Chapter 1

Introduction

In the future many services currently performed by humans will be performed by software programs on the Internet. These programs are what we call *agents*. It is conceivable that travel agencies, video rental stores and banking could all be implemented with greater efficiency and at a lower cost by agents. There will also be many new services on offer, for example a *personal* agent (owned by a user) could be continuously roaming the Internet seeking out information and services of interest to the user. We can envisage a large community of agents roaming the Internet, each with their own goals, acting on behalf of their owners. Many commercial companies will contribute to the development of this community so the design of all agents will not be identical. However, they will have to communicate to negotiate deals, find information and generally cooperate. Such a group of agents will need a commonly understood *language*: a *lingua franca* for agents. The actions agents take on behalf of their owners have effects in the real world; they may engage in financial transactions or enter legally binding contracts. Agent owners will require that certain rules are in force so that guarantees can be made about the contracts an agent can enter; this also is a concern of the communication language. This thesis describes how such languages can be specified.

1.1 Agents in Artificial Intelligence

Within the field of Artificial Intelligence (AI) is the field of Distributed Artificial Intelligence (DAI) which is itself composed of (Nwana, 1996):

- Distributed Problem Solving (DPS)
- Parallel Artificial Intelligence
- Multi-Agent Systems (MAS)

In the case of the first two, distributed entities are necessarily cooperating to solve some problem (Ferber, 1989). In multi-agent systems individual agents may be competing (Sandholm, 1996) or cooperating (Ferber and Drogoul, 1992).

We use the term *agents* to refer to software programs which have some degree of *autonomy* and which can be *delegated* to perform certain tasks. Ideally agents should be sufficiently intelligent to be able to anticipate, adapt and actively seek ways to support human users (Bradshaw et al., 1997). The notion generally carries connotations of some anthropomorphic entity, a human-like helper who can go about tasks without needing constant directions from the human user. Continuing with the anthropomorphic analogy, it is common to adopt the intentional stance both in the analysis and design of agents. This means attributing mental attitudes including *belief*, *desire* and *intention* to the agents. Many agent architectures feature explicit representations of these attitudes (Rao and Georgeff, 1992). It is not necessary for an agent to have human level intelligence to justify the attribution of human level mental abstractions to it (McCarthy, 1990). A system of less than human intelligence may be sufficiently complex that the attribution of intentions and desires to it is the easiest way to view it.

The difference between the notion of an agent and the more general notion of a software program is the higher level of abstraction at which agents are viewed. Agents differ from Object Oriented programming, AI, and distributed computing because they can be delegated high level tasks and will carry them out autonomously (Wooldridge and Jennings, 1996). The agent programming paradigm is at a higher level than object oriented programming as code is encapsulated in agents. One of the goals of multi-agent systems research is to facilitate the delegation of human level tasks to agents and to have human level interaction among agents so that agents could be seen as our electronic counterparts.

Social interactions are central to the idea of multi-agent systems. In most systems no single agent has all the knowledge or skills to complete its task, so it will require cooperation from other agents. This in turn requires a commonly understood communication language. Since agents are high level objects we expect them to communicate at a high level, with a language sufficiently expressive to capture human level mental attitudes. We need an artificial language appropriate for artificially intelligent entities. It is also relevant to talk about issues of sincerity and trust in agent communication since the agents may be competing to achieve their individual goals. Social structures may also be needed: agents may occupy certain roles in the society, with associated relationships of power and obligation; agents may need to represent these social structures (d'Inverno et al., 1997) in order to record the social relations that hold between them.

Multi-agent systems is an interdisciplinary field of research which encompasses research on individual agents as well as research on societies and group behaviour; a comprehensive account is given by Ferber (1995). Research on individual agents draws on many fields including knowledge representation and planning, neural

adaