$, *then* $(IS, r', m') \models \phi$

$(IS, r, m) \models I_i(\phi)$ *iff for all* (r', m') *such that* $(r, m)\ \mathcal{I}_i\ (r'm')$, *then* $(IS, r'm') \models \phi$

$(IS, r, m) \models X\phi$ *iff* $(IS, r, m+1) \models \phi$

$(IS, r, m) \models F\phi$ *iff for some time* $m' \geq m$ $(IS, r, m') \models \phi$

$(IS, r, m) \models G\phi$ *iff for all time* $m' \geq m$ $(IS, r, m') \models \phi$

$(IS, r, m) \models \phi U \psi$ *iff there is some time* $m' \geq m$ *such that along the run such that* $(IS, r, m') \models \psi$ *and for each* $m \leq m'' < m'$ *we have* $(IS, r, m'') \models \phi$.

There are various issues worth to comment on the semantics of $MLTL_I$: L is a labelling function on global states, that is, the truth of a primitive proposition ϕ at a state g depends only on the global state g, since the global state encapsulates all the system information at a particular point. However, there are situations, such as "agent i receiving agent j's message", where its truth does not depend on the whole global state, but only on the agents' local state. On the other hand, there are other statements which describe situations in which their truth depends on more than the global state. An statement such as "at some point in the run, the variable x is set to 5" (example from Fagin et al. (1995)) could be true at the global state g, and false at the same global state of g at a different time. This problem is solved by introducing the temporal operators, so we can easily express the idea that something is to be true in the system at some later time, namely, $F\phi$. The formula $\phi\ U\ \psi$ holds on a run if it is the case that ϕ holds continuously until ψ holds. Moreover, $\phi\ U\ \psi$ actually requires that ψ holds in some future state.

In the interpretation for beliefs, goals and intentions proposed here, these attitudes are ascribed to the agents by an external reasoner about the system. In this approach, agents do not compute their beliefs, goals and intentions in any way, and as a consequence, the communication protocol defined using $MLTL_I$ as the semantic specification language does not rely on agents' internal (mental) states. Note that the properties defined for beliefs, goals and intentions in section 5.1.1 hold in our system for every $MLTL_I$ structure. In the case of $G_i\phi$ and $I_i\phi$ the two points (r, m) and (r', m') are related if $(r'm')$ makes possible to achieve the intention (respectively, the goal) of agent i at the point (r, m).

Agents in multi-agent systems are seen as runs. In the next section we will show how $MLTL_I$ is used to externally ascribe beliefs, goals and intentions in the definition of a set of speech acts. By combining cognitive and temporal operators, we make statements about the evolution of the agents' mental attitudes in the system. For example, we can say that agent i believes that ϕ will eventually hold along a run: $B_iF\phi$.

It is also important to remark that the semantics of $MLTL_I$ could have been presented in a different way, closer to the possible world semantics models (Kripke, 1963), that is, by defining the accessibility relations over points of the system (Halpern and Vardi, 1989; van der Meyde and Wong, 2003). The choice of global states stress the intuitions related to multi-agent systems.

Following our previous definitions, we can define validity for $MLTL_I$ structures:

Definition 26 (Validity in $MLTL_I$).

A formula ϕ is valid in an interpreted system $IS = (T, L)$, that is, $IS \models \phi$, if it is valid in $MLTL_I$: $MLTL_I \models \phi$. For a class $\mathcal{V}$ of IS, we say that a formula is valid in $\mathcal{V}$, $\mathcal{V} \models \phi$, if $IS \models \phi$ for every $IS \in \mathcal{V}$.

There has been quite a lot of work in the Computer Science community on the theoretical aspects of temporal logic. In particular, the issues of decidability, complexity and axiomatizability have been largely studied. If a system is axiomatizable then there is a deductive system to prove all the valid formulae of a system; the soundness and completeness of the axiom systems are also investigated. Decidability and complexity refer to natural decision problems such those of satisfiability (given a formula, does there exist a structure that is a model of the formula?), validity (given a formula, is it true that every structure is a model of the formula?) and model checking (given a formula, together with a particular finite structure, is the structure a model of the formula?). Next section presents an axiomatization for $MLTL_I$ and discusses some issues on the complexity of reasoning about beliefs, goals and intentions with linear time. Then, we will put $MLTL_I$ into use by defining a complete set of Speech Acts.

5.1.8 $MLTL_I$ Axiomatics

Multi-agent systems quite often operate without complete information about their environment, which could include other agents. This thesis presents an approach to use a type of multimodal logic for beliefs, goals and intentions grounded in the interpreted systems model. This allows us to talk about how the characteristics of the system change over time. As far as we know, there is not available in the literature a logic with a grounded semantics for beliefs, goals and intentions, that is, a logic which encodes the informational and deliberative aspects of agents upon a computational model.

We have given a characterization of the properties of our cognitive operators by mean of their axiomatics in section 5.1.1, so we will not repeat the same points here. The axiomatic systems $KD45_n$ for belief and KD_n for goals and intentions have been already proposed, and we also discussed the interaction between beliefs, goals and desires.

The axiomatics for the cognitive operators are as follows. i denotes a set of agents such that $i = 1, \ldots, n$.

PC All instances of propositional tautologies.

MP If ϕ and $\phi \rightarrow \psi$, then ψ.

NEC_b If ϕ, then $B_i\phi$.

NEC_g If ϕ, then $G_i\phi$.

NEC_i If ϕ, then $I_i\phi$.

K_b $B_i(\phi \rightarrow \psi) \rightarrow (B_i\phi \rightarrow B_i\psi)$.

D_b $B_i\phi \rightarrow \neg B_i\neg\phi$.

4_b $B_i\phi \rightarrow B_iB_i\phi$.

5_b $\neg B_i\neg\phi \rightarrow B_i\neg B_i\neg\phi$.

K_g $G_i(\phi \rightarrow \psi) \rightarrow (G_i\phi \rightarrow G_i\psi)$.

D_g $G_i\phi \rightarrow \neg G_i\neg\phi$.

K_i $I_i(\phi \rightarrow \psi) \rightarrow (I_i\phi \rightarrow I\psi)$.

D_i $I_i\phi \rightarrow \neg I_i\neg\phi$.

Regarding the interaction between beliefs, goals and intentions, we argued in section 5.1.5 that we could assume a *weak realism* approach (Rao and Georgeff, 1998). In order to capture this property, we add the following axioms to our system:

ID $I\phi \rightarrow \neg G\neg\phi$.

IB $I\phi \rightarrow \neg B\neg\phi$.

DB $G\phi \rightarrow \neg B\neg\phi$.

Thus, it remains to present the axiomatics for the temporal component of $MLTL_I$. The following axioms are known to provide a sound and complete axiomatization for LTL (Halpern et al., 2004).

PC All tautologies of propositional logic.

T1 $X(\phi \rightarrow \psi) \rightarrow (X\phi \rightarrow X\psi)$.

T2 $X(\neg\phi) \equiv \neg X\phi$.

T3 $\phi\ U\ \psi \equiv \psi \vee (\phi \wedge X(\phi\ U\ \psi))$.

RT1 From ϕ infer $X\phi$.

RT2 From $\phi' \rightarrow \neg\psi \wedge X\phi'$ infer $\phi' \rightarrow \neg(\phi\ U\ \psi)$.

MP From ϕ and $\phi \rightarrow \psi$ infer ψ.

The axiomatic system is denoted by the expression $(B_{KD45}G_{KD}I_{KD})_{LTL}$, which is abbreviated by $MLTL_I - Ax$.

Theorem 2. *The system $MLTL_I - Ax$ is a sound and complete axiomatization with respect to the class of models $MLTL_I$ that are serial, transitive and euclidean.*

Completeness can be shown following the technique used in Halpern and Vardi (1988), who gave a sound and complete axiomatization for a logic with linear time and an operator for knowledge. Furthermore, Lomuscio and Wozna (2006) has very recently given a complete axiomatization for deontic interpreted systems for branching time. Rao and Georgeff (1998) also prove completeness for BDI with branching time. The sketch of the proof is as follows: The general idea is to show that the logic complies with the finite-model property, hence it is decidable. In order to do that, we define two structures, a Hintikka structure for a given formula φ and the quotient structure for a given model. From here we can prove that φ is satisfiable by constructing a Hintikka structure for φ and we build a pseudomodel of $MLTL_I$ structures using its quotient structure. For details, we refer to the reader to the papers cited above since the great length of this proof exceeds the purpose of this chapter.

Our work is obviously related and influenced by the work done on linear temporal logics (Manna and Pnueli, 1995) and the interpreted systems literature (Fagin et al., 1995) about knowledge. The main contribution of $MLTL_I$ is to define a logic where the accessibility relations for beliefs, goals and intentions are defined with respect to runs in the interpreted system.

Most of the formal apparatus defined in this section will be inherited by the ACL pragmatic specification language $NLTL_I$ (see next chapter). The main difference (if only) is that the we combine a deontic operator with the linear time component defined here. In the next section, we use $MLTL_I$ to propose a library of speech acts as the semantics of our unified agent communication language. We provide a complete set of speech acts following the taxonomy of Searle (1969).

5.2 Speech Acts Library (SAL)

We had three main motivations to define the logic $MLTL_I$:

- First, given that $MLTL_I$ is going to define the semantics of the speech acts of our ACL, we wanted this logic to allow operators for beliefs, goals and intentions to express the intentional character of communication.
- Second, we wanted the semantic $MLTL_I$ to be grounded in a computational model, so we had to find a way to include mental attitudes in our language without using the traditional possible world semantics.
- Finally, temporal logic provides useful tools to analyze how a system evolves over time.

Given that there was not a language available with all these characteristics, we have defined $MLTL_I$ with respect to an interpreted system IS (see 4.4) to provide some answers to the points stated above and to those requirements for ACLs considered in the agent communication literature (see section 3.2.5 for a discussion).

In this section we apply $MLTL_I$ to propose a public and grounded semantics for the unified ACL framework as defined in section 4.5. The ACL semantics consists of a Speech Acts Library which is defined using the semantic specification language $MLTL_I$. The main purpose of this semantics is to show how the different validity claims can be understood in terms of our specification language, and formalized using the logic developed. As in FIPA ACL, we provide the illocutionary act as part of the Feasibility Preconditions (FPs). We also specify Rational Effects (perlocution) for completeness and to capture the goal-based aspect that we believe characterizes agent communication. However, as it has been argued in section 4.1, in our approach it is the ACL pragmatics that will regulate conversation so that the Rational Effects can be achieved. Unlike some other alternatives to FIPA ACL (see 3.2) we view our Speech Acts Library as a contribution to the standardization effort lead by the FIPA project. In this sense, the definition of a public, verifiable and declarative semantics for agent communication aims to tackle those shortcomings of the FIPA CAL specification discussed in section 3.2.5. With this point in mind, we not only define at least one speech act or communicative action for each of the categories proposed by Searle (1969), but also a version for each of the communicative actions defined in the FIPA ACL is given. In doing so, we will make explicit several advantages of using $MLTL_I$ as a specification language. In particular, we will remark that:

1. In many cases, the informal description of a speech act includes references such as "at some point in the future", "once the given precondition is true", etc. We will claim that those aspects of the specification can be naturally expressed in a simpler way using $MLTL_I$.
2. Quite often the FIPA CAL specification gives some information about the context of use of the speech act or references about some expected behaviour by the agents. However, neither the context of use nor some constraints on agents' behaviour are determined by the semantics of the communicative acts. However, the consequence is that the language is underdetermined, that is, it is unable to capture the full communicative intention expressed by a message in a specific context. The normative pragmatic theory (NPRAG) proposed in the next chapter help the semantics to contextually enrich the semantics without constraining too much agents' freedom.
3. In relation to the previous point, the semantics of communicative acts includes the expected Rational Effects (perlocution) of performing a speech act. However, for the specification of autonomous agents we cannot guarantee the satisfaction of the Rational Effects. Furthermore, the Rational Effects are not dealt with in any other way in the FIPA ACL specification. This is also where an ACL pragmatic theory should prove useful.

Following Searle (1969), we classify communicative actions into assertives, commissives, directives, declarations and expressives (see 2.1). The last category is not relevant for the purposes of this thesis, so it will not be included (we are not considering *emotional agents*). The syntax of the speech acts is based on the FIPA ACL. Table 5.2 presents some examples of our new definitions of speech acts for each of the four types of categories by means of the four primitives plus two more (*agree* and *refuse*) which will be used later to characterize several interaction protocols.

The two performatives at the top, *inform* and *request*, represent the assertives and directives respectively. *Agree* and *refuse* are included as possible exchanges after the reception of a *request*. *Declare* is an action of the declarative class and *promise* is a commissive. Therefore, the total number of speech acts of SAL is twenty four, although we do not consider this to be a closed catalogue. Conversely, our unified ACL is scalable in the sense that we can define new actions according to our purposes. In this sense, our proposal is not only the specification of an ACL but the definition of a framework in which new semantics (actions) and pragmatics (conversation policies and protocols) can be constructed.

$\langle i, inform(j, \phi)\rangle$ $FP : B_i(\phi) \wedge G_i(B_j(\phi))$ $RE : B_j\phi$	$\langle i, request(j, \phi)\rangle$ $FP : G_i(I_j(F\phi))$ $RE : F\phi$
$\langle i, confirm(j, \phi)\rangle$ $FP : B_i(\phi) \wedge B_i(B_jF\phi \vee B_jF\neg\phi))$ $RE : B_j\phi$	$\langle i, disconfirm(j, \phi)\rangle$ $FP : B_i\neg\phi \wedge B_i(B_j\phi)$ $RE : B_j\neg\phi$
$\langle i, agree(j, \phi)\rangle$ $\langle i, inform(j, (I_i\phi\ U\ \psi))\rangle$ $FP : I_i\phi\ U\ \psi$ $RE : B_j(I_i\phi\ U\ \psi)$	$\langle i, refuse(j, \phi)\rangle$ $\langle i, inform(j, \neg(I_i\phi\ U\ \psi))\rangle$ $FP : \neg(I_i\phi\ U\ \psi)$ $RE : B_j(\neg(I_i\phi\ U\ \psi))$
$\langle i, promise(j, \phi)\rangle$ $FP : I_iF\phi$ $RE : F\phi$	$\langle i, declare(j, \phi)\rangle$ $FP : G_i(X\phi)$ $RE : X\phi$

Table 5.5. A complete set of speech acts

Note that *inform*, *request*, *agree* and *refuse* re-define using $MLTL_I$ their counterparts in FIPA ACL. The other two, which represent two categories in our taxonomy (i.e., commissives and declaratives), are our contribution since these two types of communicative actions are absent in the FIPA specification. Therefore, we add these two new speech acts to the list of primitives acts in our library (SAL), which together with *inform*,*request*, *confirm* and *disconfirm* are used to composed the rest of them. We use Searle's taxonomy in the knowledge that there is little agreement on the number of speech acts and types which should be covered, or whether it is possible at all to provide a complete list of speech acts. Therefore, we remark that the list of speech acts presented here is regarded as "complete" only with respect to Searle's taxonomy. In any case, this partial list of actions cover the usual communicative requirements imposed on agents. Thus, we present a definition for all the four categories of speech acts illustrating at the same time how $MLTL_I$ allows us to naturally specify the preconditions and effects of the communicative actions. We will analyze in considerable detail the eight speech acts provided in table 5.2. These acts are representative enough to the FIPA specification and social approaches with our alternative formalization.

5.2.1 Assertives

Assertives perform statements about the real world. The typical assertive act is *inform*. This type of actions do not intend to modify the behaviour of the receiver, but only to affect its mental states. In particular, to modify the set of beliefs the receiver holds about a proposition ϕ. The definition of *inform* proposed by FIPA ACL indicates that the sending agent believes that some proposition ϕ is true, intends that the receiving agent also believes that ϕ is true, and does not already believe that the receiver has any knowledge of the truth of ϕ. This is regarding the Feasibility Preconditions. It also states that the Rational Effect is that the receiver comes to believe ϕ. In the formalization of this communicative act (see section 3.2.1) the Feasibility Precondition consists of a conjunction: The first conjunct states quite simply that agent i has to believe the proposition ϕ, and the second one states that the sender believes that the receiver does not have any knowledge of the truth of ϕ. This provided by the form $\neg B_i(Bif_j\phi \vee Uif_j\phi)$, which it is decomposed as $\neg B_i((B_j\phi \vee B_j\neg\phi) \vee (U_j\phi \vee U_j\neg\phi))$.

Leaving aside the fact that the specification does not provide the semantics of the modal operators used, using uncertainty and believe together to express that an agent does not know ϕ is very odd and unnecessarily complex. It is probably easier to either define a knowledge operator K_i with some $S5$ axiomatization and write $\neg K_j\phi$, to express that agent j does not know ϕ. If we do not want to use yet another modal operator, we could probably say that $\neg B_i(B_jEF\phi \vee B_jEF\neg\phi)$ to express that the sender does not believe that the receiver believes that there is a run in the system in which ϕ will eventually hold or that there is such a run in which ϕ does not hold.

In any case, we believe that specifying this precondition is asking too much of the sender. Intuitively, when you want to assert something about the world, one does not think whether the person is aware of the fact or not, or whether we believe that there is certainty or uncertainty. This does not affect the general idea that when you assert (*inform*) that ϕ, the sender usually believes that ϕ and has the goal of affecting the receiver's mental states so that it comes to believe ϕ. Therefore, our new definition of an *inform* looks as follows:

$\langle i, inform(j, \phi)\rangle$
$FP : B_i(\phi) \wedge G_i(B_j(\phi))$
$RE : B_j\phi$

Table 5.6. Inform.

The first part of the Feasibility Preconditions requires the sender to believe ϕ which means that we want the sender to be sincere. This is a good assumption by default, but if we want agents to be able to negotiate in competitive scenarios they may need to deceive. We believe that a feasible solution is to specify another speech act such as *convince* that could be used when an agent *just* aims that other agent believes a proposition ϕ, irrespective of the beliefs of the sender. This could give way to a trend of defining communicative actions useful to be use in argumentation and negotiation scenarios. This task should not be too difficult to carry out because we have now available a well defined specification language.

What about the Rational Effects? The FIPA specification says that whether or not the receiver adopts the belief in the proposition ϕ will be a function of the receiver's trust in the sincerity and reliability of the sender. That is all. FIPA does not provide a method to facilitate the achievement of the Rational Effects. Besides, it is quite clear that the nature of this observation about the receiver's trust in the sincerity of the sender, etc., points out to a number of factors that transcend the ACL semantics. It becomes quite clear that we may need to encode, in a specific scenario, the information relative to trust and other relations between the agents if we want to provide a method for agents to achieve the REs. This is the traditional role of pragmatics in natural language communication (as discussed in sections 2.2 and 2.3), and we claim that it is also the role that a pragmatic theory should play in agent communication.

Inform is the classic assertive speech act, but there are many others. For example, answers are generally assertives. Thus, speech acts such as *agree* and *refuse* are also assertives. Moreover, *confirm* and *disconfirm* are also assertives so we include their analysis in this section too.

According to FIPA ACL (2002), *agree* is a general-purpose agreement which answers a previously received *request*. When an agent agrees then it is informing the receiver that it intends to comply with the *request*, but not until the given precondition is true. *Agree* is not a primitive, so it is formalized in terms of an *inform*:

$$\begin{array}{l}
\langle i, agree(j, < i, act >, \phi) > \equiv \\
\langle i, inform(j, I_i Done(< i, act >, \phi)) > \\
FP : B_i\alpha \wedge \neg B_i(Bif_j\alpha \vee Uif_j\alpha) \\
RE : B_j\alpha
\end{array}$$

Note that the arguments of the agree performative consist of an action to be performed, *act*, and the conditions of the agreement ϕ. This in turn is analyzed as informing of the intention to do an action *act* under the condition ϕ. From our point of view, this is far too elaborated for an answer to a *request*. On the one hand, an agent is requested to do or achieve something, ϕ, for the sender. If you agree, you agree to achieve something ϕ. The conditions of the agreement are not specified in the semantics. We believe that the conditions of an agreement to comply with a particular request should be provided by a conversation policy, that is, in the pragmatics. In doing so, we have a well-formed semantics for *agree* and then different conversation policies and protocols which establish the conditions, agents' roles, etc., that are needed in a particular context.

Regarding the structure of the precondition itself, note that it has to hold for the sender to comply with the request and to do *act*. This particular point is not very clear in the formalization. We think that there may be a mismatch between the informal description of the act and the actual formal model. In any case, this type of construction is where $MLTL_I$ proves useful, because we can say that $I_i\phi\ U\ \psi$ that is, the sender intends to bring about ϕ until ψ along a run. More intuitively, if ψ is true, then $I_i\phi$ as long as ψ holds. This shows that if we want to include the idea of the agreement condition we can do it in a more natural way by using the temporal operator U (until). Where ψ describes the fact that constitutes the precondition of the agreement at a global state $r(m)$.

The second conjunct in the Feasibility Preconditions of *agree* presents the same form as in the *inform* act, so we will not repeat the point about the operators for uncertainty, knowledge and the over-specification of agents' behaviour in the ACL semantics. The same goes for the Rational Effects, although in this case there is an interesting note:

> "When the recipient of the agreement (for example, a contract manager) wants the agreed action to be performed, it should then bring about the precondition by performing the necessary communicative act. This mechanism can be used to ensure that the contractor defers performing the action until the manager is ready for the action to be done)." (FIPA ACL, 2002)

Due to the fact that most of the ACLs do not propose a complementary ACL pragmatics to the ACL semantics in the same framework, this type of contextual information is not dealt with. Our approach can indeed formulate this information in form of a normative conversation policy.

Following the above discussion, the formalization of *agree* tries to capture the intuition that agent i agrees with agent j to bring about some ϕ until some precondition ψ is true. This is equal to informing j that i has the intention that ϕ will eventually hold in a run until ψ holds. The FPs state that the sender has to intend that ϕ until ψ eventually holds along a run, and the REs establish that the receiver believes that the sender possess that intention.

$$\langle i, agree(j, \phi)\rangle \equiv$$
$$\langle i, inform(j, (I_i\phi\ U\ \psi))\rangle$$
$$FP : I_i\phi\ U\ \psi$$
$$RE : B_j(I_i\phi\ U\ \psi)$$

Table 5.7. Agree.

The dual of *agree* is *refuse*. Refuse is an answer to a request but negative in this case. At least in theory, because the FIPA specification does not show many similarities between the form of *agree* and *refuse*.

According to FIPA ACL (2002), *refuse* is the action of refusing to perform a given action, and explaining the reason for the refusal. Thus, the content arguments of the performative consist of the refused action and a proposition which provides an explanation for the refusal. Moreover, *refuse* is is an abbreviation for *disconfirm* that an act is possible for the agent to perform (and explaining why is that so). An agent considers that is not possible to perform an action when the preconditions of the action to be performed are not satisfied. As an example, an agent may be requested to perform an action for which it has insufficient privilege (hence the explanation: I have not got enough privileges).

This is how *refuse* is defined in the FIPA CAL:

$$\langle i, refuse(j, < i, act >, \phi)\rangle \equiv$$
$$\langle i, disconfirm(j, Feasible(< i, act >))\rangle;$$
$$\langle i, inform(j, \phi \wedge \neg Done(< i, act >) \wedge \neg I_i Done(< i, act >))\rangle$$
$$FP : B_i \neg Feasible(< i, act >) \wedge B_i(B_j Feasible(< i, act >) \vee$$
$$U_j Feasible(< i, act >))B_i\alpha \wedge \neg B_i(Bif_j\alpha \vee Uif_j\alpha)$$
$$RE : B_j \neg Feasible(< i, act >) \wedge B_j\alpha$$

We have a number of comments to this formalization of *refuse*. First and foremost, it does not seem very consistent to have two speech acts as possible answers to a request which in theory are the dual of each other but that, in fact, have nothing in common. Second, if *disconfirm* is a primitive then why does it seem to be analyzed in terms of another primitive, *inform*? Third, the use of operators such as Feasible with the purpose to provide reasons for refusing to do an action greatly complicates the logic. In fact, leaving aside the formalization of the operator itself, this is also part of the discussion about the semantic/pragmatic interface, because the FIPA specification tries to include *everything* in the semantics whereas we opt to capture the contextual information in the pragmatics. In this sense, when humans *refuse* to comply with a *request* and give a reason, there is always contextual information that serves to explain the refusal (such as one of the agents being a policeman, or a manager, or a technical support agent, etc.). The role of agents involved in the communication is important to understand the reason of the refusal. Besides, any other contextual reason, i.e., specific of a particular situation in which it is impossible for an agent to agree a request it is also provided by the information about the world that the receiver has available. In natural language, none of this information it is considered part of the semantics, which is concerned with the linguistic meaning of expressions, not with the various scenarios in which they can be used (Grice, 1975; Sacks, 1972; Searle, 1969). Using the same strategy for agent communication should have a well-defined and meaningful but also simple semantics which can be used in a variety of contexts. These contexts can then be specified in the pragmatics of the language. We will not repeat the arguments about the over-determination of agents' behaviour and the impossibility to guarantee the Rational Effects by means of an exclusively semantic specification.

Consequently, we consider refuse to be the dual of *agree* as a possible answer to a *request*. Moreover, and following FIPA's recommendation, it is analyzed in terms of the *inform* primitive, to communicate that the receiver of the request does not intend to bring about some ϕ (the object of the *request*) until ψ (the precondition of the agreement/refusal).

$\langle i, refuse(j, \phi)\rangle \equiv$
$\langle i, inform(j, \neg(I_i\phi\ U\ \psi))\rangle$
$FP : \neg(I_i\phi\ U\ \psi)$
$RE : B_j(\neg(I_i\phi\ U\ \psi))$

Table 5.8. Refuse.

Formally, the precondition to send a *refuse* states that sender does not intend, along a run, to eventually bring about ϕ until ψ, and the Rational Effects that the receiver believes that the sender does not intend to eventually bring about ϕ along a run (i.e., to fulfil the *request*) until ψ.

There are two more assertive speech acts to be analyzed: *confirm* and *disconfirm*. The above discussion with respect to *agree*, *refuse* and *inform* is also valid with respect to *confirm* and *disconfirm*.

In this section, we have not only specified the meaning of four assertive speech acts but also we have shown how $MLTL_I$ expressiveness matches well the requirements of defining such a high-level semantics. Moreover, along with the specification it has been pointed how some shortcomings of the FIPA specifi-

$\langle i, confirm(j, \phi)\rangle$	$\langle i, disconfirm(j, \phi)\rangle$
$FP : B_i(\phi) \wedge B_i(B_j F\phi \vee B_j F\neg\phi))$	$FP : B_i\neg\phi \wedge B_i(B_j\phi)$
$RE : B_j\phi$	$RE : B_j\neg\phi$

Table 5.9. Confirm and Disconfirm.

cation can be solved. In the next sections, we will define a speech act for each of the remaining categories: Directives, Commissives and Declaratives. The *complete* Speech Acts Library (SAL) can be found in Annex B.

5.2.2 Directives

The FIPA specification of the primitive *request* consists of a sender requesting the receiver to perform some action which can also be another speech act. The argument of the performative is the action that the receiver has to perform. It seems natural to think that one precondition would be that the receiver has the goal of achieving something for the sender. However, this basic aspect is not present in the FIPA definition (see table 5.2 above).

The are two classes of directives: Questions and requests. Both types share the basic feature of the sender holding a goal which can be achieved by the receiver performing a specific action. The goal of questions is to elicit some proposition from the receiver, which involves the receiver performing an answer. Requests have the goal of getting the receiver to perform some action.

If the receiver accepts the request, it will express the intention to execute the action requested. Having the intention of executing an action means that the agent will execute that action in order to achieve a specific goal. Note that by agreeing, the agent actually informs the receiver that it intends to perform a requested action. Conversely, if the receiver refuses the request, it will inform about its intention not to comply with it. The use of precommitments Fornara and Colombetti (2004) to analyze requests fails, in our view, to express that the sender explicitly states its interest of having the receiver executing a particular action.

Note, however, that we have not defined actions in $MLTL_I$. Instead, the labelling function is over atomic propositions ϕ which describe the state of affairs of the system at a global state $r(m)$. However, this responds to a simple interpretation of goals and intentions: Usually, when a *request* is made, the goal of the sender is for the system to reach a particular state of affairs, which in our case, means that we request that some proposition ϕ is true at some global state $r(m)$ of the multi-agent system. This interpretation in terms of the proposition that the sender wants the receiver to achieve, fits well with the intuition behind requests. This is also very similar to the intuitive meaning of goals and intentions in Rao and Georgeff (1991a).

Thus, in our approach, when sending a *request*, the sender holds the goal of the receiver achieving a particular proposition ϕ, that is, of making true ϕ at some global state $r(m)$. Moreover, since we want the receiver to *really* try to achieve ϕ the preconditions also require that the receiver intends along a run that ϕ be eventually true. Finally, the rational effect to be achieved is that there is a run in which ϕ eventually holds.

$\langle i, request(j, \phi)\rangle$
$FP : G_i(I_j F\phi))$
$RE : F\phi$

Table 5.10. Request.

5.2.3 Commissives

Surprisingly, FIPA does not include any commissive speech acts. The traditional example of a commitment is a *promise*. The sender expresses the commitment to perform the action expressed in the content of the commissive. Commissives commit the sender to perform the action uttered by the message. That is, by performing a *promise*, the sender states its intention to bring about some ϕ at some point in the system. In our approach agents *promise* to make eventually true some ϕ along a run. When sending a *promise* the sender has to have the intention of making ϕ true. The Rational Effects must be indeed ϕ is made true at some later point of a run.

$\langle i, promise(j, \phi)\rangle$
$FP : I_i F\phi$
$RE : F\phi$

Table 5.11. Promise.

5.2.4 Declaratives

Declaratives are not part of the FIPA CAL. Declarations have immediate effects in an extra-linguistic institution. They are the original *performative* verbs (Austin, 1962). Declarations are particularly useful for institutional actions. For example, speech acts to start or terminate an interaction (conversation) are declaratives. In that kind of situations, it is necessary to identify which agents are *allowed* to perform a specific declaration. Usually, agents have the right or the permission to perform a communicative act depending on their role in the particular scenario. In an auction, for instance, the auctioneer has the right to declare the beginning of an auction. An agent wishing to participate should be given the permission (by the auctioneer) to do so. An agent may perform an action for which it has not the right to. Again, all these points are to be included in the pragmatic component of the ACL to be presented in the next chapter. In the meantime we content ourselves with defining that when an agent *declares* that ϕ, it has the goal to make ϕ true in the next step of the run. The perlocution states that ϕ holds at the next step of the run. Note the use of the temporal operator X to express that in the immediate next step, ϕ holds along the run.

$\langle i, declare(j, \phi)\rangle$
$FP : G_i(X\phi)$
$RE : X\phi$

Table 5.12. Declare.

After defining the semantics of our unified ACL (consisting of the specification language $MLTL_I$ and the Speech Acts Library (SAL)), we show in the next chapter how a complementary pragmatics (NPRAG) provides the extra feature that an ACL needs for its use in a variety of scenarios and to facilitate the achievement of the Rational Effects of the speech acts. But before that, next section discusses the main features of the ACL semantics and where we stand now.

5.3 Discussion

The ACL semantics presented in this chapter consists of a logic $MLTL_I$ and a library of speech acts specified using $MLTL_I$. The work on $MLTL_I$ combines two different areas of logic for computer science: that represented by the research on reasoning about knowledge in distributed systems (Fagin et al., 1995; Halpern and Moses, 1992) and the research focused on temporal logics such as LTL for the specification of distributed systems (Manna and Pnueli, 1992).

By defining the semantics of the cognitive notions of beliefs, goals and intentions with respect to a computational model IS, we are effectively grounding the ACL semantics in the computational model. Cognitive states are determined by agents' local state, so that the accessibility relation between two local states are serial, transitive and euclidean for beliefs, and serial for goals and intentions. Most importantly, agents' cognitive states are *public*, and the system can, for example, be verified by checking the history of the interactions recorded in the global state.

By ascribing beliefs, goals and intentions to agents, we are able to predict and explain the behaviour of agents without having to take into account their internal structure or operation. Besides, by using Kripke structures to define the meaning of those operators makes it possible to generate soundness and completeness results for the axiomatizations of those logics. It also allows us to give a declarative specification of the system, instead of dictating how a specification should be satisfied by an implementation. A well-known model is the BDI model of Rao and Georgeff (1991a). However, there are usually some important problems regarding these modal logics. The semantics for beliefs, desires and intentions are given using a possible world semantics, for which there is not a clear relationship between the accessibility relations that characterize agents' attitudes and any specific computational model. This problem is also shared by some other logics such as the ones presented in van der Torre et al. (2004), and Cohen and Levesque (1997).

The introduction of a computational model in which the cognitive operators are grounded represents a clear advantage over the formalization of cognitive concepts for agent communication proposed by others like Singh (2000), whose social approach was the most promising of the semantic approaches discussed in section 3.2.5. Singh extends CTL with cognitive operators for beliefs, intentions and commitment. However, there are several differences with our approach. First, Singh does not provide the axiomatics for the logic, and in particular, it does not offer any axiomatics or discussion for the interaction between the temporal and the cognitive operators. Besides, simply by extending CTL with beliefs and intentions does not ground the semantic of those cognitive states in a computational model.

Being also a type of temporal logic, $MLTL_I$ has proved to be useful to express how a system evolves over time which allows us to define some aspects of the communicative acts. In fact, we believe that the Speech Acts Library developed in this thesis provides an intuitive and natural way of defining a complete catalogue of speech acts, and in providing an alternative formalization with the aim of overcoming the shortcomings of the FIPA proposal (see section 3.2.1 for details). In this sense, we have tried not to follow other approaches to agent communication such as the procedural and the social approaches which separate themselves from the FIPA standards. For example, Singh (2000) and Fornara and Colombetti (2004) present alternative speech acts specification within a social-based approach. Leaving aside the fact that concepts such as convention, power, obligation and commitment are not given a detailed definition, by implicitly including these concepts to define the semantics, and therefore, to determine the agents communicative behaviour, the semantics are not general enough to be applicable to different systems and scenarios. Furthermore, giving a semantics to every communicative act in terms of commitments sometimes produces odd results. For example, the definition of *request* (Singh, 2000) states that "the sender commits that the receiver has committed to accepting a request from him." This clearly states that the receiver has to previously agree with the sender that it will accept the *request*. It also ignores the intentional aspect of communication, in which by making a request, the communicative intention is that the sender has the goal that the receiver will do whatever it has been requested.

Besides, these approaches are semantic-based only, and as they wish to maintain agents' autonomy, they cannot account for the perlocutionary effects, that is, they do not offer a procedure to for the achievement of the perlocutionary effects of performing a communicative action (see 5.3 for more details). This point is explicitly acknowledged by Singh (2000), who says:

> "What we usually refer to informally as *meaning* is a combination of the semantics and the pragmatics. We will treat the semantics as the part of the meaning that is relatively fixed and minimal. Pragmatics is the component of meaning that is context-sensitive and depends on both the application and the social structure within which is applied. [...] Pragmatic claims would be based on considerations such as the Gricean maxims of manner, quality and quantity." Singh (2000)

In other words, Singh himself believes that a semantic specification is not enough, and that a pragmatic component should be included in an agent communication framework. However, none of the mentalistic, procedural or social proposals reviewed have provided a grounded minimal ACL semantics together with a complementary pragmatic theory which would constrain the use of the semantics to facilitate the achievement of the perlocutionary effects. Specifically, attempts based on Grice's theory of implicatures (Holmback et al., 1999) have not been successful (see section 3.3.3 for details on this approach).

This thesis provides an agent communication framework (see 4.5) which presents these characteristics. Our proposal should be seen as aiming to contribute to the standardization effort of FIPA. At the current state of our ACL specification, things stand as follows:

Requirements	**ACLs**	
	FIPA CAL	SAL
Autonomous	?	✓
Complete	-	✓
Contextual	-	-
Declarative	✓	✓
Formal	✓	✓
Grounded	-	✓
Public	-	✓
Perlocutionary	-	-

Table 5.13. Requirements for ACL semantics

- **Autonomous**: It means that the semantic specification does not violate the autonomy of agents. We assign an interrogation mark to FIPA here due to two reasons. On the one hand, sometimes the description of the FIPA communicative act does not correspond to its formal model. Usually, the formalization does not include every aspect of the informal description because that would constrain too much agents' behaviour. On the other hand, there are cases in which agents are still asked too much (see *inform* preconditions which are clearly too restrictive). Our approach does not place too many restrictions in the preconditions of the speech act because they are usually context-related, and consequently they belong to the ACL pragmatics.
- **Complete**: FIPA CAL does not include communicative acts for commissives or declaratives, which are needed in open multi-agent systems. The Speech Acts Library presented in this thesis does.
- **Contextual**: Neither FIPA CAL nor SAL fulfils this requirement. FIPA CAL sometimes tries to encode the context of use of the communicative acts

(see *refuse*, for example) but that makes the ACL semantics unnecessarily complex. Instead, we propose an ACL pragmatics to deal with the contextual aspects.

- **Declarative**: The meaning of the speech act in both specifications is declarative because it specifies what an speech means, and not how it should be used (as in the procedural approach to agent communication).
- **Formal**: SAL provides formal definitions for the meaning of the speech acts. FIPA does that as well, but operators such as Uncertainty and Feasible are not given a formal semantics.
- **Grounded**: The semantic specification language, FIPA SL, is not grounded in a computational model. As such, the semantics is not verifiable. On the other hand, SAL's specification language, $MLTL_I$, is grounded on an interpreted system IS.
- **Public**: To allow different types of verifiability (such as looking at the history of messages) we need the semantics not to depend on agents' internal states. Unfortunately, FIPA communicative acts do depend on agent internal states. We have overcome this problem by defining the mental attitudes from an external point of view using the notion of global state and defining beliefs, goals and intentions over runs.
- **Perlocutionary**: FIPA CAL and SAL do not guarantee the achievement of the Rational Effects. As it has been argued earlier, they should not. The achievement of the perlocution is helped by normative conversation policies defined as the pragmatics of the ACL (see next chapter).

As a final point, we could add that our speech acts definitions were more simple, which effectively facilitates the applicability of the semantics. This was possible mainly by using a temporal logic to express the evolution over time of the system. Besides, the agent communication framework provided allows any designer to define new speech acts according to their specific needs or aims. Simplicity and scalability are important features of our approach.

Next chapter develops the last component of the agent communication framework defined in section 4.5. We define a pragmatic specification language, $NLTL_I$, which formalizes a deontic operator. The formal machinery defined for the definition of $MLTL_I$ is inherited by $NLTL_I$. Once $NLTL_I$ is defined, we can use it to define the conversation policies and interaction protocols of which NPRAG is composed.

6

Normative Pragmatics for ACLs

In sections 4.1 and 3.2.5 we justified the need of introducing a pragmatic component to complement a minimal semantics in an agent communication framework. Some of these points were similar to those made in natural language by Grice, Sacks, Searle and other linguists which stressed that the nature of social and contextual aspects of linguistic communication are basically pragmatic (see 2.1, 2.2 and 2.3), and that the semantics is not sufficient to capture the full communicative meaning conveyed by the use of a speech act in a specific context. Other points made in those sections are more specific to agent communication, such as the need of policies to help agents in the intention recognition process and in the achievement of the perlocutionary effects with the point in mind that the aim here is not to design a system for natural language understanding but to propose a high-level ACL which can be used efficiently by agents. In relation to this, section 5.2 showed that a strictly semantic-based approach to agent communication is not enough to satisfy the requirements discussed in sections 3.2.5 and 3.3.7. Table 6 describes the properties of each of the semantic approaches (the social approach is evaluated on the proposal of Singh (2000), and the procedural on Greaves et al. (2000)).

Traditionally, the so-called ACL pragmatics usually consists of basic interaction protocols, that is, they simply establish the order in which speech acts were to be performed without any reference to the meaning of the speech acts used (see section 3.3 for more details on this point). Conversely, we believe that ACL pragmatics do have a say in the meaning of the linguistic expressions which depend on agents' communicative intentions and on the social context in which the conversation is taking place. Semantics does not fully determine the communicative meaning of an speech act because the uttering of a speech act may depend on contextual aspects such as the authority or trust of the agents involved in the

Requirements	**ACLs**		
	FIPA ACL	Procedural	Social
Autonomous	?	-	✓
Complete	-	✓	✓
Contextual	-	✓	-
Declarative	✓	-	✓
Formal	✓	-	✓
Grounded	-	-	-
Public	-	✓	✓
Perlocutionary	-	-	-

conversation. In this sense, we say that the ACL semantics is *underdetermined* and that pragmatics is required to fully determine the meaning of an speech act. Hence the name "unified ACL"; we claim that, in agent communication, the meaning of the speech act is to be fully specified by regulating the use of speech acts according to both the content and the scenario in which messages are going to be used. Having an *underdetermined* semantics does not mean that the semantics is ambiguous, it only means that the semantic specification cannot take into account every possible scenario, conversation, etc., which affects the meaning of the speech acts performed without loss of generality, and without violating agents' autonomy. We showed in the previous chapter that in order to satisfy the requirements to design a good ACL for open multi-agent systems, a well-defined, formal, and unambiguous semantics that can be used in many scenarios does not fully determine the meaning of the speech acts.

The ACL pragmatics (NPRAG) that we present in this chapter to tackle these problems is normative. We have been discussing why we need pragmatics in agent communication, but we still have not made an explicit point on why such pragmatics should be normative. Why not having a co-operative view of pragmatics (Grice, 1975), or a social-conventional view (Sacks, 1972), or indeed design simplistic protocols where messages' meaning depend on the order they are uttered as discussed in section 3.3.7?

Humans do (usually) understand each other by taking into account (unconsciously) contextual and intentional aspects about the speaker. There is a process of intention recognition which is, more often than not, successful. However, artificial agents obviously do not possess these natural abilities, so we may as well find an alternative way to guarantee that agent communication is successful. In other words, we need to bridge the gap between the semantic meaning of the linguistic expressions (message specification) and the communicative meaning (i.e., including the contextual implications) that the use of the messages can convey or that affect the meaning of a message. NPRAG contextually determine the linguistic meaning of the communicative acts and it will capture those social aspects (role, protocols, history of the conversation, etc.) that also influence the interaction. Crucially, unlike simplistic procedural protocols, NPRAG does not fully determine agents' behaviour, but its normative approach is based on specifying agents' freedom (rights).

There is an enormous amount of work done on normative multi-agent systems (Dignum and Kuiper, 1997; Esteva et al., 2001; Jones and Sergot, 1993; Meyer and Wieringa, 1993; Norman et al., 1998; van der Torre, 2003) traditionally related to specification of multi-agent systems using various types of deontic logic. Some of these approaches include a communicative module to model agents' allow a very restrictive interaction (Esteva et al., 2001), while others have tried to build commitment-based ACLs within an institutional framework (Fornara et al., 2004) (see section 3.2.4 for a criticism of the commitment-based ACLs). As far as we know, our approach is novel in using normative and organizational concepts to design an all-purpose unified ACL framework for agent communication, where the normative concepts are given a precise and formal definition. The basic concept of our normative pragmatic approach is the notion of 'right'. Note that we are not trying to investigate what the nature of rights are, or how many different types of rights can be distinguished or anything of the like (as discussed by Jones and Sergot (1993)). Instead, we give a formal definition of *right* which is convenient for our ACL framework, and that is the only meaning that 'right' would have in our system. We do not aim to develop a comprehensive theory on normative multi-agent systems, but to show how norms help agents in the intention recognition process that is communication relating the mental and the social aspects of communication. Having said that, our unified ACL can be adopted on top of a normative multi-agent system. In fact, although we think of NPRAG and SAL as a unified ACL, the pragmatic component can in theory be adopted to regulate the use of a different semantic theory.

The normative pragmatics presented in this chapter regulates the application of the semantics within a social structure. The social structure will take into account organizational concepts such as the roles of the agents (participants) in the conversation, their obligations with respect to other agents and the rights they hold (the general social structure considered here is inspired by Ferber and Gutknecht (1998); van der Torre (2003); van der Torre et al. (2004)). Moreover, in order to enforce the conversation policies, notions such as sanction will be defined. Specifically, NPRAG consists of interaction protocols which encode convention protocols such as Request and Query-if, and of conversation policies which take care of how context affects the meaning of messages.

The chapter is structured as follows: First, in section 6.1 we discuss the advantages of specifying norms for multi-agent systems. Second, we define an organizational structure from which several key concepts are used in the definition of $NLTL_I$ (section 6.2). Central to it is the notion of *right* defined in section 6.4.2. Once this is done, we propose a variant of $MLTL_I$ in which instead of cognitive operators we introduce a deontic operator grounded in the interpreted system IS of section 4.4. As with $MLTL_I$, we provide a syntax, semantics and discuss the properties of the logic provided by the axiomatics (section 6.4). $NLTL_I$ is then used in 6.4.2 to provide a formal definition of the organizational concepts considered earlier.

We then distinguish between conversational policies and interaction protocols, and how these can be understood as declarative rules expressing the rights, obligations and permissions of agents when involved in conversation (see section 6.5). In order to do that, we translate $NLTL_I$ into finite state automata, which are then coded in Prolog as Definite Clause Grammars (DCG) to represent conversation protocol. This gives us a direct relation between the logic used to reason about normative and communicative agents, $NLTL_I$, and allows us to use type of systems that are both easy to implement and verify.

Furthermore, we show how the pragmatics helps agents to achieve their perlocutionary effects, and we represent some FIPA Interaction Protocols in NPRAG. We finish the chapter with a discussion on other work and we draw some conclusions.

6.1 Normative Multi-Agent Systems

Normative systems have largely been studied by legal philosophers (Alchourron and Bulygin, 1971). Meyer and Wieringa (1993) state that in normative systems in which norms play an important role and where normative concepts are needed for them to be described or specified. They also claimed that deontic logic is an adequate tool to formalize such systems. Using deontic logic it is possible to specify not only the legal behaviour, but also the illegal behaviour which was usually ruled out of the system specification. Deontic logic introduces an operator for obligation, meaning "it is obligatory to see to it that x", and its dual, permission. These two operators were supposed to be the analogous of necessity and possibility from alethic logic, with the exception that the later are characterized by a $KD45$ logic and deontic modalities traditionally take KD.

Human societies are governed (partially) by norms. In the study of the role of norms in societies, Habermas (1984) distinguished four types of sociological action models which influenced various approaches to agent communication (see, for example, Singh (2000)). One is the *rational choice model*, in which agents are goal directed and try to maximize their choice of means to obtain a goal. In this model, agents act according to their beliefs about the existing state of affairs and their intentions to bring about desired state of affairs in the world. The second action model is the *normative action model*. In this model, agents are members of groups and follow a set of norms. Agents are supposed not to break the rules, and they act with respect to a normative context which defines the possible interactions between the agents. The third model is the *dramaturgical action model*. The central concept in this model is the presentation of the self, which is defined as a collection of beliefs, desires, intentions and needs. This is considered the subjective point of view of the agent. Finally, there is the *communicative action model*. This model consists of the three functions of language described by the previous action models (see Boella et al. (2005) for a good introduction to the concept of normative multi-agent systems). Thus, agents use the language to achieve some goals, they use language to create or update current norms and to express their beliefs, intentions and desires. Singh (2000) uses these three uses as *validity claims* to construct an ACL in which the objective, social and mental realms are present in the language. Our approach also takes into account these ideas. Thus, the pragmatic component of the unified ACL corresponds to the social world, whereas the other two are represented in the semantic specification of speech acts (SAL) given in the previous chapter. Our approach differs from Singh's because our ACL semantics stresses the teleological aspect of communication, whereas in Singh's approach *commitment* is the central aspect (as discussed in section 3.2.4).

The ACL semantics defined in the previous chapter considered cognitive states of individual agents which were either preconditions (on the sender) or rational effects (on the receiver generally). Moreover, communication is also a *social activity* that involves an intention recognition process. However, the mere representation of the cognitive states that some agent intended to achieve does not give a method to actually achieve them. This is the basic problem of specifying the rational effects, as we have mentioned in several occasions. This, among other reasons (see 5.3), points out to the use of social concepts in agent communication if we want to be able to characterize communicative behaviour not only for individual agents, but for multi-agent systems.

Norms have been proposed in agent theory to link the individual agents' cognitive states with their social behaviour in multi-agent systems. Using the motivational notion of *obligation* to specify normative agents serves well this purpose because social notions such as coordination can be expressed in terms of obligations, permissions, etc. Besides, it has been argued that agents should (in principle) be able to violate the norms (agents are seen as autonomous). This is important in applications such as e-commerce, where it has been shown that systems are more efficient when agreements can be broken.

The two main issues discussed here, that norms establish a link between individual agents and social behaviour, and that norms can be violated, will be adopted in our normative pragmatics approach. But first, we will introduce basic organizational concepts needed to structure NPRAG.

6.2 Organizational Concepts

The concepts of role, group, institution, etc., are key concepts in sociology to the understanding of human societies and to the study of human linguistic communication. Several theories incorporate these concepts in order to design and specify normative multi-agent systems. Besides, the concept of role has been widely used in Object Oriented Programming. Many of these theories use the concept of role to define agents' behaviour within a society. This allows us to specify the behaviour of an agent regardless of its internal structure. Thus, organizational concepts are specially interesting for the design of open multi-agent systems, where the internals of agents are often unknown. This is due to the fact that norms describe the social structure of the system. In this section we introduce several concepts that will be used for the specification of the normative ACL pragmatics. We first discuss the idea of electronic institutions and then we introduce the idea of group, role and role relations (section 6.2.2).

6.2.1 Institutions

Electronic institutions as a tool to model e-commerce have been developed by Rodriguez-Aguilar et al. (1997); Esteva et al. (2001). The institution provides the social rules that regulate the interaction between agents. The activities carried out within the institution are located in *scenes*. The behaviour of an agent in a scene is defined by the role and responsibility of the participant. The general patterns of behaviour are defined in terms of a set of speech acts, which in their proposal is called the *dialogic framework*. The roles defined within the institution are ordered hierarchically. The set is represented as a pair $\mathcal{R} = \langle Roles, \preceq \rangle$ reflecting a role hierarchy. The relation $\preceq \subseteq Roles \times Roles$ is reflexive, antisymmetric and transitive. According to this, if $r \preceq r'$ holds, then we say that r subsumes r', meaning that an agent playing role r is also enabled to play role r'. Besides, roles can be conflicting. For example, in an auction scenario, no agent can act as the auctioneer and as a bidder. A policy of static separation of duty is defined to mean that roles specified as mutually exclusive cannot be both authorized to an agent. The aim is to protect the institution against malicious behaviour. This requirement is specified as the relation $ssd \subseteq Roles \times Roles$. A pair $(r, r') \in ssd$ denotes that r, r' cannot be authorized to the same agent.

The conversations that take place in a scene are specified by a protocol which restrict the speech acts that can be performed by different roles. Besides, the number of participants can change, so the protocol also specifies the condition under which an agent can enter or leave a conversation. The concept of dialogic framework specifies the set of messages and the ontology to be used by the agents. The dialogic framework is defined as a tuple $DF = \langle O, L, I, CL, Time \rangle$ where O stands for an ontology (vocabulary). CL defines the communication language and L the content language. Following speech acts theory, I represents a set of illocutionary particles. CL expressions are constructed as a formula of the type $\iota(\alpha_i : \rho_i, \alpha_j : \rho_j, \varphi, \tau)$ where $\iota \in I$, sent in the instant τ by an agent α_i to a receiver α_j, with a content $\varphi \in L$. The main difference with FIPA ACL is that the role of the agents, ρ_i, ρ_j are explicitly included in CL.

A performative structure is defined to specify relations between the different scenes which are part of the institution. Thus, agents interactions go from scene to scene, according to the restrictions that the performative structure define. Note that it is not the purpose of Esteva et al. (2001) to specify a standard ACL, but they create a performative module that is useful for agents to communicate in their institution (only).

Another approach to institutions comes from Colombetti et al. (2002), which was already discussed in chapter 2. They propose a commitment-based ACL from an institutional point of view. An institution provides the social context in which a group of agents interact. The behaviour of the agents is defined by roles. Besides, the set of actions which agents can perform are specified by a set of authorizations, and the restrictions on the actions are associated to a particular role. The institution is completed by an ontology and the inscription rules. The ontology states the institutional facts which allow to specify the actions for the roles of the institution. The inscription rules state how agents can become members of the institution playing a particular role.

Colombetti et al. (2002) also order the roles hierarchically. However, the relation is between roles of different institutions. Thus, if r_1 of I_1 subsumes r_2 of I_2, the authorizations, the ontology and the interaction rules are imported by I_1. There is a basic institution which is the core of every interaction context. The Core Institution defines one role, *speaker*, and a set of basic communicative actions: inform, request, query-ref, query-if, accept and refuse. For every role of every institution to communicate, they subsume the speaker's role. This gives them authorization to perform any of the basic set of communicative actions. Thus, the ACL is defined in terms of the institution and its roles. Note that this approach does not present a pragmatics of agent communication that contextually affects and constrains the meaning of the speech acts. Conversely, the meaning of the commitment-based speech acts stays unchanged, and they propose some protocols based on interaction diagrams which merely specify the order in which speech acts are to be performed (see section 3.3.7 for a detailed discussion on this issue).

In our view, by avoiding the use of cognitive notions such as goals and intentions, the approach presented by Colombetti et al. (2002) does not take into account the intentional character of agent communication. Although the organizational models are indeed important in order to facilitate interoperability, it is also true that communication is a goal-based action. By ignoring the motivational aspect of communication, the resultant definitions of the communicative acts are somewhat peculiar (see sections 3.3.7 and 5.3). In any case, we will show (see below) how notions such as institutions and roles will prove useful to develop the normative ACL pragmatics.

6.2.2 Groups and Roles

One of the first approaches that used organizational concepts to model multi-agent systems comes from Ferber and Gutknecht (1998). They take into account the structural level of the system instead of the internal architecture of the agents. They developed a new model *Agent/Group/Role* and a development platform MADKIT, which implements the Agent/Group/Role model. Figure 6.1 shows the model which describes the relations between group structures, organization structures and agent classes, which they use to define groups, organizations and agents respectively. A group is defined in terms of a set of roles, a set of role interactions, and an interaction language.

The model allows the description of different types of organizations. A role represents the functionality of an agent as member of a group. Agents are defined functionally, i.e., as communicative entities that play roles within groups. Agents can be members of different groups in which they can play more than one role.

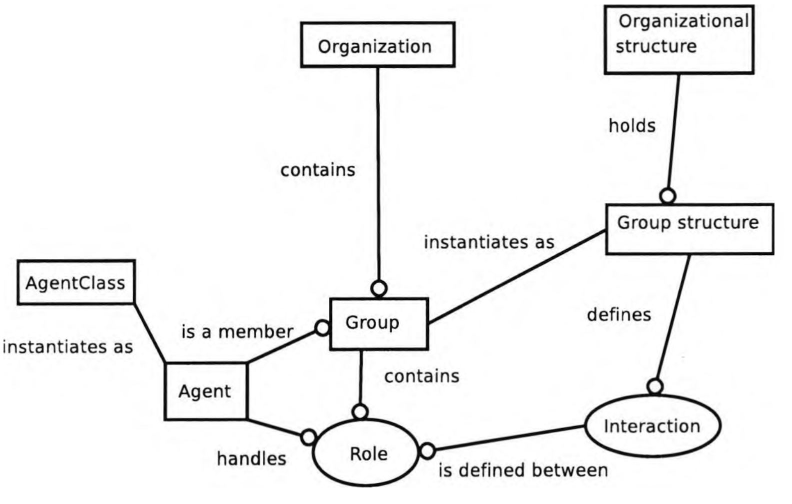

Fig. 6.1. The Agent/Group/Role model.

The Agent/Group/Role model is still a valid abstraction to organize a multi-agent system, and it will be used in the remainder of the chapter. Thus, the basic ideas about groups, roles and role relations are adopted for our purposes (see Ferber and Gutknecht (1998); van der Torre et al. (2004) for more details). Summarizing,

- Informally, a **role** is understood as a set of constraints that are to be satisfied when an agent plays that role. Thus, the role of auctioneer establishes a set of conditions/constraints which are defined in terms of obligations, permissions and rights of the agent playing the role. The definition of a role depends always on a institutional activity. Thus, the role of bidder only makes sense during an auction.
- **Groups** can be described as sets of roles that share a group characteristic. For example, in e-commerce, the roles in charge of selling on behalf of a company are part of the selling department (group).

- **Role relations** are generally understood as restrictions on the relations between roles. Role relations coordinate the behaviour of the agents playing the roles.

Note that when agents communicate, they are always playing a role. This may seem obvious, but it is only very recently that roles have been considered when specifying agent communication languages (Fornara et al., 2004). In most of the human linguistic interactions roles are crucial for the success of communication. For example, Searle distinguished between the speech act "The lecture is over" performed by a lecturer in front of his class and the same sentence uttered by a student. In this example, the role of the agent that performs the speech act crucially determines its success. Roles therefore should play a significant part in the social component of our unified ACL, but roles should not fully determine agents communicative behaviour, that is, an agent could have the right to challenge a boss's request, for example.

6.3 Norms in Agent Communication

So, the big question can be formulated as follows: What are the benefits of using norms in agent communication pragmatics?

The following example will illustrate the basic problem of a semantic-based approach to agent communication: Following a semantic-based approach, our specification of *inform* in SAL states that when an agent i performs this act:

i It believes its propositional content ϕ and
ii It has the goal that the receiver j will eventually come to believe that ϕ holds.
iii The perlocution is that j comes eventually to believe that ϕ holds.

The process of intention recognition would presumably be described as follows: When agent j receives the message, it will assume that the first two preconditions hold. As a consequence, j should believe that i believes that ϕ, if j trusts the sender's message, j will believe ϕ, which corresponds to communicative goal i wanted to achieve. Assuming that agents do this process is, however, too idealistic. Moreover, it is computationally expensive to let agents do all this reasoning. While this is an interesting problem for computational linguistics, an agent communication language should allow agents to communicate with each other effectively and efficiently in open environments, where it could be the case that i is not entirely honest about ϕ, for example, in a competitive scenario.

This is where interaction protocols come to work. Interaction protocols define the sequences in which speech acts can be performed, so that agents can follow the conversational pattern without doing all this complex reasoning pictured above. As it has been argued in section 3.3.7, several accounts have been proposed to deal with this problem. Although interaction protocols are necessary for agent communication, most of them restricted agent conversation to a "follow-the-rule" activity, in which agents are not autonomous any more and where a conversation simply becomes and exchange of meaningless tokens. These approaches do not consider how the use of messages can be regulated according to their content and meaning in specific contexts. This is due to the fact that none of these approaches consider both aspects, semantic and pragmatic, as the two sides of communicative meaning, according to which speech acts acquire their full communicative meaning when pragmatics contributes the relevant contextual information that is, agents' roles, background information, record of the conversation exchange so far, etc.

The basic intuition behind the normative pragmatics presented in this thesis is that the intention recognition process described above can be regulated by means of rights, obligations and permissions. In fact, when we were talking about the process itself we were saying how agents *should* believe, and which beliefs *should* they adopt. If we could make policies to take into account contextual information to regulate the use of the semantics, we do not need agents to do all that complex mental reasoning. Note that this is coherent with the *external* point of view adopted to formalize cognitive concepts in $MLTL_I$.

The FIPA CAL specification itself provides another good example of why a normative pragmatics may be useful to regulate the use of the speech acts. In the definition of the *agree* communicative act, there is a *pragmatic note* that reads:

> "The precondition on the action being agreed to can include the perlocutionary effect of some other CA, such as an *inform* act. When the recipient of the agreement (for example, a contract manager) wants the agreed action to be performed, it should then bring about the precondition by performing the necessary CA. This mechanism can be used to ensure that the contractor defers performing the action until the manager is ready for the action to be done". (FIPA ACL, 2002, p.4)

There are few other remarks like this one throughout the ACL semantics of FIPA. In our view, it points out to the need of somehow regulate the use of the communicative acts, but the FIPA specification does not go further. Leaving aside the particularities of the preconditions stated by the FIPA specification, this note illustrates the valuable role that a normative pragmatic theory can play. First, it states that agents play a specific role in the interaction. Second, it prescribes the behaviour of the agents in a specific context and even the timing of executing a particular action. Furthermore, the fact that this note is part of the ACL semantics suggests that on the one hand, there is a suspicion that something is missing in the ACL specification and, on the other hand, that whatever is missing cannot go into the ACL semantics without violating agents' autonomy. As far as we know, *none* of the various approaches to agent communication reviewed have tried to tackle the problem posed by this remark. This is the main reason to propose a Normative Pragmatics theory (NPRAG) for agent communication. Proposing and formalizing *rights* to do so is a contribution of this thesis. Next section distinguished interaction protocols from conversation policies to model agent conversations.

6.3.1 Protocols and Policies

We have argued that many of the interaction protocol approaches developed so far provide a low-level procedural characterization of interactions. Representations based on monolithic finite-state diagrams are suitable only for the most trivial scenarios. Still, protocols play a central role in agent communication. Interaction protocols are efficient using institutional contexts to model turn-taking strategies. Interaction protocols dictate which sequence of messages is appropriate for specific situations. For example, in auctions, turn-taking might underlie the specific rules to ensure that they are created only when they make sense. Thus, a bidder should not make a bid prior to the advertisement.

However, this is not the whole story. The institutional interactions created by an Interaction Protocol such as an English Auction can be seen as the *constitutive rules* according to which communication takes place. Constitutive rules only establish the order in which speech acts are to be performed. However, this does not fully account for the kind of *pragmatic* constraint we need to consider for situations such as the one described above. In other words, interaction protocols do not *regulate* or modify the use of the speech acts according to their content. In order to do so, we need *regulative rules* that specify agents' rights, obligations and permissions for specific conversational contexts. This distinction between *constitutive* and *regulative* rules in communication is due to Searle (1969, 1995).

> "Some rules regulate antecedently existing forms of behaviour. For example, the rules of polite table behaviour regulate eating, but eating exists independently of these rules. Some rules, on the other hand, do not merely regulate an antecedently existing activity called playing chess; they, as it were, create the possibility of or define the activity.[...] The institutions of marriage, money, and promising are like the institutions of baseball and chess in that they are systems of such constitutive rules or conventions" (Searle, 1969, p.131)

We say that the *constitutive rules* that define the English Auction interaction protocol conform the institution. Therefore, institutional speech acts are those whose meaning depend on the institution in which they are used. The pragmatics of institutions are thus taken into account by a library of normative interaction protocols. Moreover, *regulative* rules in agent communication deal with context-dependent aspects: Level of trust between agents, and other particularities brought about by the speaker, for example. We can for example make a *politeness* rule by which we can say that agents have the obligation to always answer (either positively or negatively) to a request of an agent from a company X regarding a subject Y. In our framework, regulative rules are expressed by normative *conversation policies*, and they try to capture the social aspects of communication as expressed in the FIPA's pragmatic note example discussed above. Additionally, *conversation policies* aim to facilitate the achievement of the perlocutionary effects of the speech acts. Conversation policies can also affect the meaning of speech acts in institutions because the object of the rule can refer to an institutional fact. Note that the distinction between interaction protocols and policies is not new, but it is present in other approaches under different names (Flores and Kremer, 2002; Phillips and Link, 1999). However, NPRAG is the first normative approach to ACL pragmatics that relates them to constitutive and regulative rules of the dialogue. Besides, our approach claims that the conversation policies enrich the semantic meaning of messages and facilitate the achievement of the rational effects.

According to the characterization of pragmatics of section 4.1, NPRAG deals with the effect that the following issues have on the sender's choice of expression and the receiver's interpretation of an utterance:

- **Context of utterance**: Conversation policies state the relation between participants' roles and any particular contextual information (politeness, etc.) specific of the scenario.

- **Perlocutionary Acts**: NPRAG specifies policies about agents' communicative behaviour for a given speech act.
- Participants' methods of **turn-taking**, constructing sequences of messages across turns, and how conversation works in different conventional settings are mainly dealt with the constitutive rules of the theory.

When we say that communication is a social activity, that does not only mean that it is a process that takes place within a group of agents. It is also important to stress that communication involves a number of social structures which specify and regulate the social component created by communicative actions. Thus, an important function of agent communication is to create (give, modify) rights, permissions and obligations involving the sender, the receiver and possibly third parties that belong to the scenario where the message interchange is taking place.

6.3.2 Rights in Agent Communication

The central concept of our normative approach to develop agent communication pragmatics is the concept of *right*. The main reason for using rights in our pragmatics is to give agents enough freedom, but also limit agent's behaviour. We believe that there is a middle ground between traditional obligations and permissions as defined in standard deontic logic, and that the concept of right which we define below is appropriate to capture that middle ground. We do not follow any definition of right in the literature because its definition depends on the logic used, nor we try to provide a solution for any possible ambiguity that could be found in the notion of right, that is, in the fact that right has been usually used to refer to various things. For example, having the right to live, the right to work, a right to feel proud, a right to make pre-emptive attacks, a right to vote, etc. In this sense, rights can be classified as liberties, privileges, claims, power, active, passive, etc.

We do not intend to define a notion of right to capture all those different meanings nor we will try to classify or discuss them. Instead, we define a formal concept of right which is to be applied in a normative approach to agent communication according to the intuition that there is some middle ground between pure obligations and permissions and that defining a concept to express that will provide a normative notion which helps to coordinate agent communication without completely pre-determining agents' behaviour. This concept is in some sense close to what Castelfranchi (1997) calls *strong permission*. A general idea of right is provided by the following characterization from (Wenar, 2005, p.1):

> "Rights dominate most modern understandings of what actions are proper and which institutions are just. Rights structure the forms of our governments, the contents of our laws, and the shape of morality as we perceive it. To accept a set of rights is to approve a distribution of freedom and authority, and so to endorse a certain view of what may, must, and must not be done."

Using our framework to express how the system evolves over time extended with a deontic operator will prove useful to capture how the obligations and rights of agents change as the system circumstances change. Besides, the need of temporality when modelling normative systems has been defended by other authors (Dignum and Kuiper, 1997; van der Torre et al., 2004).

Norman et al. (1998) use dynamic logic to formalize a notion of right (which resemble traditional permissions) to model agreements. Alonso (2004) adapts the proposal of Norman et al. (1998) to formalize a notion of right which, unlike obligations and permissions, is universal. Alonso (2004) claims that economic-based theories of rational choice, such as game theory, cannot provide a satisfactory explanation of co-operation and collective action. The reason is that in game theory, agents calculate individually their best choice. Communication does not help either, because agents do not trust each other, and will not respect any commitment. Games with multiple equilibria or with no equilibria at all also pose problems. In particular, it is not possible to reach a rational decision about the agreements agents should make. To solve this, either *ad hoc* solutions or *local points* are proposed. We generally agree with the intuitions behind these claims, and leaving aside other formal issues, the notion of right formalized by Alonso (2004) is based on dynamic logic using possible world semantics. Boella and van der Torre (2005) propose that rights are sets of strategies of agents' roles. Their proposal is interesting because they argue that rights are exercised by roles, but in our view it is not clear how their idea of right is different from the set of choices that agents have available, or the set of permissions that can be specified for a specific role.

An interesting point in the etymological meaning of the word 'right' comes from that *that is fair* or *just*. This sense allows to talk about a society that is "rightly ordered", for example. When applied to individuals, rights entitle their holders to some *freedom*. For example, an agent can be entitled with the power or privilege to act in certain ways. Both senses, although different, are important for us since they are related to this idea of talking about agents' freedom to follow a conversation policy or to violate it. At the same time *rights* are not merely seen as the absence of obligations.

In our approach, if an agent has the right to perform a speech act, then:

- It is permitted to perform it (under certain obligations), since it does not constitute a violation.
- The rest of the agents are not allowed to perform any action that violates a right-holder's action, otherwise, they are sanctioned.
- The normative system, the group, which is represented by a special type of agent, has the obligation to sanction any violation.

The function of norms in agent communication is to stabilize social interactions by making the behaviour of agents predictable to the other agents of the system. Permissions are usually defined as the dual of obligation, meaning that an agent that is not obliged not to bring about ϕ is permitted to bring about ϕ. Having the right to perform an speech act means that the agent must be given permission to do so and that should not be a violation. In this sense, not being obliged not to do α $(\neg O \neg \alpha)$ does not mean that the agent has the right to do α $(R\alpha)$.

The description of agent's rights and obligations can be stored and accessed by every agent at any time, so that the ACL pragmatics is public. In this sense, an agent may not know whether another agent is sincere, but it can know which rights and obligations the other agent should abide to.

In the next section we present the pragmatic specification language $NLTL_I$. $NLTL_I$ shares the same characteristics as $MLTL_I$ but instead of containing cognitive operators we define a deontic operator. Once the syntax, semantics and axiomatics of $NLTL_I$ are presented, we define the notions of violation, right and sanction, which are also to be used in the development of the interaction protocols and conversation policies that conform NPRAG.

6.4 $NLTL_I$

The normative temporal logic $NLTL_I$ we define in this section follows the general structure of $MLTL_I$ in the previous chapter. The main difference is that while $MLTL_I$ was designed to express agents informational and deliberative states to specify an ACL semantics from an intentional point of view, $NLTL_I$ includes the temporal branching operator but combined with a deontic operator.

$NLTL_I$ structures are also defined by associating structures with deontic accessibility relations to the computational model IS defined in section 4.4. We will refer quite often to the definition of run, global state, point, interpreted system of section 4.3, and to the methodology of associating Kripke structures to interpreted systems of section 5.1.3. Finally, the syntax of the temporal operators is the same as in section 5.1.6.

We therefore carry on with the idea of grounding the semantics of our ACL specification language in the interpreted system IS. Traditionally, the deontic logics defined to model normative multi-agent systems (Dignum and Kuiper, 1997; Norman et al., 1998; Alonso, 2004; van der Torre et al., 2004; Meyer and Wieringa, 1993) whether they are based on Standard Deontic Logic or not (von Wright, 1951), have their semantics based on possible worlds. As we argued in the previous chapter, there is not a clear relation between the possible world semantics and computer programs. Furthermore, some of those logics are not easily decidable because their complexity is greatly increased by the combination of deontic, dynamic and temporal operators.

However, there is a recent approach to deontic logic which actually offers a grounded semantics (Lomuscio and Sergot, 2003) for Deontic Interpreted Systems. In short, Deontic Interpreted Systems consist of a static interpreted system of global states where there are two types of global states, those that are allowed and disallowed states of the computation. The interpreted system presented by Lomuscio and Sergot is *static* because they do not include the notion of run which provides the temporal component in the original interpreted systems model. They give a special $KD45_n$ axiomatization for the deontic operator and claim that future work includes incorporating a temporal component into the logic. In this sense, $NLTL_I$ differs from the Deontic Interpreted Systems in various ways. First, we define $NLTL_I$ with respect to a interpreted system which is adapted to model agent communication and in which the global states of the system are not required to be exclusively deontic. For example, we assume that information about the history of conversation, social structure, institutional facts, etc., could be encoded in the environment's state, whereas the obligations, rights, etc. of agents are to be kept in agents' local states. Second, we include a linear time component in our logic to capture the evolution of the system over time. Moreover, the temporal operators also gives us extra expressiveness to talk about sanctions, rights, and other normative notions of our framework.

6.4.1 $NLTL_I$ Syntax

We need to express obligations and rights within an organizational structure in which agents have roles assigned. Rights, Violation and Sanction are not defined as primitives. The only deontic primitive operator of our framework is obligation, denoted by O_i. Following the definition of the cognitive operators in the previous chapter, we will accommodate the interpretation of the primitive deontic operator for its use with respect to runs in an interpreted system.

Regarding, roles, we use the following notation:

- $i\ rr\ j$, means that i and j are role-related by rr.
- i is a member of group c, is expressed by c_i.
- r_i denotes that i plays the role r.

A role is a set of constraints that should be satisfied when an agent plays the role. For example, the role of auctioneer constrains obligations, permissions and rights of the agent that plays that role. The scope of the role depends on the institutional reality in which it is defined (e.g., auction). A group is a set of agents (roles) that share a specific feature (i.e., being auctioneers). Finally, role relations constrain the relations between roles (e.g., the auctioneer-bidder relation).

The syntax of $NLTL_I$ associated to the interpreted system IS consists of the vocabulary of the interpreted system IS introduced in 4.4 which is extended with temporal operators and the deontic accessibility relation. $NLTL_I$ structures are actually the result of the combination of IS with the accessibility relations $\mathcal{O}_i$ of a Kripke structure M.

Definition 27 **($NLTL_I$ Syntax).**

Given a finite set of agents $i = (1, \ldots, n)$, a finite set of group names CN, a finite set RN of role names, a finite set RR of role relations, and a countable set AP of primitive propositions, the syntax is defined as follows:

C1 If ϕ is an atomic proposition of AP then ϕ is a $NLTL_I$ formula.
C2 If ϕ and ψ are $NLTL_I$ formulae, then so are $\neg\phi$ and $\phi \wedge \psi$.
C3 If ϕ is a $NLTL_I$ formula then $O_i\phi$ is also a $NLTL_I$ formula.
C4 If ϕ is a $NLTL_I$ formula then so are $X\phi$, $F\phi$, $G\phi$ and $\phi\ U\ \phi$.

The notation used here is the same as in $MLTL_I$ (see section 5.1.6). Regarding the deontic operator $O_i\phi$, the traditional reading has been something like "agent i is obliged to bring about ϕ", or "agent i ought to bring about ϕ", or "agent i must bring about ϕ". In our case, any of these interpretations are fine, although it is also interesting the interpretation proposed by Lomuscio and Sergot (2003) where the operator $O_i\phi$ expresses the idea that "if agent i is functioning correctly, then ϕ holds", where ϕ can refer to global or local states in the system. This second interpretation also fits well with $NLTL_I$ because their semantics are given in terms of interpreted systems. In this sense, and given that we introduce time in our models, we propose that $O_i\phi$ expresses that "the system is at a point in which ϕ holds if agent i works (acts) correctly", which is very similar to the formulation used for the cognitive concepts of $MLTL_I$.

In any case, we define $P_i\phi$ as the dual of $O_i\phi$ such that

$$P_i\phi = \neg O_i\neg\phi$$

Which we could gloss as meaning that "agent i is permitted to bring about ϕ" or, closer to interpreted systems, it may mean that "the system could be at a point in which $\neg\phi$ holds if agent i is not working (acting) correctly".

6.4.2 $NLTL_I$ Semantics

$NLTL_I$ structures are generated by grounding a deontic Kripke structure M into the interpreted system IS. For full details of some semantic properties, the read should backtrack to section 5.1.7 for issues such as validity, satisfaction, and some general points on grounding the logic in a computational model that we do not reproduce here to avoid repetition. Every definition of that section except those directly related to the cognitive operators apply here.

Definition 28 (Deontic Kripke Structure).

A Deontic Kripke structure $M = (S, \mathcal{O}_i, \ldots, \mathcal{O}_n, L)$ is serial if for any accessibility relation $\mathcal{O}_i$ we have that for all s there is a t such that $(s,t) \in \mathcal{O}_i$.

From the Deontic structure M and IS we generate $NLTL_I$ structures:

Definition 29 ($NLTL_I$ structure).

Given a system of runs T, $NLTL_I$ is generated by associating the interpreted system $IS = (T, L)$ with the serial Kripke structure $M = (S, \mathcal{O}_i, L)$, such that $NLTL_I = (GS, \mathcal{O}_i, L)$ where:

- *GS corresponds to the sets of global states in IS.*
- *L is a labelling function $L : S \rightarrow 2^{AP}$ from global states to truth values, where AP is a set of atomic propositions. This function assign truth values to the primitive propositions AP at each global state in GS.*

- $\mathcal{O}_i$ *where* $i = (1, \ldots, n)$ *is a set of agents, gives the accessibility relation on global states, which is serial. Thus, we have that* $(l_e, l_1, \ldots, l_n)$ $\mathcal{O}_i$ $(l'_e, l'_1, \ldots, l'_n)$ *if* $l'_i \in GS_i$. *If* $g = (l_e, l_1, \ldots, l_n)$, $g' = (l'_e, l'_1, \ldots, l'_n)$, *and* l_i $\mathcal{O}_i$ l'_i, *then we say that* g *and* g' *are* $\mathcal{O}_i$*-accessible to agent* i. *The formula* $O_i\phi$ *is defined to be true at* g *exactly if* ϕ *is true at all the global states are* $\mathcal{O}_i$*-accessible from* g.

Definition 30 ($NLTL_I$ semantics).
The semantics of $NLTL_I$ *is inductively defined as follows:*

$(IS, r, m) \models \phi$ *iff* $L(r, m)(\phi) = true$
$(IS, r, m) \models \phi \wedge \psi$ *iff* $(IS, r, m) \models \phi$ *and* $(IS, r, m) \models \psi$
$(IS, r, m) \models \neg\phi$ *iff it is not the case that* $(IS, r, m) \models \phi$
$(IS, r, m) \models O_i\phi$ *iff* $\forall(r', m')$ *such that* (r, m) $\mathcal{O}_i$ (r', m'), *then* $(IS, r', m') \models \phi$
$(IS, r, m) \models X\phi$ *iff* $(IS, r, m+1) \models \phi$
$(IS, r, m) \models F\phi$ *iff for some time* $m' \geq m$ $(IS, r, m') \models \phi$
$(IS, r, m) \models G\phi$ *iff for all time* $m' \geq m$ $(IS, r, m') \models \phi$
$(IS, r, m) \models \phi U \psi$ *iff there is some time* $m' \geq m$ *such that along the run such that* $(IS, r, m') \models \psi$ *and for each* $m \leq m'' < m'$ *we have* $(IS, r, m'') \models \phi$.

In the interpretation for obligations proposed here, this motivational attitude is ascribed to the agents by an external reasoner about the system. In this approach, agents do not compute their obligations in any way. In its definition, two points (r, m) and (r', m') are $\mathcal{O}_i$-related if $(r'm')$ makes possible that agent i functions correctly at the point (r, m).

It remains to define the normative notions of $NLTL_I$ that are not primitive. Specifically, we define what it means for some ϕ to be a violation, for an agent i to have the right to bring about ϕ, and an intuitive notion of sanction.

In order to define violation, we extend the language of $NLTL_I$ to include the propositional constant V as an abbreviation of the formula defined below. The meaning of the expression $V\phi$ states that ϕ holding in the system at some point is a violation (using a violation symbol is of course based on the work of Anderson (1967)).

Definition 31 (Violation).
From each literal built from a variable ϕ, $V\neg\phi$ *means that* $\neg\phi$ *is a violation at some point* (r, m) *in the system for some* $ns \in NS$, *such that* NS *is a set of norms, iff*

$$O_i(\phi\ U\ \psi) \rightarrow (\neg\phi\ U\ \psi)$$

If the system is at a point in which ϕ holds if agent i acts correctly until ψ holds, then $\neg\phi$ holds until ψ holds. Agent i not working correctly means that ϕ does not hold and that constitutes a violation in our system.

Anderson (1967) proposed a reduction schema for defining deontic operators within intensional logics. He defended that the logic of norms should be explored by treating normative statements as certain kind of conditionals. Thus, to say that i has the obligation to perform action α means that if the the action is not performed, then some undesirable state-of-affairs results. This was formalized by (a more complex version of):

$$Op = \neg p \rightarrow V$$

Some authors argued that undesirable states-of-affairs do not always follow infractions, and that not all violations are sanctioned. Further discussion produces the so-called contrary-to-duty paradoxes.

We can imagine a context in which if an agent i is functioning correctly then it will send an *accept* message to a *request* when some agreement preconditions hold, then agent i does not *accept* the request. In this situation, we say that agent's i not bringing about ϕ violates the pragmatic specification of accepting the *request*.

In some cases, agents have their behaviour specified in a way that performing some action does not constitute a violation. Rights give agents some freedom to act in some specific way. In this sense, *rights* are considered here exceptions to obligations. An agent has the right to bring about ϕ under some condition ψ if bringing about ϕ is not a violation ($\neg V(\phi)$). From an external point of view, we say that "there is a point in the system where agent i is functioning rightly if the holding of ϕ does not constitute a violation". We formalize this concept as follows:

Definition 32 (Right).

Let NS be a set of norms $(ns_1, \ldots, ns_n)$, and let the variables of agent Ag contain a set of violation variables $V = V(\phi)$ such that $\phi \in AP$. Agent i's functioning is right when ϕ holds, $R_i\phi$, for some $ns \in NS$ at some global state $r(m)$, $r(m) \in GS$ iff

$$\neg V\phi\ U\psi$$

Therefore, having the right to bring about ϕ under some precondition ψ means that until ψ holds along a run, then ϕ not being a violation also holds along that run.

In our approach, a right is a social concept that rules agents' behaviour by specifying their freedom. This is what we meant when referring to a "middle ground" between obligations and permissions. A normative concept of these characteristics makes it possible to formulate policies on autonomous agents' conversations without completely predetermining their behaviour, and without making conversation policies merely an ordering of message sequences to be followed.

- In NPRAG, rights are not only permissions as in Norman et al. (1998). In the formalization of right, we say that bringing about some ϕ is not a violation. When an agent is exercising a right, its freedom is specified in relation to that right.
- From a linguistic point of view, we can understand right-based rules as *defaults* for which there can be some exceptions; for example, when law changes and an exception to a right is made, that means that from now onwards exercising that particular right is now considered a violation. The linguistic interpretation is that if by default an agent has the right to *agree* or *refuse* to a *request*, then there can be a new policy that overrules the default and states that from now on exercising the right to *refuse* to a *request* sent by some agent-manager is a violation of the agent-manager's rights.

So, what happens when an agent not functioning correctly or rightly brings about some ϕ, which constitutes a violation? We stated that in these cases, there is an agent, called the normative agent, that, if working correctly, will sanction the offending agent. The specific nature of the sanction varies from system to system, and within the same system, from one scenario to another. The general pattern, however, is that the sanctioned agent will have the obligation to do something as a punishment for its violation. For example, agent i wants to participate in a bidding process to buy a property on behalf of some estate agents. Say that to enter the auction, you need to pay some deposit of 1,000 in advance. If the agent (its role is bidder, $bidder \in RN$) wins the auction with an agreed price of 200,000 for the property, but decides to break the agreement and not buy the house after winning the auction, then this agent has the obligation of paying a fine. In this case, the fine can be the 1,000 deposit paid to enter the auction in the first place. The agent with the right to impose fines in this scenario can be the agent playing the role of auctioneer *auctioneer* $\in RN$. We can formalize this notion of sanction as follows:

Definition 33 (Sanction).

Let b denote the role of bidder such that $b \in RN$, then a agent i such that $i \in Ag$ playing the role of bidder b has the obligation to pay a fine (by bringing about ϕ) iff

$$b_i \wedge (O_i\phi \; U \; \psi \wedge \neg F\phi) \; U \; \psi) \rightarrow O_i\omega$$

Thus, if the system is at a point in which if an agent playing the role b (bidder) is acting correctly, ϕ holds until ψ holds, and $\neg\phi$ eventually happens while ψ, then i is sanctioned with the obligation of paying some fine ω.

In agent communication, the auctions processes are specified Auction interaction protocols (see below). Moreover, the notion of sanction presented here can be greatly complicated by considering more complex behaviour to detect and sanction violations. However, for our purposes this minimal normative structure is sufficient to formulate a normative pragmatics for agent communication. In any case, the normative specification of multi-agent systems is a difficult problem in its own, and it is not within the aims of this thesis.

After giving a grounded semantics to the normative concepts that will be used in the interaction protocols and policies of NPRAG, we present in the next section the axiomatics for $NLTL_I$.

6.4.3 $NLTL_I$ Axiomatics

Studying the complexity of the specification language $NLTL_I$ is important because it is interesting that the complexity of the reasoning on protocols defined using $NLTL_I$ is not too hard computationally.

A variety of logics have been proposed combining operators for knowledge and time for the interpreted systems model (Fagin et al., 1995; Halpern and Vardi, 1989), each producing differences in their complexity. Furthermore, it is well-known that the system KD_n that characterizes Standard Deontic Logic is sound and complete. In this section we give a complete and sound axiomatization of $NLTL_I$ which consists of the axioms for obligations and temporal logic.

Halpern and Vardi (1989) in particular classified many of these logics and studied their differences in complexity. They noticed that there are at least two factors in the variance of the logics: The language itself and the characteristics of the underlying distributed system. With respect to the language, the choices include whether the temporal logic is a linear time or a branching time logic, whether operators for common knowledge are considered, and whether our system consists of one agent or it is a multi-agent system. We will not consider here additional requirements such as perfect recall, unique state and synchrony of our systems since we are interested in giving an axiomatization for the most general case.

We say that the Kripke structure M associated to the interpreted system IS where the accessibility relation $\mathcal{O}_i$ models obligation is serial. Therefore, the axiomatics of the deontic accessibility relation are given by the system KD_n. This corresponds to the axiomatization of Standard Deontic Logic (von Wright, 1951). The general idea is that if you have the obligation of bringing about ϕ, then you do not have the obligation of bringing about $\neg\phi$. This guarantees consistency in the agents' obligations. Adding to that the axioms for the temporal operators, the result is that the following axioms provide a sound and complete axiomatization of $NLTL_I$:

PC All tautologies of propositional logic.
T1 $X(\phi \rightarrow \psi) \rightarrow (X\phi \rightarrow X\psi)$.
T2 $X(\neg\phi) \equiv \neg X\phi$.
T3 $\phi\ U\ \psi \equiv \psi \vee (\phi \wedge X(\phi\ U\ \psi))$.
RT1 From ϕ infer $X\phi$.
RT2 From $\phi' \rightarrow \neg\psi \wedge X\phi'$ infer $\phi' \rightarrow \neg(\phi\ U\ \psi)$.
MP From ϕ and $\phi \rightarrow \psi$ infer ψ.

The axiomatics for the deontic operator is as follows. i denotes a set of agents such that $i = 1, \ldots, n$.

PC All instances of propositional tautologies.
MP If ϕ and $\phi \rightarrow \psi$, then ψ.
NEC If ϕ, then $O_i\phi$.
K $O_i(\phi \rightarrow \psi) \rightarrow (O_i\phi \rightarrow O_i\psi)$.
D $O_i\phi \rightarrow \neg O_i\neg\phi$.

Theorem 3. *The system $NLTL_I - Ax$ is a sound and complete axiomatization with respect to the class of models $NLTL_I$ that are serial.*

The proof of the axiomatics of $NLTL_I$ follow the same technique as that of $MLTL_I$ (see section 5.1.8).

We finish here the definition of the syntax, semantics and axiomatics of $NLTL_I$. We had various motivations to define this logic: First, given that $NLTL_I$ is going to define the semantics of the normative operators used in NPRAG, a deontic component was needed. We have introduced an standard operator for obligation which was then used to define several other normative concepts. Among them, the notion of right is going to be widely used in the specification of the ACL pragmatics. Second, the semantics of $NLTL_I$ is grounded in a computational model which gives us several methods to verify the ACL pragmatics. Finally, the temporal operators provide useful tools to analyze how agents' rights and obligations change over time. This also means that coordinating communication through norms allows us to focus on the external behaviour of agents, instead of modelling their mental reasoning to interpret messages. As far as we know, there is not a language available with all these characteristics and that could be used out of the box as a specification language for a public and verifiable ACL. The characteristics of $NLTL_I$ help to fulfil the requirements for ACLs discussed throughout this thesis.

Next section presents the interaction protocols and conversation policies that form the ACL normative pragmatics. $NLTL_I$ is used to specify the meaning of the normative notions used in the pragmatics: obligation, violation, right and permission. Moreover, the adequacy of using $NLTL_I$ itself to specify the interaction protocols is discussed.

6.5 Conversation Norms

$NLTL_I$ as a specification language provides a formal, unambiguous, and grounded meaning for the key social concepts to be used for NPRAG. These concepts offer the following picture:

- Agent conversations often occur within an institution. In fact, there are specific speech acts such as *declare* that are pure institutional facts. When the appropriate role uses the adequate speech act within an institution, the agent has done *something* by sending that message. The rules defining the institution are denominated *constitutive rules* and are specified by means of *conversation protocols.*
- Constitutive rules mainly specify interaction protocols such as English Auction, whereas *regulative rules* are concerned with more context-dependent aspects in the form of *conversation policies.* Both constitutive rules and regulative rules are declarative and their aim is to stabilize communication by contextually enriching the meaning of the ACL semantics so that intention recognition is not necessary and agents are given rules to achieve the Rational Effects.

- Agents play roles, and those roles influence agents' communicative behaviour towards the achievement of the rational effects.
- Right is a normative notion that rules agents' communicative behaviour by specifying their freedom instead using pure restrictions and/or obligations. Furthermore, definitions of violation and sanction are provided.

ACL normative pragmatics consists of the effect that the context of utterance, the Rational Effects and turn-taking has on the sender's choice of expression and the receiver's interpretation of an utterance. Agents would have their communicative behaviour stabilized by the use of conversation policies and interaction protocols for a conversation to be meaningful and achieve their particular goals state in the semantics of the messages.

We discussed in chapter 3.3 a number of approaches to the specification of interaction protocols for agent communication. We offered a comparison with respect to the desirable requirements for agent communication languages, stressing the difficulty of designing effective protocols using procedural languages or diagram-based specifications (Greaves et al., 2000; Fornara et al., 2004; Cost et al., 1999a). In particular, we argued that what is called in most of the literature "conversation policies" is actually "interaction protocols" (in the way we use these two concepts in this thesis) (Greaves et al., 2000). Interaction protocols are merely concerned with the structure of the conversation.

In our view, an ACL specification (both its semantics and pragmatics) ought account for the *meaning* of speech acts' use in specific contexts. Given that the semantics is specified using a logic-based language, then it is only natural that a logic-based language is also used in the ACL pragmatics. However, this is not only a matter of aesthetics; the protocols and policies to be specified using the logic-based language will be declarative so they specify *what agents can achieve* using the rules instead of *how to achieve* a particular result. Furthermore, formal logic constitutes a more appropriate tool to model and reason about multi-agent systems than procedural programming languages or ontology-based languages like OWL (Kagal et al., 2003). Besides, there are a number of verification techniques for logic-based specification languages (Manna and Pnueli, 1995) of systems we can put to good use in the verification of agent communication languages.

Note that we are discussing the syntax of the language and not its semantics, since the meaning of the normative concepts of NPRAG have been already defined by $NLTL_I$. When considering which language used for the formalization of the speech acts, we conclude that, although the semantics of the cognitive and temporal operators was defined by $MLTL_I$, the syntax of the messages was going to follow the FIPA specification. We gave two reasons for this decision: First, there is nothing wrong with the form of the speech acts specification in FIPA. Most of the criticisms have been addressed to its semantics. Second, we are interested in contributing to the standardization effort of agent communication led by FIPA, so we provided an alternative formalization for its semantics which would not present those problems attributed to the FIPA ACL semantics.

We follow this general methodology for the specification of the ACL pragmatics. We stress that pragmatics plays a more important role that has been conceded so far in the literature, and propose a theory accordingly. The two main consequences of this approach is that meaning is not longer a semantic issue, but the combination of semantic specification and pragmatic principles, and that the pragmatics have to be further developed so that it does not only consist in the order in which messages can be uttered. In order to do so, we have proposed a normative theory based on rights, and we have given a grounded semantics to a number of normative concepts that will conform the pragmatic rules of the ACL. The next issue to consider is which representation is the adequate to built such rules. Once that issue is settled, we provide examples of how the normative rules can be specified.

6.5.1 Representation

Leaving aside the procedural and diagram-based approaches already discussed, there is a recent trend in the specification of interaction protocols based on propositional linear temporal logic (PLTL) (Endriss, 2005) and finite-state machines (Endriss et al., 2004).

Endriss (2005) proposes to specify the class of all sequences of messages that are allowed by a given protocol. He uses propositional LTL (PLTL) to specify the protocols and model-checking techniques to verify the runtime conformance of conversations to the protocol. Conversation templates are defined as sequences of dialogue moves (speech acts). Those dialogues that can be captured by protocols based on finite-state machines are legal according to a protocol if and only if they are accepted by the finite-state machines that correspond to the protocols.

Standard finite-state protocols and PLTL are not suitable to interactions involving commitments, social expectations and, in our case, rights and obligations. For example, we are interested in attributing to the (role of) auctioneer the obligation to close the auction at some point, and to give the bidder the right to bid after the auctioneer *declares* the auction open. In other words, we need to consider how the system evolves as a result of agents' performing actions (speech acts in our case). It is convenient that the execution of speech acts be ruled by some protocols and policies if we want communication to be efficient.

6.5.2 Normative Protocols and Policies

Thus, for the formulation of a high-level norms of conversation, we need to consider taking into account the following elements:

1. A set of atomic propositions P to describe facts. They usually consist of propositional content of messages.
2. A set of **agents** that participate in the conversation.
3. A set of **speech acts** (query, request, etc.) that convey the illocutionary and perlocutionary acts of performing a communicative action.
4. A set of **normative rules** of the form $np_i(sa(i, j, P))$ which consist of a normative predicate (right, obligation), the action (a speech act) and the content of the speech act ϕ.
5. A set of **broadcasting actions**. Broadcasting actions denoting events state that a speech act sa is sent, received, answered or not-answered. This aspect refers to the history of the conversation.
6. A set of **roles** taken by the agents involved in the interaction. Roles are specified as facts about individual agents $role_i$.
7. An agent performing the role of **normative system** ns encoded in the environment's local state of the system. ns has the obligation of monitoring the conversations to detect violations, apply sanctions and making sure that messages are delivered.

In $NLTL_I$, we formalized obligations, rights and permissions as entirely dependent on agents' local states. Thus, any communicative actions they take are a function of their local state. Their local states also contain information regarding their initial state in the execution and the history of messages sent and receive (i.e., its conversational record; we build on the knowledge-based interpreted system model (Fagin et al., 1995) to model the history of conversation).

Definition 34 (History).

Let us consider an agent i such that $i \in Ag$, a set of broadcasting actions BE, a set of speech acts SA, a set of initial states S_{0i} for agent i, and a set of contextual actions DO_i for i. A history for agent i is a sequence where

1. *The first element is in S_{0i},*

2. *the later elements consist of nonempty sets of broadcasting actions such as* $sent_i(sa(i,j,P))$, $receive_i(sa(i,j,P))$, *or* $do(i,\alpha)$ *such that* $\alpha \in DO_i$.

The history of conversation of an agent i at some point (r,m) of the system is composed by its initial state and the sequence of steps corresponding to i's actions up to time m. We can also say that if an agent i at a point (r,m) has only sent an *agree* speech act to agent j, $sent_i(agree(i,j,P))$, then its history at point (r,m) is the result of appending the set $\{sent(i,j,agree(P))\}$. Furthermore, a broadcasting event occurs in round $m+2$ of run r if it is contained in some agent's history of conversation in $(r,m+2)$.

We have mentioned above that our framework models the system environment as a normative agent ns whose task is to decide when performing a speech act is a violation and the sanctioning it when appropriate. In order to take these decisions the environment's local state must record the events that take place in the system, namely, the speech acts performed by the agents involved in a conversation. Furthermore, it need to keep an up to date record of the evolution of agents' rights, obligations and permissions according to the actions they have performed so far, taking into account the fact that performing speech acts' cause social expectations. Note, however, that determining and reasoning about the actions that ns can perform is part of the social structure of the system. Therefore, the ACL specification does not account for the acquisition of knowledge or beliefs by ns nor the reasoning employed to sanction violations. Doing so is not within the purposes of this paper.

Thus, we need to consider both agents' and the environment's actions to explain how their actions cause the system to change state: $(\alpha_e, \alpha_1, \ldots, \alpha_n)$ and a transition function $\delta(\alpha_e, \alpha_1, \ldots, \alpha_n)$ to map global states to global states. We can now define a protocol as a mapping from the set L_i of agent i's local states to nonempty sets of acts in BE_i. Furthermore, a protocol P_e for the normative agent ns is a mapping from the set of the environment's local states L_e to nonempty sets of actions in DO_e.

We include normative concepts and propositional variables in our protocol rules. Furthermore, these rules must be declarative, that is, they say what the rights and permissions of the agents are, rather than a procedure to move from to one state to another. This secures the high-level character of our ACL. Interaction protocols are defined in NPRAG using if-then rules as the constitutive rules that specify the legal interactions of conversations. If agent j receives a *request* then agent j has the right to answer either by *agreeing* or by *refusing*.

We elaborate on these points in order to give specify some of the FIPA interaction protocols.

6.5.3 Request

Typically, protocols are described by means of programs written in some programming language. For clarity of exposition we will use in this paper $NLTL_I$ extended with parameters for agents, roles and actions. Having extended the Interpreted Systems model to express normative notions for their use in agent communication languages, we could have employed a similar strategy and adapt a simple programming language defined within the interpreted systems model (Fagin et al., 1995) to express protocols that include agents' roles, rights, obligations, speech acts and broadcasting actions. After showing in this section how our approach can be used to specify an ACL pragmatics using norms, we will offer an example of a protocol using a simple programming language.

Let us consider the FIPA Request interaction protocol. This protocol allows one agent to request to bring about some propositional content ϕ. If the receiver of the *request* speech act is functioning *rightly*, then it will send an *agree* or a *refuse* as a response to the *request*. If the answer is an *agree*, and the agent is functioning correctly at that point, then it will communicate an *inform* if the request is satisfied, or a *failure* if the object of the request is not achieved. The specification of this protocol in NPRAG looks is composed by the following norms of conversation:

1. $principal_i \wedge secretary_j \rightarrow R_i(request(i, j, \phi))$
2. $receive_j(request(i, j, \phi)) \wedge \neg sent_j(refuse(j, i, \phi)) \rightarrow R_j(refuse(j, i, \phi))$
3. $receive_j(request(i, j, \phi)) \wedge \neg sent_j(agree(j, i, \phi)) \rightarrow R_j(agree(j, i, \phi))$
4. $sent_j(agree(j, i, \phi)) \wedge F\phi \rightarrow O_j(inform(j, i, \phi))$
5. $sent_j(agree(j, i, \phi)) \wedge \neg F\phi \rightarrow O_j(failure(j, i, \phi))$

Note that the proposition of the normative predicates for rights, obligations and permissions are taken as expressing a communicative action like "agent i agrees with agent j to bring about some ϕ".

In the Request specification there are two agents i and j that take the roles of *secretary* and *principal* respectively. As a propositional content of the speech acts, we can think of a situation in which agent *principal* has the right to request to agent *secretary* to book a number of flights.

The rules state that the *principal* has the right to send any request message to the secretary, and that the secretary can answer to these messages either by agreeing or refusing if an answer has not been produced yet. The two obligation rules state that an agent has the obligation to send an *inform* having already sent an *agree* message and not having sent yet *inform* that the request has been satisfied.

As it is, the reasoning rules presented above capture the transitions that a system functioning rightly can perform under the NPRAG Request interaction protocol. However, we need something else, that is, to instantiate some of the facts of the NPRAG specification of *request*. In particular, we need to say which messages have been sent or are still pending. As discussed above, the history of conversation is part of agents' local state, whereas the status of messages and agents' rights and obligations are encoded in the environment's local state. None of these components are part of the interaction protocol specification. Indeed, for the sake of generality, it is desirable that our protocols only provide a set of norms of conversation to facilitate agents' next move *in absence of any specific circumstances*.

6.5.4 Query-If

In the FIPA Query-IF interaction protocol, an agent i queries agent j whether or not a proposition ϕ is true. The receiver has the right to either *agree* or *refuse* to send and *inform* message providing an answer. In the case that agent j agrees, then it has obligation to send a notification which can be an inform stating the truth of falsehood of the proposition ϕ. If agent j sends a refuse message the protocol ends there. We only show the relevant normative rules of this protocol:

1. $journalist_i \wedge policitian_j \rightarrow R_i(queryif(i,j,\phi))$
2. $receive_j(queryif(i,j,\phi)) \wedge \neg sent_j(refuse(j,i,\phi)) \rightarrow R_j(refuse(j,i,\phi))$
3. $receive_j(queryif(i,j,\phi)) \wedge \neg sent_j(agree(j,i,\phi)) \rightarrow R_j(agree(j,i,\phi))$
4. $sent_j(agree(j,i,\phi)) \wedge F\phi \rightarrow O_j(inform(j,i,\phi))$
5. $sent_j(agree(j,i,\phi)) \wedge \neg F\phi \rightarrow O_j(failure(j,i,\phi))$

We can see that its structure is almost equivalent to the Request protocol; only the use of *queryif* instead of *request* is different. This means that our proposal is high-level enough so that it is easily adaptable to represent different interaction protocols and different contexts. Only the content of the messages and the roles of the agents may change.

The specification of the constitutive rules of conversations enable us to formulate a number of policies that contextually contrain agents' communicative behaviour within the protocol in terms of their rights, obligations and permissions.

6.5.5 Conversation Policies

Since conversation policies usually restrict agents' behaviour within conversations, the notation of the pragmatic regulative rules that conform NPRAG conversation policies consists of the components used in the specification of interaction protocols. Moreover, we would like to stress the importance of one of the elements and propose a new one:

- A set of **contextual actions** DO_i that depend on specific scenarios, e.g., the action of *bidding* depends on the agent being in an auction.
- A **conflict resolution action** so that in case of conflict between rules of a policy, one rule has *priority* over another one.

Constructs such as the conflict resolution actions, the contextual and broadcasting actions depend on the platform in which agents run. That is, these actions are defined by the programming language in which agents are built. For example, in Java built platforms like JADE, sending messages is simply a case of creating an ACLMessage, setting the parameters (sender, receiver, reply-to, performative, etc.) and then sending it using the send() method in the agent object.

If the normative rules in the interaction protocols specify the legal structure of the conversation, conversation policies regulate agents' behaviour according to contextual information within the protocol. Roles and background knowledge provide valuable information for agents to choose the right course of action. Unlike the specification of the interaction protocols, we consider the content of the speech acts when proposing normative rules. Furthermore, note that the policies are tightly combined with the ACL semantics defined in the previous chapter. Thus, the meaning of a speech act such as *queryif* is enriched by the rights, obligations and permissions of agents to use that particular speech act.

We can imagine a situation in which an agent *paxman* has the right to *queryif* a politician agent *pm* about the truth of the "peersmoney" scandal as long as we are not in electoral campaign.

$$paxman_i \wedge pm_j \rightarrow R_i(queryif(i, j(peersmoney))U\neg(elections)$$

Another example can be of an agent j acting on behalf of an airline company serving flights to European countries, that could have a policy that states that it should agree to every request regarding flight tickets to Europe (i.e., answering about flight times and providing the best offer for a potential buyer) and another one specifying that it has the obligation to refuse every request about flights to non European countries.

- $customer_i \wedge seller_j \wedge receive_j(request(i, j, europeanFlight)) \rightarrow O_j(agree(j, i, \phi))$.
- $receive_j(request(i, j, nonEuropeanFlight)) \rightarrow O_j(refuse(j, i, \phi))$.

This issue shows how using normative conversation policies help agents to achieve the perlocutionary effects since the perlocution of *agree*, namely, that the receiver satisfies the object of the requested action, is now specified to be an obligation of the seller. This is a crucial point to help agents to achieve the rational effects of an speech act. For example, we can specify a rule to state that if an agent makes a promise to increase the taxes on air planes fuel, then it has the obligation to do so:

$$G(send_i(promise(i, public, taxairplanesFuel)) \rightarrow \\ O_i(increaseTaxes(airplanesFuel))$$

The extension of our approach to other protocols and policies in the FIPA specification is fairly straightforward. Our approach shows how a well-defined normative concepts can be used to propose a high-level ACL pragmatics that are declarative, takes into account the context and that helps agents to achieve the perlocutionary effects of the speech acts. These two properties of the normative pragmatics, *contextual* and *perlocutionary*, fill in the last gaps in the list of requirements for ACLs discussed in section and table 5.2. Next section offers a comparison to other approaches and discusses some short term future work necessary to improve the ongoing work presented in this paper. As a final note, the simplicity of the protocols and policies specified in this section was intentional. An important point for any future application of agent communication languages remains the proposal of high-level but simple ACL semantics and pragmatics.

6.5.6 Programs

Fagin *et al.* introduce a simple programming language which can be easily related to an Interpreted System (Fagin et al., 1995). Although the language is designed to express agents' knowledge, it can be adapted for its use in specifying norms of conversation. The basic standard program for agent i consists of statement of the form

case of

if $t_1 \wedge k_1$ **do** a_1

if $t_2 \wedge k_2$ **do** a_2

end case

where the t_i's are tests about some facts, k_i are knowledge test for agent i and a_i denote agent i's actions. We modify these knowledge-based programs to express tests over obligations, rights and permissions of agents, namely, to normative-based programs. The normative component consists of a Boolean combination of the form $O_i\varphi$ where φ can be an arbitrary formula that may include other deontic and temporal operators. Using this simple language we can express high-level protocols for agent communication. We represent the Fipa Request protocol specified above in table 6.1.

case of

if $(principal_i \wedge secretary_j) \wedge R_i(request(i,j,\phi))$ **do** $send_i(request(,j,\phi))$

if $receive_j(request(i,j,\phi)) \wedge R_j(refuse(j,i,\phi))$ **do** $sent_j(refuse(j,i,\phi))$

if $receive_j(request(i,j,\phi)) \wedge R_j(agree(j,i,\phi))$ **do** $sent_j(agree(j,i,\phi))$

if $sent_j(agree(j,i,\phi)) \wedge F\phi$ **do** $O_j(inform(j,i,\phi))$

if $sent_j(agree(j,i,\phi)) \wedge \neg F\phi$ **do** $O_j(failure(j,i,\phi))$

end case

Table 6.1. Program for Request Protocol.

At first glance, it may seem a bit odd to specify agents' obligations and rights after the operator **do** instead of actions. However, in the interpretation of obligations and rigths provided by $NLTL_I$, $O_i\varphi$ means that "φ holds in agent i is working correctly" whereas $R_i\varphi$ is interpreted as "φ holds at some point of the system (r,m) if agent i is acting rightly". Therefore, the last statement of the program denotes that if agent j has *agreed* to bring about some ϕ to agent j and ϕ does not eventually happens in the run of the system then agent j does send a failure message to agent i if working correctly. The protocols and policies that conform our normative pragmatics can therefore be expressed by a programming language in a high-level manner.

In fact, the conversation norms presented in this thesis can be expressed by other high-level declarative languages, including all purpose programming languages such as Prolog. If we are interested in the verification and implementation the conversation protocols, it is worth investigating the possibility to represent our ACL pragmatics by means of automata. Therefore, we are interested in a language which gives us sufficient expressive power to elaborate complex and fine-grained pragmatic rules, but that at the same time can be *translated* into an automata form for their posterior conformance testing. A declarative language such as Prolog maybe the an appropriate choice to represent the norms of conversation by means of protocols and policies. This is supported by the fact that $NLTL_I$ models can be translated into Büchi automata which can be naturally represented using Prolog SWI-Prolog (2005). Note that this is not dissimilar to the traditional model-checking verification by which temporal models are translated into automata for its verification. In our case, an extra difficulty lies in the fact that we need to relate the deontic operator O_i to the set of states of a transition system.

6.6 Applicability and Verification

The approach to agent communication presented so far complies with most of the desirable properties for agent communication languages discussed throughout the thesis. However, we have not shown yet how our proposal and, in particular, the ACL pragmatics may be verified. There are various methods of verification which depend on the type of ACL, on the information available, and on whether we are interested in verifying the ACL at design time or at run time (Guerin and Pitt, 2002). Unlike other approaches, we are particularly interested in verifying the ACL pragmatics (only) because the pragmatics encodes the general communicative behaviour of agents. Following this, the type of the ACL to be verified corresponds, in our approach, to the normative component.

Logic has been used for various tasks in the development of multi-agent systems: As a specification language, as a verification language and as an implementation language. Temporal and dynamic logic have been the most widely used logics for specifying agents. We have discussed already several reasons on why logic is a convenient tool to specify software agents. Summarizing, logic is useful to express the expected behaviour of the system without specifying how the specification should be satisfied by an implementation. However, we also pointed out that using the possible world semantics creates a number of problems, the most important being that possible world semantics are ungrounded. In other words, it is not clear the relation between the accessibility relations that characterize agents' internal states and a computational model. Most of the specification languages used to give the semantics of ACLs are multimodal logics defined in the tradition of possible world semantics, which effectively means that it is not possible to verify the ACLs.

There are two main traditions in the verification of systems at design time: Axiomatic verification and semantic verification. Axiomatic verification consists of a proof problem, and it is therefore limited by the difficulty of the proof problem. The difficulty is greatly increased when we consider the multimodal logics that act as ACL specification languages. Nowadays, a semantic approach is favoured. In this second approach, the problem consists of determining whether a formula ϕ is valid in a model M, that is, whether $M \models \phi$. This method is known as model-checking and it has been particularly useful for temporal logics. Model checking relies on the relation between the semantic models and finite-state machines. Which is to say that a we can model-check a formula ϕ for a model M if the model can be represented as a finite state machine. As we showed earlier, both our pragmatic specification language $NLTL_I$ and Prolog-based protocols (constitutive rules) can be translated into finite state machines and then use a model checking algorithm to perform the verification (see, for example, Clarke et al. (1999) for model-checking using automata). Furthermore, a method to model-check a logic similar to $NLTL_I$ has been developed by Raimondi and Lomuscio (2004).

6.6.1 From $NLTL_I$ to Automata

Traditionally, the models of Linear Temporal Logic are transition systems. A transition system models a system by means of states and transitions.

Definition 35 (Transition System).

A transition system $M = (S, R, L)$ is a set of states S and a transition relation on S such that each $s \in S$ has some successor $t \in S$ such that $(s, t) \in R$. L is a labelling function for the atomic propositions AP that describe facts about the system, $L : S \rightarrow 2^{AP}$.

Thus, a transition system has a collection of states S, a total transition relation R and the labelling function that labels true atomic propositions at a given state. Transition systems have been widely used for the specification and verification of reactive systems Manna and Pnueli (1995). Reactive systems are a class of software and/or hardware systems which have ongoing behaviour, that is, they do not terminate. Examples of reactive systems include operative systems, traffic lights, data communication protocols such as the Internet, etc.

The semantics of LTL is given in terms of single computation (or execution) paths π, that is, the operators of LTL allow to describe how the system evolves in a single execution. If we assume that the system consists of a set of runs (infinite sequences of global states) it is easy to see that Interpreted Systems IS can also represented by state transition graphs. In fact, automata are usually described as labelled transition graphs where the Kripke structure $M = (S, R, L)$ consists of a set of states, a transition relation $R \subseteq S \times S$ and the labelling function $L : S \to 2^{AP}$. In other words, automata can be described by transition systems with a fixed number of boolean atomic propositions. Thus, transition relation R can be seen as the infinite sequence of states (or run) r that is produced by the application of the transition relation over the set of states S. Automata over infinite executions are called Büchi automata.

Thus, transition functions in Büchi automata assign to each state a positive boolean formula over states.

Definition 36 (Büchi Automaton).

An Büchi automaton is a tuple $A = (\Sigma, S, s_0, \delta, F)$ *where*

1. Σ *is a finite input alphabet.*
2. S *is a finite non empty set of states.*
3. $s_0 \in S$ *is an initial state.*
4. δ *is the state transition function:* $\delta : S \times \Sigma \to S$*, where* $\delta(s, a)$ *gives the set of states to which the automaton can move with* a *from state* s*.*
5. F *is the set of good states,* $F \subseteq S$ *that is visited infinitely often.*

An execution or run of A *over a word* $\phi = a_0 a_1 \ldots \in \Sigma$ *is an infinite sequence of* $\rho = q0 q_1 \ldots$ *of locations* $s_i \in S$ *such that* $q_0 \in S_0$ *and* $(q_i, a_i, q_{i+1}) \in \delta$ *holds* $\forall i \in \mathbb{N}$*. The run* ρ *is accepting iff* $\exists q \in F$ *such that* $q_i = q$ *holds for infinitely many* $i \in \mathbb{N}$*.*

We are interested in the translation of the structures generated by adding a deontic accessibility relation on (the global states of) IS, that is, we are interested to represent $NLTL_I$ structures in terms of Büchi automata. Thus, we have to deal with the notion local state. In other words, for the above graph to represent interpreted systems the states should correspond to global states, to be able to represent a set of agents and their local states. Therefore, we need two more definitions:

An automaton with local states is a tuple $(\Sigma, S, s_0, \delta, F)$. However, we will not use F and Σ in the definition to make things clearer.

Definition 37 (Automaton with local states).

A Büchi automaton with n agents is a tuple $AL = (S, s_0, \delta, L_i, \ldots, L_n)$, *where*

1. *S denote a set of states, s_0 denotes the initial global state and δ denotes a transition relation over states.*
2. *$L_i, \ldots, L_n$ are the agents' local states, where for each i, $L_i \subseteq S$. The local state of agent i at state s is defined as $s \cap L_i$. This is to identify as the global state of the automaton: $g(s) = (s \cap L_1, \ldots, s \cap L_n, s)$.*

We now follow the same procedure that was used for the definition of $NLTL_I$, namely, we associate the Büchi automaton with local states with the interpreted system $IS = (T, L)$ to generate an interpreted automaton.

Definition 38 (Interpreted Automaton).

A system $AL_I = (T, g_0, L)$ is the automaton generated by associating and interpreted system IS with an automaton AL where T are sets of runs such that

1. *Each global state in T consists of the $g(s)$, that is, of the local states of the agents at some state s of the automaton. Thus, we say that for all times m, $r(m)$ has now the form of $(s \cap L_1, \ldots, s \cap L_n, s)$.*
2. *Thus, $r_i(m)$ denotes agent i's local state if $r_i(m) = s \cap L_i$.*
3. *g_0 corresponds to the global state $r(0)$.*
4. *$s \cup \mathbb{N}(t)$ satisfies $\delta(s, t)$ for all times m, if $g(s) = r(m)$ and $g(t) = r(m+1)$ for some s and t.*

Thus, agents' local states in the automaton are related to agents' local states in the interpreted system IS. Furthermore, the deontic accessibility relation of $NLTL_I$ is a relation between agents' local states, so it is quite clear that we can define an interpreted deontic automaton as a tuple $AL_d = (AL, O_i, \ldots, O_n)$ where $O_i, \ldots, O_n$ denotes agent i's deontic states over AL.

Translating $NLTL_I$ to Büchi automata is the first step to its eventual verification via model-checking.

6.6.2 From Automata to Protocols

Büchi automata have been applied in the area of computational linguistics to the implementation of regular grammars, corpora, speech recognizers and morphological analyzers. The basic idea for the application of automata to our approach is to simply consider the transition function of the transition facts to be speech acts. Thus, if we were interested in very simplistic protocols to merely specify the order in which some speech acts can be uttered, we would use the following representation. Consider *request* the initial transition action for agents i and j.

```
transition(i,j,request):-
  transition(i,j,agree) ;
  transition(j,i,refuse).
```

This means that a *request* can be answered by sending either an *agree* or a *refuse* actions. This form means that the labels of transitions determine the sequence of states in the execution of the machine, so we omit the states and focus on the transitions, which in our case are represented by speech acts. For example, the Request interaction protocol defined by FIPA ACL (2002) states that this protocol sets up the structure of a conversation in which one agent i requests another agent j to do some P. Agent j can *agree* or *refuse* to achieve P (that describes a fact). If it *agrees*, then it has to send a notification after P is achieved. The notification can consists of a *inform* that P is achieved or a *failure* if j has failed to achieve P.

The problem is obvious, namely, we need an inference which states that sending an *agree* or a *refuse* is true if one has received a *request*. We can express this using the following form (we represent only the clauses directly related to the discussion):

```
send(i,j,agree) :-
    receive(i,j,request).

send(i,j,refuse) :-
receive(i,j,request).
```

A problem with these rules is that they are too restrictive. These are *deterministic* rules that completely restrict agents' behaviour whereas a set of rules based on agents' rights will model their communicative behaviour by asserting their freedom. We can try by adding the normative concepts defined in $NLTL_I$:

```
right(Y,X,agree) :-
```

```
    receive(Y,X,request).

right(Y,X,refuse) :-
    receive(Y,X,request).
```

However, this is still not sufficient when we are interested in specifying pragmatic rules that contextually constrain the content of speech acts. This representation is appropriate for the automaton A defined in the previous section, but here we are interested in representing AL_I automata in Prolog. That is, automata with local states variables which correspond in our case with the normative concepts of $NLTL_I$. This means that the clauses of our protocols and policies must be more expressive. The obvious extension is to add variables for the content of messages, since NPRAG rules modify agents' communicative behaviour by taking into account the content of speech acts.

So, we need to include normative concepts and propositional variables in our protocol rules. Furthermore, these rules are declarative, that is, they say what the rights and permissions of the agents are, rather than a procedure to move from to one state to another. This secures the high-level character of the unified ACL. Interaction protocols are defined in NPRAG using if-then rules as the constitutive rules that specify the legal interactions of conversations. If agent Y receives a *request* then agent Y has the right to answer either by *agreeing* or by *refusing*:

```
right(X,request(X,Y,P)).

right(Y,agree(Y,X,P)):-
  receive(Y,request(X,Y,P).

right(Y,refuse(Y,X,P)):-
  receive(Y,request(X,Y,P).
```

We elaborate on these points in order to give a declarative formulation to some of the interaction protocols of FIPA. In a normative pragmatic approach, the protocols consist on specifying agents' rights and agents' duties in terms of obligations.

6.6.3 Prolog Interaction Protocols

Prolog Request Protocol

Let us consider again the FIPA Request interaction protocol. This protocol allows one agent to request to bring about some P (by performing some action). If the receiver of the `request` speech act is functioning *rightly*, then it will send an `agree` or a `refuse` to the `request`. If the answer is an `agree`, and the agent is functioning correctly at that point, then it will communicate an `inform` if the request is satisfied, or a `failure` if the object of the request is not achieved. The specification of this protocol in Prolog would be as follows:

```
principal(pri).
secretary(sec).

sender(X):-
  principal(X).

receiver(X):-
  secretary(X).

right(X,request(X,Y,P)):-
  sender(X),
  receiver(Y),
  content(P).

right(Y,agree(Y,X,P)):-
notrepliedagree(Y,request(X,Y,P)).

right(Y,refuse(Y,X,P)):-
notrepliedrefuse(Y,request(X,Y,P)).

notrepliedagree(Y,request(X,Y,P)):-
pending(Y,agree(Y,X,P)).

notrepliedrefuse(Y,request(X,Y,P)):-
pending(Y,refuse(Y,X,P)).

obligation(Y,inform(Y,X,P)):-
```

```
sent(Y,agree(Y,X,P)),
pending(Y,inform(Y,X,P)).

obligation(Y,failure(Y,X,P)):-
sent(Y,agree(Y,X,P)),
pending(Y,failure(Y,X,P)).
```

Just to remind the reader that we have two agents, sec and pri, that take the roles of principal in a company and its secretary respectively. We have also given some content to illustrate a simple situation in which agent pri has the right to request to agent sec to book a number of flights.

The rules state that the principal has the right to send any request message to the secretary, and that the secretary can answer to these messages either by agreeing or refusing if an answer has not been produced yet. The two obligation rules state that an agent Y has the obligation to send an inform having already sent an agree message and not having sent yet inform to notify that the request has been satisfied. The rest of the rules describe the various relations between the broadcasting actions and the status of the conversation.

However, as we argued earlier, we need something else, that is, we need to instantiate some of the facts of the NPRAG specification of request. Although recording the status of messages is not a problem to be solved by the ACL specification, we can assume that the system is at some point in which some messages present the following status:

```
sent(pri,request(pri,sec,flightAntartida)).
sent(pri,request(pri,sec,flightItaly)).
sent(pri,request(pri,sec,flightFinland)).
sent(pri,request(pri,sec,flightChina)).
sent(sec,agree(sec,pri,flightAntartida)).
sent(sec,agree(sec,pri,flightItaly)).
sent(sec,refuse(sec,pri,flightFinland)).

pending(sec,agree(sec,pri,flightChina)).
pending(sec,refuse(sec,pri,flightChina)).
pending(sec,failure(sec,pri,flightAntartida)).
pending(sec,inform(sec,pri,flightItaly)).
```

If we query the system, we will learn that the agent pri still has the right to send any message and that agent sec can answer them by agreeing or refusing. In that sense, these rights are universal under the constitutive rules that conform the protocol. Furthermore, we can now test various facts at the point of the conversation characterized by the status of the sent and pending messages.

```
?- right(X,agree(X,Y,P)).

X = sec
Y = pri
P = flightChina ;

No
```

This query shows that at the current point of the system agent `sec` has the right to `agree` to satisfy the request to book a flight to China because this message is still pending. Since `agree` is one of the two possible answers to a `request`, if we query whether `sec` has the right to refuse a request, we would get the same answer. If we query whether some agent has the obligation to send an `inform` message, we get the following answer:

```
?- obligation(X,inform(X,Y,P)).

X = sec
Y = pri
P = flightItaly ;

No
```

In this case, the system concludes that since agent `X` has sent an `agree` message to a previous `request`, and that a notification of the result of the request is still pending, then agent `sec` has the obligation to perform an `inform` to report the result of the booking of the flight to Italy. Before we describe the FIPA Query-IF interaction protocols using our approach, we can show the explicit relation between Büchi automata our pragmatic approach. We have specified the rights of the principal in the request protocol by saying that, as long as the sender is the principal and the receiver is the secretary, the principal make requests about any content available in our small world. However, we can also propose an alternative representation of the principal's rights using Definite Clause Grammars. Thus, we can state that if agent `pri` is working *rightly*, then the set of legal speech acts that it can perform is captured by the following grammar:

```
rightsPrincipal --> principal,secretary,force,prop.

principal --> [pri].
secretary --> [sec].
force --> [request].
prop --> [flightCaribbean].
prop --> [flightAntartida].
prop --> [flightItaly].
prop --> [flightFinland].
prop --> [flightChina].
```

If we query the system asking all *right* speech acts that can be performed by agent `pri`, we get the following result:

```
?- rightsPrincipal(X,[]).

X = [pri, sec, request, flightCaribbean] ;

X = [pri, sec, request, flightAntartida] ;

X = [pri, sec, request, flightItaly] ;

X = [pri, sec, request, flightFinland] ;

X = [pri, sec, request, flightChina] ;
```

Similar grammars can be defined to specify the set of legal speech acts that allowed to be performed by the secretary. However, for the sake of simplicity of exposition, we will follow the form used in the Request Protocol.

We argued in the previous section that Prolog had a natural way to represent automata. Considering the grammar defined to specify the communicative rights of the principal, we can do the following query:

```
?- listing(rightsPrincipal).

rightsPrincipal(A, B) :-
        principal(A, C),
        secretary(C, D),
        force(D, E),
        prop(E, B).
```

Which represents the structure of an automaton as defined in the previous section.

Prolog Query-If Protocol

In the FIPA Query-IF interaction protocol, an agent X queries agent Y whether or not a proposition P is true. The receiver has the right to either *agree* or *refuse* to send and *inform* message providing an answer. In the case that agent Y agrees, then it has obligation to send a notification which can be an inform stating the truth of falsehood of the proposition P. If agent Y sends a refuse message the protocol ends there:

```
journalist(paxman).
politician(pm).

sender(X):-
  journalist(X).

receiver(X):-
  politician(X).

right(X,queryif(X,Y,P)):-
  sender(X),
  receiver(Y),
  content(P).

right(Y,agree(Y,X,P)):-
notrepliedagree(Y,queryif(X,Y,P)).

right(Y,refuse(Y,X,P)):-
notrepliedrefuse(Y,queryif(X,Y,P)).

obligation(Y,inform(Y,X,P)):-
sent(Y,agree(Y,X,P)),
pending(Y,inform(Y,X,P)).

obligation(Y,failure(Y,X,P)):-
sent(Y,agree(Y,X,P)),
```

```
pending(Y,failure(Y,X,P)).
```

6.6.4 Prolog Conversation Policies

Let us consider again the situation in which an agent paxman has the right to queryif a politician agent pm about the truth of the "peersmoney" scandal as long as we are not in electoral campaign. If we have an obligation rule then that states that an agent must perform an action before its applicability condition becomes false; a permission rule establishes that the agent can perform an action if its condition(s) is true.

Another example consider above was that of an agent Y acting on behalf of an airline company serving flights to European countries, that could have a policy that states that it should agree to every request regarding flight tickets to Europe (i.e., answering about flight times and providing the best offer for a potential buyer) and another one specifying that it has the obligation to refuse every request about flights to non European countries.

```
customer(customer).
seller(seller).

obligation(Y, agree(Y, X, P)) :-
    receive(request(X, Y, europeanFlight)).

obligation(Y, refuse(Y, X, P)) :-
    receive(request(X, Y, nonEuropeanFlight)).
```

We can then consider one of the NPRAG interaction protocols defined above and show an example in which conversation policies regulate agent communication. Thus, if we consider the NPRAG Query-If interaction protocol in which an agent paxman queries an agent pm on various issues, then we can add the following record of the linguistic exchange so far:

```
sent(pm,agree(pm,public,iraq)).
sent(pm,agree(pm,public,menezes)).

pending(pm,inform(pm,public,menezes)).
pending(pm,inform(pm,paxman,peersMoney)).
```

These facts combined with the constitutive rules provided by the interaction protocol produces the following answer to the question of which information messages has pm the obligation to answer:

```
?- obligation(pm,inform(X,Y,P)).

X = pm
Y = paxman
P = lebanon ;

X = pm
Y = public
P = menezes ;
```

The answer simply follows from the fact that at the current point of the system `pm` has sent an `agree` to both the content of both questions and that `pm` has not sent an `inform` yet. Note that under these two facts the agreement to `inform` the public of the truth about the `menezes` issue is not simply another state of the dialogue but it has become a sort of commitment that `pm` has to comply with. That is, saying that it agrees to talk about the truth of the menezes issue means that it will inform the public about that issue, and that it believe the content of the `inform`.

We can add as a policy that regulates `pm`'s behaviour with respect to queries directed to some `secret` content. Thus, if we query about the rights of the agents involved in the conversation, we get:

```
?- listing(right).

right(A, queryif(A, B, C)) :-
        sender(A),
        receiver(B),
        content(C).
right(A, agree(A, B, C)) :-
        notrepliedagree(A, queryif(B, A, C)).
right(A, refuse(A, B, C)) :-
        notrepliedrefuse(A, queryif(B, A, C)).
right(pm, notunderstood(pm, paxman, secret)) :-
        obligation(pm, hide(pm, public, secret)).

Yes
```

The result of this query means that agents have the right to perform `queryif` speech acts according to the constitutive rules of the conversation but also that there is an exception which states that agent `pm` has the right to send a `notunderstood` message if the topic of the conversation is about something that `pm` considers to be secret. This is a crucial point to help agents to achieve the rational effects of an speech act. For example, we can specify a rule to state that if an agent makes a promise to increase the taxes on air planes fuel, then it has the obligation to do so:

```
obligation(A,increaseTaxes(A,public,airplanesFuel)):-
  send(A,promise(A,public,airplanesFuel)).
```

Similarly, other conversation policies can be defined to state that an agent can deceive, or that it has the right to do so in particular circumstances. Thus, if an agent a arrives late to work but no one notices, then if asked by its boss b whether a arrived on time, b has the right to deceive b to save its job.

```
right(a,deceive(a,b,arrivalTime)):-
  arriveslate(a).
```

We can also specify that an agent X will always answer to every message it receives, etc.

```
obligation(reply(x,sa(y,x,p)):-
  receive(x,sa(y,x,p)).
```

The extension of our approach to other protocols and policies in the FIPA specification is fairly straightforward. Our approach shows how a well-defined normative concepts can be used to propose a high-level ACL pragmatics that are declarative, takes into account the context and that helps agents to achieve the perlocutionary effects of the speech acts. These two properties of the normative pragmatics, *contextual* and *perlocutionary*, fill in the last gaps in the list of requirements for ACLs discussed in section 3.2.5 and table 5.2. Next section offers a comparison to other approaches and discusses some short term future work necessary to improve the ongoing work presented in this paper. As a final note, the simplicity of the protocols and policies specified in this section was intentional. An important point for any future application of agent communication languages remains the proposal of high-level but simple ACL semantics and pragmatics.

6.7 Discussion

Our approach analyzes agent communication from a perspective in which the meaning of speech acts results of the combination of the semantic and pragmatic specifications. Every speech act is used in a specific context, and that context usually affects agents' communicative behaviour. We propose a normative pragmatics to stabilize interaction in agent communication. There have been quite a lot of work in the specification of ACL semantics: FIPA ACL (2002); Labrou and Finin (1994b); Singh (2000); Fornara and Colombetti (2004) and some other authors have proposed several interaction protocols to define the order in which messages are used (FIPA ACL, 2002; Fornara and Colombetti, 2004; Pitt and Mamdani, 1999; Endriss et al., 2004).

Two specification languages $MLTL_I$ and $NLTL_I$, provide the semantics for the ACL semantics and pragmatics. Unlike most of other approaches, our specification languages are grounded in a computational model. This would facilitates the verification of the ACL van der Hoek and Wooldridge (2003). We have shown how our proposal can be used to define pragmatic rules using a declarative language. Thus, the rules of NPRAG do not only consist of establishing the order of messages. By defining normative policies, we facilitate the fulfilment of the perlocutionary effects. This is done by specifying the rights and obligations of agents with respect to contextual information, role of the sender and content of messages. In this sense, our unified ACL performs better with respect to the requirements discussed than the other proposals.

Although it is possible to express some types of protocols by means of $NLTL_I$, we decided to specify both protocols and policies using Prolog. We believe that this facilitates its integration to different agent platforms where NPRAG rules can be taken as the norm base that rule agent communication. Furthermore, $NLTL_I$ not only provides a precise definition of the normative concepts used to specify NPRAG rules, but its value as a powerful grounded formalism to reason about normative agent systems is equally important. In this thesis, we have decided to show our proposal can be applied using Prolog, the fact that we have a system defined in terms of interpreted systems and $NLTL_I$ allows us to separate the specification and implementation processes.

The characterization of roles is inspired by the work done on organizational concepts Ferber and Gutknecht (1998); van der Torre et al. (2004), and they are adapted for their use in ACL pragmatics. Other authors (Dignum and Kuiper (1997)), have also presented temporal deontic logic with dynamic operators, but we believe that the combination of deontic, dynamic and temporal notions results in highly complex logics. $NLTL_I$ is far simpler that those formalisms and therefore is easier to axiomatize.

In a very recent paper, Boella et al. (2006) present a role-based approach to ACL semantics. They intend to make the ACL semantics public by attributing mental states to social roles instead of agents. Thus, there are two sets of beliefs, those that are public and are ascribed to roles, and those that are private and belong to the agents' private mental states. In our view, having two sets of cognitions seems to make things unnecessarily complex. A role is constrained by a set of social rules (rights, obligations, permissions, etc.) that define the expected behaviour of any agent playing the role. These social rules may or may not conflict the private beliefs and goals of agents. In any case, even if beliefs and goals are attributed to roles, agents playing a role would still need to reason about their beliefs and goals. An additional issue is that the semantics of beliefs and goals is not grounded. From a semantic point of view, defining the ACL semantics in terms of roles makes the semantics less general, since the meaning of speech acts would be affected by agents' role. For example, two roles that are considered are those of *speaker* and *receiver*.

Our proposal is also related to Kagal et al. (2003), but there are several differences: First, they do not provide a formal definition for any of the deontic operators they use. Second, they claim that policies are independent of the ACL semantics, and that in fact policies should be specified in the general structure of the system. Third, the use of obligations produces policies that in some cases could be too restrictive for autonomous agents. Finally, they use an ontology language based on OWL as the policy specification language, but we believe that logic is a more suitable language to reason about multi-agent systems.

For standardization reasons, our proposal intends to be as close as possible to the FIPA ACL specification. With this purpose in mind, we have provided definitions for the actions absent in the FIPA Communicative Actions Library: Commissives and declaratives. We think that in FIPA CAL some of the definitions are unnecessarily complex. This is partially due to the multimodal language used as the semantic language. Current and future work involves further development of protocols and policies in Prolog and studying different verification techniques based on existing temporal logic algorithms.

7

Conclusion

We believe that this thesis offers a new approach to agent communication where the meaning of speech acts consists of the combination of the semantic specification and the NPRAG rules that constrain their use.

First, it clearly distinguishes semantics and pragmatics of the language. Semantically, it offers a computationally grounded specification language based on $MLTL_I$. This enables to define meaningful and public communicative actions. Regarding the pragmatics, it presents a procedure using normative rules to help agents to understand messages when involved in conversations. Besides, it also helps to regulate the use of the communicative actions defined in the semantics. Unlike research in ACL semantics, there are not many works that attempt to capture both aspects of communication in the same framework. Terminology and theoretical approaches are still being worked out, formal approaches are still fairly unsettled, and the role of research in natural language pragmatics and discourse theory is still evaluated.

Although some previous work has discussed the social and mental aspects of communication, this thesis presents a coherent combination between the two components. By doing so, it is possible to develop a semantics that can serve the needs of agents operating and communicating in open environments. It has been shown how a high-level social semantics can be specified capturing at the same time the intuitive meaning of communication, and how pragmatics can constraint the use of the actions depending of the scenario. Thus, considering the list of requirements for ACLs discussed throughout the thesis, we can see that the unified ACL match them quite well. After the semantics of the language was specified in chapter 5, NPRAG aims were to produce a pragmatic theory that would consider how contextual information constrains agents' behaviour, and how proposing normative rules for the use of speech acts facilitate the achievement of the perlocutionary effects.

Requirements	**ACLs**
	unified ACL
Autonomous	✓
Complete	✓
Contextual	✓
Declarative	✓
Formal	✓
Grounded	✓
Public	✓
Perlocutionary	✓

Table 7.1. Requirements for ACLs.

1. **Autonomous**: The ACL semantics (SAL) do not completely fix agents communicative behaviour because the fulfillment of the perlocutionary effects are left to the ACL pragmatics. The ACL pragmatics (NPRAG) uses a normative notion of rights, which allows us to specify agents' freedom instead of pre-determining their communicative behaviour.
2. **Complete**: We have defined a complete set of speech acts, understanding "complete" as representing every category in Searle's taxonomy. Searle's taxonomy is by no means a closed list; one could imagine a more fine-grained taxonomy including more systematic distinctions between types of directives such as yes/no questions, prohibitives, etc. This thesis leads the way in including the commissive and declarative categories, notably absent in some other approaches Labrou and Finin (1997); FIPA ACL (2002).
3. **Context**: An important contribution of the normative pragmatics consists of its account of contextual information to specify agents' communicative behaviour. In agent communication contextual factors include the role that agents play in the application scenario, the delegated tasks agents try to achieve, the propositional content of messages, and the record of previous exchanges. In chapter 6, we modelled these contextual factors by means of normative concepts that can be embedded in a normative multi-agent system. The use of normative concepts to model ACL pragmatics keep to a minimum agents' reasoning about each others' mental states. In that sense, it is more *efficient*. Furthermore, by avoiding that reasoning, the specification of conversation protocols and policies is greatly *simplified*. The nor-

mative concepts defined for its use in the communication protocol benefit from the properties of $NLTL_I$, a temporal and normative logic defined upon an interpreted system. Furthermore, $NLTL_I$ not only presents interesting formal properties, but it is designed to facilitate pre-runtime verification.

4. **Declarative**: By providing a declarative definition of ACL semantics and pragmatics, specifying what the meaning is instead of a follow-the-rule low-level procedure, the resultant unified ACL is a high-level language suitable to express agents' goals, beliefs and other notions such as promises or declarations. Defining a declarative pragmatics represents a strong point of this thesis.
5. **Formal**: The unified ACL is specified using two formal logics, $MLTL_I$ and $NLTL_I$ that describe the evolution of a multi-agent systems with respect to the agents' beliefs, goals, intentions, obligations and rights. A particular care was to provide an external interpretation of beliefs, goals and intentions in a way that those attitudes would refer states of a system instead of private mental states of the agents. In doing so, we were paving the ground provide a semantics and pragmatics suitable for verification, as it was discussed in section 6.6. Two more properties discussed below are closely related to verification, namely, *public* and *grounded*.
6. **Grounded**: In chapter 4 we introduced the notion of interpreted system which is based on the work done by Fagin et al. (1995). The novelty of our approach was not to restrict ourselves to knowledge and time, but to define two specification languages in $MLTL_I$ and $NLTL_I$ for the semantics and the pragmatics respectively, which were grounded upon the Interpreted System previously defined. The benefit of this approach is to be able to easily relate the semantics of our logics to a computational model. Moreover, the use of a temporal logic allows us to provide more natural and simple speech acts specification (see section 4.2.1 and A).
7. **Public**: We claim that the illocutive/intentional aspect of communication should be preserved in the ACL semantics. However, we acknowledge that using BDI approaches based on modal logics to model agent communication do not result in public and verifiable ACLs. The lack of publicity is clear in the fact that in those formalisms, the satisfiability conditions of the semantics depend on private and unverifiable mental states. As an alternative, chapter 5 proposes an external interpretation of the cognitive concepts by relating them to states of agents in a system. In addition to the previous

point, using norms for the ACL pragmatics also makes the ACL pragmatics public since we do not need to reason about agents private mental states.

8. **Perlocutionary**: NPRAG rules in the form of protocols and policies enable agents to achieve the perlocutionary effects by specifying obligations and rights on the participants. Traditionally, the perlocutionary effects were accounted for in the ACL semantics, which was too restrictive for agents' autonomy. An alternative approach consists in not worrying at all about the perlocutionary effects, which means that we do not get agents to act upon receiving messages. This thesis presents a novel approach based on a normative pragmatic theory for agent communication, NPRAG, in which the perlocutionary effects of the speech acts are to be fulfilled by the protocols and policies that rule conversations. In order to preserve agents' autonomy, we offer a notion of right which specifies agents' behaviour when acting *rightly*.

We should also add that we have studied the applicability of our pragmatic approach by presenting examples of protocols and policies coded in Prolog. Additionally, it has been argued how the semantics of $NLTL_I$ can be verified in various ways. This has been achieved by translating $NLTL_I$ structures into finite state automata, which are widely used in verification via model-checking. An extra benefit of it is the fact that Prolog have built-in features to implement automata (for example, DCGs). A short term aim should be to provide a verification of $NLTL_I$ $MLTL_I$. Current work on pre-runtime verification of complex formal logics (Raimondi and Lomuscio, 2004) looks very promising. Furthermore, it would be interesting to produce more sophisticated implementations of conversation protocols and policies in a manner that they could be integrated with platforms such as 3APL and BOID Dastani et al. (2004); van der Torre et al. (2004). These development would include pragmatic principles to treat deception, sincerity, etc.

Deontic concepts are increasingly used in the specification and verification of multi-agent systems. It is unrealistic to assume that a whole open multi-agent system may be controlled by the same vendor. Thus, this makes it difficult to verify agents' conformance with the set of semantic and pragmatic specifications of ACLs. In this sense, by adopting a normative point of view, it seems more sensible to leave the theoretical possibility of agents violating the norms. We can then use the formal language provided to reason about the consequences that result from those violations. In this sense, having a separate specification language (from the implementation language) allows us to reason about external properties of the system. Further work on these issues would include the definition of more normative notions to complement *right* which may be more suitable to specific circumstances, and to embed our ACL in a normative multi-agent system.

If it is considered that agents are rational, and that they can access formal specification of protocols, there must be a way for the agent to automatically produce an instantiation of its own. This goal may be still remote but we believe that some steps have been taken in this thesis towards reaching it. This work should not seen as incompatible with other approaches such as FIPA. Conversely, this thesis hopes to have contributed to show what remains to be done to improve current standards in agent communication languages.

A

Appendix A. Speech Acts Library

In this Annex we present a definition of every communicative act defined in the FIPA CAL using our semantic specification language $MLTL_I$. The speech acts are classified into assertives, directives, commissives and declaratives. Each speech acts is accompanied by a brief description of its meaning in English.

A.1 Assertives

We start with the *accept-proposal* speech act. According to the FIPA CAL, this is speech act is a general purpose acceptance of a proposal made by performing a *propose* act. When an agent i sends an *accept-proposal* message it is informing the receiver j that i intends that (at some point in the future) j will perform the action, once the given precondition is true. FIPA uses a dynamic operator *Done* to express the time between the precondition and the proposition to be achieved. We believe that we can represent this idea in a more natural way using the temporal operators defined in $MLTL_I$.

$\langle i, accept-proposal(j,\phi)\rangle \equiv$
$\langle i, inform(j, G_i(I_j phi\ U\ \psi))\rangle$
$FP : G_i(I_j\phi\ U\ \psi)$
$RE : B_j(G_i\phi\ U\ \psi)$

Table A.1. Accept Proposal.

In our definition of *accept-proposal*, the sender i does not intend that someone brings about ϕ when the precondition ψ for the proposal is true. Instead, agent i has the goal along a run that j will bring about ϕ while ψ is true. Note that the until operator allows us to better express the idea of "some point in the future" as well as the idea that ϕ is true while (until) some precondition ψ is true. Literally, our definition states that agent i sends an *accept-proposal* message to agent j that there is a run at some point in the future where, if the precondition of the proposal ψ holds, then ϕ holds. This is equivalent to agent i informing j that i has the goal that j intends along a run that there is run where ϕ until ψ holds. The preconditions state that i has the goal that j intends to bring about the proposal ϕ where the preconditions ψ hold. As an effect, j believes that i has the goal that if ψ holds, then ϕ will eventually hold.

$\langle i, agree(j, \phi)\rangle \equiv$
$\langle i, inform(j, I_i\phi\ U\ \psi)\rangle$
$FP : I_i\phi\ U\ \psi$
$RE : B_j(I_i\phi\ U\ \psi)$

Table A.2. Agree.

If an agent i agrees with agent j to bring about a proposition ϕ, this is equivalent to agent i informing agent j that i intends along a run that ϕ will eventually hold until the precondition ψ holds. The FPs before sending an *agree* state that agent i has to intend for some run that ϕ until ψ eventually holds. The Rational Effects need agent j to believe that agent i intends to do so.

The next assertive is *cancel*. In the description of *cancel* provided by FIPA, this speech act allows an agent i to inform another agent j that i no longer intends that j performs a previously requested action. Besides, it is made clear that *cancel* is simply used to let an agent know that i no longer has a particular intention, it is not an order that j stops performing an action. This perlocutionary effect is not guaranteed by the semantics.

$\langle i, cancel(j, \phi)\rangle \equiv$
$\langle i, disconfirm(j, G_i(I_jF\phi))\rangle$
$FP : \neg G_i(I_jF\phi) \wedge B_j(G_i(I_jF\phi))$
$RE : B_j\neg G_i(I_jF\phi)$

Table A.3. Cancel.

Thus, an agent i cancel some (previously requested) proposition ϕ is equivalent to i disconfirming j that i has the goal along a run that j has the intention to eventually bring about ϕ. The preconditions require that i does not have the goal that j intends to eventually make ϕ true and that j actually believes that i has the goal that j intends that ϕ is eventually made true. The effect is that j comes to believe that i does not have the goal that j does so.

$\langle i, confirm(j, \phi)\rangle$
$FP : B_i(\phi) \wedge B_i(B_j F\phi \vee B_j F\neg\phi))$
$RE : B_j\phi$

Table A.4. Confirm.

An agent i *confirms* to an agent j that ϕ holds. The preconditions state that i believes that ϕ is true and that i believes that j either believes that ϕ will or will not eventually hold along a run. As an effect, agent j believes that ϕ is true.

$\langle i, disconfirm(j, \phi)\rangle$
$FP : B_i\neg\phi \wedge B_i(B_j F\phi \vee B_j F\neg\phi)$
$RE : B_j\neg\phi$

Table A.5. Disconfirm.

Disconfirm is the dual of *confirm*. Thus, an agent i disconfirms to agent j that ϕ holds. The preconditions are that i believes that ϕ is not true, and that ϕ will or will not eventually hold at some run. The effect is that agent j believes that ϕ is not true.

$\langle i, failure(j, \phi)\rangle \equiv$
$\langle i, inform(j, (I_i\phi\ U\ \psi)) \wedge \neg\psi \wedge G\neg\phi)\rangle$
$FP : B_i\phi\ U\ \psi$
$RE : B_j\neg(\phi\ U\ \psi)$

Table A.6. Failure.

An agent i sends a *failure* message to agent j stating that the bringing about of ϕ has failed. This is an abbreviation for i informing j that i intended that ϕ be true until ψ. The proposition ψ is taken to be the condition for ϕ to be true. Thus, if ϕ cannot be made true globally in a run, the reason is that ψ is not true either. The preconditions for this act is that i believes that there is a run where if ψ holds then ϕ eventually holds, and the effect is that j comes to believe that ϕ does not hold.

$\langle i, inform(j, \phi)\rangle$
$FP : B_i(\phi) \wedge G_i(B_j(\phi))$
$RE : B_j\phi$

Table A.7. Inform.

Agent i informs j that a proposition ϕ holds at point in the system. The preconditions for i to send an *inform* require i to believe that ϕ holds and to have the goal of j believing that ϕ holds. The Rational Effects are that agent j believes that ϕ is true.

$\langle i, inform-if(j, \phi)\rangle \equiv$
$\langle i, inform(j, \phi)\rangle \vert \langle i, inform(j, \neg\phi)\rangle$
$FP : (B_i\phi \vee B_i\neg\phi) \wedge G_i(B_j\phi \vee B_j\neg\phi)$
$RE : B_j\phi \vee B_j\neg\phi$

Table A.8. Inform-if.

As in the FIPA CAL, *inform-if* is a macro act which can be planned an requested but cannot be directly performed. In this case, agent who enacts an *inform-if* will actually perform an *inform* act whose content will be either ϕ or $\neg\phi$.

$\langle i, inform-ref(j, \phi)\rangle \equiv$
$\langle i, inform(j, \phi)\rangle$
$FP : B_i(\phi) \wedge G_i(B_j\phi)$
$RE : B_j(\phi)$

Table A.9. Inform-ref.

Inform-ref is also a macro speech act. In this case, the object of the act is a referential expression ϕ which is quite often a definite description of an object. We include this speech act for completeness reasons although we believe that this act overlaps with *inform*. Since $MLTL_I$ is a propositional logic, we do not distinguish whether a proposition ϕ is a definite or indefinite description, that is, whether they are referential expressions or not. Atomic propositions ϕ and ψ describe facts about the system.

Agent i sending a *non-understood* message means that i did not understand the meaning of a message ϕ, where ϕ can be a speech act itself. If ϕ is the condition for i to act and bring about some proposition ψ, then *non-understanding* ϕ means that agent i does not bring about ψ along the run. The preconditions

$\langle i, not - understood(j, \phi)\rangle \equiv$
$\langle i, inform(j, (\neg\psi\ U\ \neg\phi))\rangle$
$FP : B_i\neg\psi\ U\ \neg\phi)$
$RE : B_j(B_i\neg\psi\ U\ \neg\phi))$

Table A.10. Not-understood.

establish that agent i does not believe that ϕ eventually holds, and therefore ψ cannot hold either. The effect is that agent j believes that agent i believes so.

$\langle i, propose(j, \phi)\rangle \equiv$
$\langle i, inform(j, I_i\phi\ U\ \psi))\rangle$
$FP : I_i\phi\ U\ \psi$
$RE : B_j(I_i\phi\ U\ \psi)$

Table A.11. Propose.

An agent i sends a *propose* message stating that it will bring about a proposition ϕ under certain preconditions ψ. This is equivalent to agent i informing j that i intends along a run that if ψ holds, then ϕ (the proposal) also holds. The preconditions require that agent i has that intention and the perlocution states that the receiver j believes that agent i intends to do so.

$\langle i, proxy(j, \phi)\rangle \equiv$
$\langle i, inform(j, G_i(I_j X\phi))\rangle$
$FP : G_i(I_j X\phi)$
$RE : B_j(G_i(I_j X\phi))$

Table A.12. Proxy.

The *proxy* speech act states that an agent i wants an agent j to have the intention that ϕ holds at next state, where ϕ consists of a speech act that j has to send to other agents in the system (these agents are usually selected according to their role in a particular scenario).

$\langle i, refuse(j, \phi)\rangle \equiv$
$\langle i, inform(j, \neg I_i F\phi)\rangle$
$FP : \neg I_i F\phi$
$RE : B_j(\neg I_i F\phi)$

Table A.13. Refuse.

An agent i *refusing* to j to bring about ϕ is equivalent to agent i informing agent j that i does not intend (along a run) that ϕ eventually holds. The preconditions before performing this act establish that that i does not intend along a run that ϕ eventually holds. As the Rational Effect, agent j believes that agent i does not intend to do so.

$\langle i, reject - proposal(j, \phi)\rangle \equiv$
$\langle i, inform(j, \neg G_i(I_j\phi\ U\ \psi))\rangle$
$FP : \neg G_i(I_j\phi\ U\ \psi)$
$RE : B_j(\neg G_i\phi\ U\ \psi)$

Table A.14. Reject-Proposal.

Reject-proposal is effectively the dual of *accept-proposal*, so this speech act states that i does not have the goal that j intends along a run to bring about ϕ when ψ holds. See the definition of *accept-proposal* above for more details.

A.2 Directives

$\langle i, cfp(j, \phi)\rangle \equiv$
$\langle i, query - ref(j, G_i(I_jF\phi))\rangle$
$FP : G_i(I_jF\phi)$
$RE : F\phi$

Table A.15. Call for Proposal.

The argument of the cfp performative should be usually a *propose* speech act. Thus, sending a *cfp* abbreviates for sending a *query-ref* to j so that i has the goal that j intends along a run that ϕ (the result of sending a *propose*) is true.

$\langle i, propagate(j, < i, cact, n >)\rangle \equiv$
$\langle i, inform(j, G_i(I_jF < i, cact, n >))$
$FP : FP(cact) \wedge G_i(I_j < i, cact, n >)$
$RE : F < i, cact, n >$

Table A.16. Propagate.

Propagate is a forwarding speech act. The argument of a propagate performative is a speech act addressed to a number of agents n. We can substitute

$< i, cact, n >$ by a proposition ϕ which describes the the proposition which j has to bring about, that is, that j has propagated some message *cact* from i.

$\langle i, query - if(j, \phi)\rangle \equiv$
$\langle i, request(j, \langle j, inform - if(i, \phi)\rangle)\rangle$
$FP : G_i(I_j F(\langle j, inform - if(i, \phi)\rangle))$
$RE : I_j F(\langle j, inform - if(i, \phi)\rangle))$

Table A.17. Query-if.

An agent i sending to j a *query-if* message is an abbreviation to agent i requesting agent j to send to i an *inform-if* that ϕ. The preconditions state that i has the goal that j intends along a run that eventually j will send such a message, and the perlocutionary effects result in agent j intending to do so.

$\langle i, query - ref(j, \phi)\rangle \equiv$
$\langle i, request(j, \langle j, inform - ref(i, \phi)\rangle)\rangle$
$FP : G_i(I_j F(\langle j, inform - ref(i, \phi)\rangle))$
$RE : I_j F(\langle j, inform - ref(i, \phi)\rangle))$

Table A.18. Query-ref.

Query-ref is analyzed as *query-if* but now the proposition ϕ is usually a definite description of an object or an agent.

$\langle i, request(j, \phi)\rangle$
$FP : G_i(I_j F\phi)$
$RE : F\phi$

Table A.19. Request.

When an agent i requests that agent j brings about some proposition ϕ the preconditions to be satisfied consist of agent i having the goal that agent j intends along a run that ϕ eventually holds at that run. The rational effect is that ϕ eventually holds.

$\langle i, request - when(j, \phi))\rangle \equiv$
$\langle i, inform(j, (G_i(I_j\phi\ U\ \psi) \rightarrow F\phi)\rangle$
$FP : G_i(I_j\phi\ U\ \psi)$
$RE : I_j\phi\ U\ \psi$

Table A.20. Request-When.

A *request-when* message is used when an agent i wants to express to agent j that as soon as some precondition ψ holds, i has the goal along a run that if j intends to bring about ϕ until ψ then, ϕ will eventually hold.

$\langle i, request - whenever(j, \phi)\rangle \equiv$
$\langle i, inform(j, (G_i(I_j\phi\ U\ \psi) \rightarrow G\phi)\rangle$
$FP : G_i(I_jG\phi)$
$RE : I_jG\phi$

Table A.21. Request-Whenever.

This speech act is formalized as *request-when* but the the proposition holds globally.

$\langle i, subscribe(j, \phi)\rangle$
$\langle i, request - whenever(j, inform - ref(i, \phi))\rangle$
$FP : G_i(I_jG(\langle j, inform - ref(i, \phi))\rangle)$
$RE : I_jG(\langle j, inform - ref(i, \phi))\rangle$

Table A.22. Subscribe.

Agent i performs a *subscribe* action means that i *requests-whenever* that j sends an *inform-ref* that ϕ, where ϕ usually is a definite description of an object. *Subscribe* is a universal version of *query-if.*

A.3 Commissives

$\langle i, promise(j, \phi)\rangle$
$FP : I_i F\phi$
$RE : F\phi$

Table A.23. Promise.

For an agent i to promise to agent i that ϕ will be true agent i has to intend along a run that ϕ will eventually hold. The perlocutionary effect state that ϕ eventually holds indeed.

A.4 Declaratives

$\langle i, declare(j, \phi)\rangle$
$FP : G_i(X\phi)$
$RE : X\phi$

Table A.24. Declare.

For an agent i to declare to agent j that ϕ is true, the preconditions of the act require that i intends along a run that ϕ holds in the next immediate global state. The perlocutionary effect to be achieved is that ϕ holds at the next state.

q

B

Appendix B. Proofs

In this annex, we present some well-known proofs that make our framework more self-contained.

B.1 Validity of $KD45_n$

K $If(M,s) \models \Box \wedge \Box(\phi \rightarrow \psi)$ then for all states t that $(s,t) \in \mathcal{B}_i$, we have both that $(M,t) \models \phi$ and $(M,t) \models \phi \rightarrow \psi$. By the definition of $\models$, we have that $(M,t) \models \psi$ for all such t, and therefore $(M,s) \models \Box\psi$.

D $If(M,s) \models \Box\phi$, then for all t such that $(s,t) \in \mathcal{B}_i$, we have $(M,t) \models \phi$. Since $\mathcal{B}_i$ is serial, there is a state t such that $(s,t) \in \mathcal{B}_i$ we have $(M,t) \models \phi$. Hence, $(M,t) \models \phi$, i.e., $(M,s) \models \neg\Box\neg\phi$. Therefore, axiom D is valid in the class of serial models.

4 Suppose that $(M,s) \models \Box\phi$. Consider any t such that $(s,t) \in \mathcal{B}_i$ and any u such that $(t,u) \in \mathcal{B}_i$. Since $\mathcal{B}_i$ is transitive, we have $(s,u) \in \mathcal{B}_i$. Since $(M,s) \models \Box\phi$, it follows that $(M,u) \models \phi$. Thus, for all t such that $(s,t) \in \mathcal{B}_i$, we have $(M,t) \models \Box\phi$. It now follows that $(M,s) \models \Box\Box\phi$. Therefore, axiom 4 is valid in the class of transitive models.

5 Let us assume that $(M,s) \models \neg\Box\neg\phi$. Then for some $t \in S$ such that $(s,t) \in \mathcal{B}_i$, we must have $(M,t) \models \phi$. Suppose that $(s,u) \in \mathcal{B}_i$. Then, we have $(s,u) \in \mathcal{B}_i$ and $(s,t) \in \mathcal{B}_i$, and therefore, by euclideanness, we have $(u,t) \in \mathcal{B}_i$. So, for any u such that $(s,u) \in \mathcal{B}_i$ there exists a t such that $(u,t) \in \mathcal{B}_i$ and $(M,t) \models \phi$. Therefore, for any u such that $(s,u) \in \mathcal{B}_i$ we have $(M,u) \models \neg\Box\neg\phi$. Thus, $(M,s) \models \Box\neg\Box\neg\phi$. Hence, axiom 4 is valid in the class of euclidean models.

NEC If $M \models \phi$, then $(M,t) \models \phi$ for all states t in M. For any state s in M, it follows that $(M,t) \models \phi$ for all t such that $(s,t) \in \mathcal{B}_i$. Therefore, $(M,s) \models \Box\phi$ for all states s in M, and hence $M \models \Box\phi$.

B.2 Soundness and Completeness of $KD45_n$

(a) We consider first soundness. Soundness is quite easy to prove because we show that each of the axioms correspond to seriality, transitivity and euclideanness. For axiom D, we need to show that D_n is valid in all the structures where accessibility relation is serial. Suppose that agent i believes ϕ in a state s, $B_i\phi$, so we need to show that $\neg B_i \neg\phi$. This means that by seriality ϕ is true in every $s' \in s$ such that there is an s', $(s,s') \in \mathcal{B}_i$. Hence, $M_{ste}, s' \models \phi$, i.e, $(M_{ste}, s) \models \neg B_i \neg\phi$, that is, $M_s, s \models B_i\phi \rightarrow \neg B_i \neg\phi$. It follows that every serial structure satisfies all the axioms of D_n, and thus is a model of D_n. Consequently, D_n is sound with respect to M_{ste}.

For axiom 4, we need to show that 4_n is valid in all structures where the accessibility relation is transitive. If agent i believes ϕ in s, $(M_{ste}, s) \models B_i\phi$, then $M_{ste}, u \models \phi$ for any t such that $(s,t) \in \mathcal{B}_i$ and any u such that $(t,u) \in \mathcal{B}_i$. Therefore, we have that $(M_{ste}, t \models B_i\phi$ for all t such that $(s,t) \in \mathcal{B})i$ and from here it follows that $(M_{ste}, s) \models B_iB_i\phi$.

Finally, for axiom 5 we need to focus on the euclidean relation. If agent i does not believe ϕ is false in s, $M_{ste}, s \models \neg B_i \neg\phi$, then ϕ is true in an state t, $M_{ste}, t \models \phi$, such that $(s,t) \in \mathcal{B}_i$. If we further suppose that there is an state u such that $(s,u) \in \mathcal{B}_i$ in a structure where the accessibility relation is euclidean M_{ste}, which means that for any u such that $(s,u) \in \mathcal{B}_i$ there is a t such that $(u,t) \in \mathcal{B}_i$ for which ϕ is true, $(M_{ste}, t \models \phi$. Hence, $(M_{ste}, u) \models \neg B_i \neg\phi$. If that is the case, then $(M_{ste}, s) \models B_i \neg B_i \neg\phi$. From this it follows that $(M_{ste}, s) \models \neg B_i \neg\phi \rightarrow B_i \neg B_i \neg\phi$. We can now prove completeness.

(b) It would be quite convenient if every structure that satisfies all instances of axioms $KD45_n$ are in M_{ste}. Although this is not the case, axioms $KD45_n$ force the accessibility relations in the canonical structures to be serial, transitive and euclidean, which give us enough ground to prove that the axiom system $KD45_n$ is complete with respect to M_{ste}.

Let C be a class of structures. A system S is complete with respect to M if for any set of formulae $\Gamma \cup \phi$, if $\Gamma \models_M \phi$, then $\Gamma \vdash_S \phi$. The system S is complete with respect to M iff:

(1) For every formula ϕ, if $\nvdash_S \phi$, then there is a model M in which for some $g \in G$, $M \nvDash_g \phi$. We say that a formula ϕ is inconsistent in S iff $\vdash_S \neg\phi$, and it is consistent iff $\nvdash_S \neg\phi$.
(2) For every formula ϕ, if ϕ is consistent in the system S then there is a model M in which $M \models_g \phi$ for some $g \in G$.

If (2) is true, then (1) is also true. By saying $\nvdash_S \phi$ we mean that $\neg\phi$ is S-consistent. Thus, if (2) holds, then there is a model such that $M \models \neg\phi$. This means that $M \nvDash \phi$. Therefore, (1) is true.

The method of the canonical models shows completeness by showing (2). For every S-consistent formula ϕ, there is a class of models that verifies it. If we consider a set of formulae $\Gamma = (\phi_1, \phi_2, \ldots, \phi_n)$, then Γ is S-consistent iff $\nvdash_S \neg(\phi_1 \wedge \phi_2 \wedge \ldots \wedge \phi_n)$. If Γ is not finite, then we say that every finite subset of Γ is S-consistent. In other words, there is not a finite subset of Γ such that $\vdash_S \neg(\phi_1 \wedge \phi_2 \wedge \ldots \wedge \phi_n)$. But this is saying something stronger that statement (2) above. That is, we are saying that

(3) If Γ is a S-consistent set of formulae, then there is a class of models M such that if $\phi \in \Gamma$, then $M \models \phi$.

In this case, Γ is a maximal (and consistent) set of formulae. This means that Γ es maximal iff for every $\phi \in \Gamma$, we have $\phi \in \Gamma$ or $\neg\phi \in \Gamma$.

Lemma 1. *Let Γ be a consistent-maximal set of formulae. This set presents the following properties: For every formula ϕ,*

- *we have that either $\phi \in \Gamma$ or $\neg\phi \in \Gamma$.*
- *$\phi \vee \psi \in \Gamma$ iff $\phi \in \Gamma$ or $\psi \in Gamma$.*
- *$\phi \wedge \psi \in \Gamma$ iff $\phi \in \Gamma$ and $\psi \in \Gamma$.*
- *If $\vdash_S \phi$ then $\phi \in \Gamma$.*
- *If $\phi \in \Gamma$ and $\phi \rightarrow \psi \in \Gamma$, then $\psi \in \Gamma$.*
- *If $\phi \in \Gamma$ and $\vdash S\phi \rightarrow \psi$, then $\psi \in \Gamma$.*

Theorem 4. *Every consistent set of formulae can be extended into a consistent-maximal set of formulae in a system. Thus, suppose that Δ is a consistent set of formulae. Then, there is a consistent-maximal set of formulae Γ such that $\Delta \leq \Gamma$ (which is true for every system that contains Propositional Logic).*

Suppose that Δ is a set of formulae of Propositional Modal Logic (PML). We write $\Box^-(\Delta)$ to denote the set of every formulae ϕ such that $\Box\phi \in \Delta$.

$$\Box^-(\Delta) = \phi : \Box\phi \in \Delta$$

Lemma 2. *Let S be a normal modal system, and let Δ be a consistent set of formulae that contains $\neg\Box\phi$. Then, we say that $\Box^{-}(\Delta) \cup \neg\phi$ is S-consistent.*

Lemma 3. *Let S be a normal modal system and let Δ be a S-consistent set of formulae that contains $\Diamond\phi$. Then, we say that $\Diamond^{-}(\Delta) \cup \phi$ is S-consistent.*

We can now define the concept of canonical models. For completeness, we need to show that every S-consistent formula is true at some state g of the canonical model. If we can show that for some normal modal system, that its canonical model is a model M of a particular class, then we are proving that S is complete with respect to the class of models M. In other words, we need to show that for every S-consistent formula there is some model M that verifies it. Given that the states in a canonical model for a system of modal logic will always verify just the formula that they contain, it follows that the formula that are true in such a model are precisely the theorems of the system.

Definition 39 (Canonical Model). *Let S be some PML normal system. Its canonical model Mc is defined as follows:*

1. *G^S is the set of all maximal S-consistent set of formulae.*
2. *R^S is the binary relation on G^S. For every g, $g' \in G^S$, $g\mathcal{R}^S g'$ iff $\Box^{-}(g) \leq g'$.*
3. *For some variable ϕ and some $g \in G^S$, $Mc \models_G^S \phi$ if $\phi \in G^S$. Otherwise, $Mc \not\models_G^S \phi$.*

To show that every S-consistent formula is true at some Mc of S, we show that for some Mc, every formula is true at some $g \in G^S$ if $\phi \in g$.

Theorem 5. *Let Mc be a canonical model for some PML normal system. For every formula ϕ and every $g \in G^S$, $Mc \models_g \phi$ iff $\phi \in G^S$.*

Proof.

- *ϕ follows from the definition of Mc.*
- *For the negation, we need to show that $Mc \models_g \neg\phi$ iff ϕ is not in G^S. Thus, ϕ is not in G^S iff $\neg\phi \in G^S$.*
- *For the conditional, we have that $Mc \models_g \phi \rightarrow \psi$ iff $\phi \rightarrow \psi \in G^S$. Thus, we have that either $\neg\phi \in G^S$ or that $\psi \in G^S$, that is, $\neg\phi \vee \psi \in G^S$, which is equivalent to $\phi \rightarrow \psi \in G^S$.*
- *Regarding the modality operator, we have that $Mc \models_g \Box\phi$ iff $Mc \models_g' \phi$ for every g' such that $g\mathcal{R}^S g'$. This means that $\phi \in g'$ for every g' such that $Box^{-}(g) \leq g'$. Suppose that $\Box\phi$ is not in g. Then, ϕ is not in $\Box^{-}(g) \leq g'$, which means that ϕ is not in g'. Which leads to a contradiction.*

Corollary 1. *Every formula ϕ is valid in a canonical model Mc of a system S iff $\vdash \phi$.*

$$Mc \models_g \phi \text{ iff } \vdash_S \phi.$$

We say for every g' if $\phi \in g$, then for every $g \in G^S \vdash_S \phi$ (is a member of every maximal consistent set).

Lemma 4. *For any system S and any state $g \in G^S$, if $\Diamond\phi \in g$, then there is a state $g' \in G^S$ such that $(g, g') \in \mathcal{R}^S$ and $\phi \in g'$.*

This lemma is called the Existence Lemma (see Blackburn et al. (2001)).

We can now apply this result to specific systems. For example, to system $KD45_n$. We can show its completeness if we show that its canonical model is a model of the class of specified models, that is, those that are serial, transitive and euclidean. We say that for every formula ϕ such that $\nvdash_S \phi$ there is some $g \in G^S$ such that $Mc \not\models_g \phi$ iff ϕ not in g. Thus, if the canonical model Mc of S is model of a class C, then exists a C-model in which $Mc \not\models_g \phi$ (by the previous corollary). In practice, the completeness of the system $KD45_n$ is now reduced to prove that the canonical model of $KD45_n$ is a serial, transitive and euclidean model.

Proof.

1. The system is complete with respect to the models where the accessibility relation is serial. For every $g \in G^S$ there is a $g' \in G^S$ such that $(g, g') \in \mathcal{R}^S$. Since the relation is serial, the model contains $\Box\phi \to \Diamond\phi$. By closure, under uniform substitution it contains $\Box\top \to \Diamond\top$. By generalization, $\Box\top$ belongs to the model and by modus ponens, $\Diamond\top \in g$. Finally, by the Existence Lemma, $g\mathcal{R}^S g'$.
2. For transitivity, we have that for every $g, g', g'' \in G^S$ if $g\mathcal{R}^S g'$ and $g'\mathcal{R}^S g''$, then $g\mathcal{R}^S g''$. Which means that if $\Box^-(g) \leq g'$ and $\Box^-(g') \leq g''$, then $\Box^-(g) \leq g''$. We need to show that $g\mathcal{R}^S g''$. Suppose that $\neg\phi \in g''$ holds. By axiom 4., we get that $\Box\Box\phi \in g$. By definition, this means that $\Box\phi \in g'$ and that $\phi \in g''$.
3. Euclideanness: We have that if $\Box\phi \in g$, then $\phi \in g'$, and that if $\Box\phi \in g$, then $\phi \in g''$, then, if $\Box\phi \in g'$ then $\phi \in g''$. Suppose that $\phi \in g''$ is false. Then, we have that $\Box\phi \in g'$ and that $\neg\phi \in g''$. By axiom 5., we get that $\neg\Box\phi \to \Box\neg\Box\phi \in g''$. From which $\neg\Box\phi \in g'$ follows. This contradicts $\Box\phi ing'$.

References

Agentcities (2002). Open network of systems hosting agent-based services. http://www.agentcities.org.

AgentLink (2005). Agentlink. http://www.agentlink.org.

Agerri, R. and Alonso, E. (2002a). On meaning in ACL Pragmatics. In *UK Workshop in Multi-Agent Systems (UKMAS-02)*, Liverpool, UK.

Agerri, R. and Alonso, E. (2002b). Performatives and Agent Communication. In *Second International Workshop on History and Philosophy of Logic, Mathematics and Computation (HPLMC-02)*, pages 88–96.

Agerri, R. and Alonso, E. (2005a). Normative pragmatics for agent communication languages. In Akoka, J., Liddle, S. W., Song, I.-Y., Bertolotto, M., Comyn-Wattiau, I., vanden Heuvel, W.-J., Kolp, M., Trujillo, J., Kop, C., and Mayr, H. C., editors, *Perspectives in Conceptual Modelling, LNCS 3770*, pages 172–181, Heidelberg. Springer-Verlag.

Agerri, R. and Alonso, E. (2005b). Semantics and pragmatics for agent communication. In Bento, C., Cardoso, A., and Dias, G., editors, *Progress in Artificial Intelligence, (EPIA 05), LNAI 3808*, pages 524–536, Heidelberg. Springer-Verlag.

Agerri, R. and Alonso, E. (2006). A semantic and pragmatic framework for the specification of agent communication languages: Motivational attitudes and norms. In Kolp, M., Bresciani, P., Henderson-Sellers, B., and Winikoff, M., editors, *Agent-Oriented Information Systems III (Post-Proceedings)*, volume 3529, pages 16–31, Heidelberg. Springer-Verlag.

Akkermans, H., Gustavsson, R., and Ygge, F. (1998). Pragmatics of agent communication. In *KAW'98 Banff*, Canada.

Alchourron, C. E. and Bulygin, E. (1971). *Normative Systems*. Springer.

Allen, J. and Perrault, C. (1980). Analyzing intention in utterances. *Artificial Intelligence*, 15(3):143–178.

Alonso, E. (2004). Rights and argumentation in open multi-agent systems. *Artificial Intelligence Review*, 21:3–24.

Amgoud, L., Maudet, N., and Parsons, S. (2000). Modelling dialogues using argumentation. In Durfee, E., editor, *Proceedings of the 4th International Conference on Multi-Agent Systems (ICMAS-2000)*, pages 31–38. Boston (MA): IEEE Press.

Anderson, A. R. (1967). Some nasty problems in the formalization of ethics. *Nous*, 1:345–360.

Austin, J. L. (1962). *How to do Things with Words*. Oxford University Press, Oxford.

Bach, K. and Harnish, R. (1979). *Linguistic Communication and Speech Acts*. Cambridge (MA): The MIT Press.

Blackburn, P., de Rijke, M., and Venema, Y. (2001). *Modal Logic*. Cambridge University Press.

Boella, G., Damiano, R., Hulstijn, J., and van der Torre, L. (2006). Acl semantics between social commitments and mental attitudes. In *In proceedings of the AC 2006*.

Boella, G. and van der Torre, L. (2005). Role-based rights in artificial social systems. In *Proceedings of IAT05. IEEE, 2005*.

Boella, G., van der Torre, L., and Verhagen, H. (2005). Introduction to normative multi-agent systems. In *Proceedings of the AISB-05 Convention*.

Bradshaw, J., Dutfield, S., Benoit, P., and Woolley, J. (1997). KaOS: Toward an industrial-strength generic Agent architecture. In Bradshaw, J. M., editor, *Software Agents*, pages 375–418. AAAI Press / The MIT Press, Cambridge (MA).

Bratman, M. E. (1987). *Intentions, Plans, and Practical Reason*. Cambridge (MA): Cambridge University Press.

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

Castelfranchi, C. (1997). Practical permission: Dependence, power and social commitment. In *Proceedings of 2nd workshop on Practical Reasoning and Rationality*, London.

Chaib-Draa, B. and Dignum, F. (2002). Trends in agent comunication language. *Computational Intelligence*, 18(2):89–101.

Chellas, B. (1980). *Modal Logic: An Introduction*. Cambridge University Press.

Ciancarini, P. and Wooldridge, M., editors (2001). *Agent-oriented software engineering: first international workshop, AOSE 2000*, volume 1957 of Lecture Notes in Computer Science, Limerick: Ireland. New York: Springer-Verlag.

Clarke, E. M., Grumberg, O., and Peled, D. A. (1999). *Model Checking*. The MIT Press, Cambridge, MA.

Cohen, P. and Levesque, H. (1997). Communicative actions for artificial agents. In Bradshaw, J. M., editor, *Software Agents*. AAAI Press / The MIT Press, Cambridge (MA).

Cohen, P. R. and Levesque, H. (1985). Speech acts and rationality. In *23rd Annual Meeting of the Association for Computational Linguistics*. University of Chicago Press.

Cohen, P. R. and Levesque, H. (1990). Intention is Choice with Commitment. *Artificial Intelligence*, 42:213–261.

Cohen, P. R. and Levesque, H. J. (1991). Confirmations and joint action. In *Proceedings of 12th International Joint Confererence on Artificial Intelligence, Sidney, Australia*. Morgan Kaufmann Publishers, Inc., San Francisco, CA.

Cohen, P. R. and Perrault, C. R. (1979). Elements of a plan based theory of speech acts. *Cognitive Science*, 3:177–212.

Cohen, P. R., Perrault, C. R., and Allen, J. (1982). Beyond question answering. In Lehnert, W. and Ringle, M., editors, *Strategies for Natural Language Processing*. Lawrence Erlbaum Associates, Hillsdale, NJ.

Colombetti, M., Formara, N., and Verdicchio, M. (2002). The role of institution in multi-agent systems. In *Atti del VII convegno dell'Associazione italiana per l'intelligenza artificiale (AI*IA 02)*.

Cost, R. S., Chen, Y., Finin, T., Labrou, Y., and Peng, Y. (1999a). Modeling agent conversations with colored petri nets. In *Workshop on Specifying and Implementing Conversation Policies, Third International Conference on Autonomous Agents (Agents '99)*, Seattle.

Cost, R. S., Finin, T., Labrou, Y., Luan, X., Peng, Y., Soboroff, I., Mayfield, J., and Boughannam, A. (1999b). An Agent-based Infrastructure for Enterprise Integration. In *First International Symposium on Agent Systems and Applications (ASA'99)/Third International Symposium on Mobile Agents (MA'99)*, Palm Springs, CA, USA.

Dastani, M., Riemsdijk, M., Dignum, F., and Meyer, J.-J. (2004). A programming language for cognitive agents: Goal-directed 3apl. In Dastani, M., Dix, J., and Fallah-Seghrouchni, A. E., editors, *Programming Multi-Agent Systems (LNAI 3037)*, pages 111–130. Springer-Verlag, Berlin.

Dastani, M., van der Ham, J., and Dignum, F. (LNCS 2003). Communication for goal directed agents. In Huget, M.-P., editor, *Communication in Multiagent Systems - Agent Communication Languages and Conversation Policies*, pages 239–252. Springer-Verlag.

Dennett, D. C. (1981). *Brainstorms*. Harvester Press.

Dignum, F. and Kuiper, R. (1997). Combining dynamic deontic logic and temporal logic for the specification of deadlines. In Sprague, J. R., editor, *Proceedings of thirtieth HICSS*, Hawaii.

Endriss, U. (2005). Temporal logic for representing agent communication protocols. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2005)*. ACM Pres.

Endriss, U., Maudet, N., Sadri, F., and Toni, F. (2004). Logic-based communication protocols. In *Advances in Agent Communication, LNAI, volume 2922, pages 91-107, Springer-Verlag.*

Esteva, M., Rodriguez, J. A., Sierra, C., Garcia, P., and Arcos, J. L. (2001). On the formal specification of electronic institutions. In Dignum, F. and Sierra, C., editors, *Agent-mediated Electronic Commerce (The European AgentLink Perspective)*, volume volume 1191 of LNAI, pages 126–147. Springer, Berlin.

Fagin, R., Halpern, J. Y., Moses, Y., and Vardi, M. Y. (1995). *Reasoning about Knowledge*. The MIT Press, Cambridge, MA.

Ferber, J. (1999). *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Addison-Wesley.

Ferber, J. and Gutknecht, O. (1998). A meta-model for the analysis of organizations in multi-agent systems. In *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS'98)*, pages 128–135.

Fikes, R. and Nilsson, N. (1971). Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 5(2):189–208.

Finin, T., Fritzson, R., McKay, D., and McEntire, R. (1994). KQML as an Agent Communication Language. In Adam, N., Bhargava, B., and Yesha, Y., editors, *Proceedings of the 3rd International Conference on Information and Knowledge Management (CIKM'94)*, pages 456–463, Gaithersburg, MD, USA. ACM Press.

Finin, T., Labrou, Y., and Mayfield, J. (1997). KQML as an Agent Communication Language. In Bradshaw, J., editor, *Software agents*. MIT Press, Cambridge.

Finin, T. and Weber, J. (1993). Specification of the KQML Agent-Communication Language. Finin T. and Weber J. Draft. Specification of the

KQML Agent Communication Language. The DARPA Knowledge Sharing Initiative, External Interfaces Working Group.

FIPA ACL (2002). FIPA Communicative Act Library Specification. http://www.fipa.org/repository/aclspecs.html.

Flores, R. A. and Kremer, R. C. (2002). To commit or not to commit: Modelling agent conversations for action. *Computational Intelligence*, 18(2).

Fornara, N. and Colombetti, M. (2004). A commitment-based approach to agent communication. *Applied Artificial Intelligence an International Journal*, 18(9-10):853–866.

Fornara, N., Vigano, F., and Colombetti, M. (2004). Agent communication and institutional reality. In *AAMAS 2004 Workshop on Agent Communication (AC2004)*, New York.

Gazdar, G. and Mellish, C. (1989). *Natural Language Processing in PROLOG. An introduction to Computational Linguistics.* Addison-Wesley, Wokingham, England.

Genesereth, M. R. and Ketchpel, S. P. (1997). Software Agents. *Communications of the ACM*, 37(7):48–53.

Greaves, M., Holmback, H., and Bradshaw, J. (2000). What is a Conversation Policy? In Dignum, F. and Greaves, M., editors, *Issues in Agent Communication*, pages 118–131. Heidelberg, Germany: Springer-Verlag.

Grice, H. P. (1957). Meaning. *Philosophical Review*, 67:377–388.

Grice, H. P. (1969). Utterer's meaning and intentions. *Philosophical Review*, 78:147–177.

Grice, H. P. (1975). Logic and Conversation. In Cole, P. and Morgan, J., editors, *Syntax and Semantics vol. 3: Speech Acts*, pages 41–58. Academic Press, New York. Reprinted in Grice (1989) pages 21-40.

Grice, H. P. (1989). *Studies in the Way of Words.* The MIT Press, Cambridge, MA.

Guerin, F. and Pitt, J. (2002). Agent Communication Frameworks and Verification. In *AAMAS 2002 Workshop on Agent Communication Languages, Bologna.*

Habermas, J. (1984). *The Theory of Communicative Action.* Polity Press, Cambridge, UK.

Halpern, J. and Moses, Y. (1992). A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54:311–379.

Halpern, J., van der Meyden, R., and Vardi, M. (2004). Complete axiomatizations for reasoning about knowledge and time. *SIAM Journal of Computing*, 33(3):449–478.

Halpern, J. and Vardi, M. (1989). The complexity of reasoning about knowledge and time.i. lower bounds. *Journal of Computer and System Sciences*, 38(1):195–237.

Halpern, J. Y. and Vardi, M. Y. (1988). The complexity of reasoning about knowledge and time: synchronous systems. Technical Report RJ 6097, IBM.

Hintikka, J. (1962). *Knowledge and Belief.* Ithaca (NY): Cornell University Press.

Holmback, H., Greaves, M., and Bradshaw, J. (1999). Agent A, Can you pass the salt? - the role of Pragmatics in Agent Communication. http://citeseer.nj.nec.com/holmback98agent.html.

Huhns, M. N. and Singh, M. P., editors (1997). *Readings in Agents.* Morgan Kaufmann, San Francisco, CA, USA.

JADE (2002). Java agent development framework. http://sharon.cselt.it/projects/jade/.

Jones, A. J. I. and Sergot, M. (1993). On the characterisation of law and computer systems: The normative systems perspective. In Meyer, J.-J. and Wieringa, R., editors, *Deontic Logic in Computer Science: Normative System Specification.* Wiley.

Kagal, L., Finin, T., and Joshi, A. (2003). A policy language for a pervasive computing environment. In *IEEE 4th International Workshop on Policies for Distributed Systems and Networks.*

Kripke, S. (1963). Semantical considerations on modal logic. *Acta Philosophical Fennica*, XVI:83–94.

Kumar, S., Huber, M. J., Cohen, P. R., and McGee, D. R. (2002). Toward a formalism for conversation protocols using joint intention theory. *Computational Intelligence*, 18(2).

Labrou, Y. and Finin, T. (1994a). A semantic approach for KQML - a general purpose Communication Language for Software Agents. In *Third International Conference on Information and Knowledge Management (CIKM'94)*, pages 447–455.

Labrou, Y. and Finin, T. (1994b). A semantic approach for KQML - a general purpose communication language for software agents. In *Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94)*, pages 447–455.

Labrou, Y. and Finin, T. (1997). A Proposal for a new KQML Specification. Technical Report TR CS-97-03, University of Maryland, Baltimore County, USA.

Lee, M. (1998). *Belief, Rationality and Inference: A General Theory of Computational Pragmatics*. PhD thesis, Univ. of Sheffield.

Lomuscio, A. and Sergot, M. (2003). Deontic interpreted systems. *Studia Logica*, 75.

Lomuscio, A. and Wozna, B. (2006). A complete and decidable axiomatization for deontic interpreted systems. In *8th International Workshop on application of Deontic Logic to Computer Science (DEON-06)*, LNAI, Heidelberg. Springer-Verlag.

Manna, Z. and Pnueli, A. (1992). *The Temporal Logic of Reactive and Concurrent Systems: Specification*. Springer-Verlag, New York.

Manna, Z. and Pnueli, A. (1993). Models for reactivity. *Acta Informatica*, 30:609–678.

Manna, Z. and Pnueli, A. (1995). *The Temporal Logic of Reactive and Concurrent Systems*. Springer-Verlag, Berlin, Germany.

Mayfield, J., Labrou, Y., and Finin, T. (1996). Evaluation of KQML as an Agent Communication Language. In Wooldridge, M., Muller, J. P., and Tambe, M., editors, *Lecture notes in Artificial Intelligence*, pages 1–12. Berlin: Springer-Verlag.

McCarthy, J. (1990). Ascribing mental qualities to machines. In Lifschitz, V., editor, *Formalizing Common Sense: Paper by John McCarthy*, pages 93–118. Norwood (NJ): Albex Publishing.

Meyer, J.-J. and Wieringa, R., editors (1993). *Deontic logic in computer science: normative system specification*. Wiley.

Moore, R. (1980). Knowledge and action. Technical Report 191, SRI International, Menlo Park, California.

Norman, T. J., Sierra, C., and Jennings, N. R. (1998). Rights and commitment in multi-agent agreements. In *Proceedings of the Third International Conference on Multi-Agent Systems*, pages 222–229.

Nwana, H. (1996). Software agents: An overview. *Knowledge Engineering Review*, 11(3):205–244.

Phillips, L. R. and Link, H. E. (1999). The role of Conversation Policy in carrying out Agent Conversations. In *Workshop on Specifying and Implementing Conversation Policies, Autonomous Agents'99, Seattle.*

Pitt, J. and Mamdani, A. (1999). A protocol-based semantics for an agent communication language. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence IJCAI'99*, pages 486–491, Stockholm. Morgan-Kaufmann Publishers.

Raimondi, F. and Lomuscio, A. (2004). Automatic verification of deontic properties of multi-agent systems. In *Deontic Logic*, volume LNCS 3065, pages 228–242. Springer Verlag.

Rao, A. and Georgeff, M. (1991a). Modeling rational agents within a bdi-architecture. In Allen, J., Fikes, R., and Sandewall, E., editors, *2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, pages 473–484. Morgan Kaufmann Publishers.

Rao, A. and Georgeff, M. P. (1991b). Modeling agents within a bdi-architecture. In Fikes, R. and Sandewall, E., editors, *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, pages 473–484, San Mateo (CA). Morgan Kaufmann.

Rao, A. S. (1996). AgentSpeak(L): BDI agents speak out in a logical computable language. In van Hoe, R., editor, *Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, Eindhoven, The Netherlands.

Rao, A. S. and Georgeff, M. P. (1998). Decision procedures for bdi logics. *Journal of Logic and Computation*, 8(3):293–342.

Rodriguez-Aguilar, J., Noriega, P., Sierra, C., and Padget, J. (1997). Fm96.5 java-based electronic auction house. In *2nd Int. Conf. on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'97)*, London.

Rovatsos, M., Nickles, M., and Weiss, G. (2003). Interaction is meaning: A new model for communication in open systems. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-03)*, Melbourne, Australia.

Sacks, H. (1972). *Lectures on conversation.* Blackwell, Oxford. Edited by G. Jefferson. Published in 1995 with an introduction by E. A. Schegloff.

Schild, K. (2000). On the relationship between bdi-logics and standard logics of concurrency. *Autonomous Agents and multi-agent systems*, 3:259–283.

Searle, J. R. (1969). *Speech Acts. An Essay in the Philosophy of Language.* Cambridge: Cambridge University Press.

Searle, J. R. (1979). *Expression and Meaning. Studies in the Theory of Speech Acts.* Cambridge: Cambridge University Press.

Searle, J. R. (1983). *Intentionality: An Essay in the Philosophy of Mind.* New York: Cambridge University Press.

Searle, J. R. (1995). *The Construction of Social Reality.* The Free Press, New York.

Searle, J. R. and Vanderveken, D. (1985). *Foundations of Illocutionary Logic.* Cambridge University Press, Cambridge, UK.

Sierra, C., Jennings, N. R., Noriega, P., and Parsons, S. (1998). A framework for argumentation-based negotiation. In Singh, M. P., Rao, A., and Wooldridge, M. J., editors, *Fourth International Workshop on Agent Theories Architectures and Languages (ATAL-97)*, volume 1365 of *Lecture Notes in Computer Science vol. 1365*, pages 177–192. Springer-Verlag.

Singh, M. P. (2000). A social semantics for agent communication languages. In *Issues in Agent Communication, volume 1916 of LNAI.* Berlin: Springer-Verlag.

Stalnaker, R. (1989). On Grandy on Grice. *Journal of Philosophy*, 86:526–527.

SWI-Prolog (2005). University of amsterdam. http://www.swi-prolog.org.

Tuomela, R. and Miller, K. (1988). We–intentions. *Philosophical Studies*, 53:367–389.

van der Hoek, W. and Wooldridge, M. (2003). Towards a logic of rational agency. *Logic Journal of the IGPL*, 11(2):135–159.

van der Meyde, R. and Wong, K. (2003). Complete axiomatizations for reasoning about knowledge and branching time. *Studia Logica*, 75:93–123.

van der Torre, L. (2003). Contextual deontic logic: Normative agents, violations and independence. *Ann. Math. Artif. Intell.*, 37(1-2):33–63.

van der Torre, L., Hulstijn, J., Dastani, M., and Broersen, J. (2004). Specifying multiagent organizations. In *Proceedings of the Seventh Workshop on Deontic Logic in Computer Science (Deon'2004), LNAI 3065*, pages 243–257. Springer.

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

Wenar, L. (Winter 2005). Rights. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy.* http://plato.stanford.edu/archives/win2005/entries/rights/.

Winograd, T. and Flores, F. (1987). *Understanding Computers and Cognition: A New Foundation for Design.* Addison-Wesley, Reading (MA).

Wooldridge, M. (1999). Verifying that agents implement a communication language. In *Sixteenth National Conference on Artificial Intelligence (AAAI-00)*, Orlando, FL.

Wooldridge, M. (2000a). Computationally grounded theories of agency. In Durfee, E., editor, *Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS 2000)*, IEEE Press.

Wooldridge, M. (2000b). Semantic issues in the verification of agent communication languages. *Journal of Autonomous Agents and Multi-Agent Systems*, 3(1):9–31.

Wooldridge, M. (2002). *An Introduction to MultiAgent Systems*. John Wiley and Sons.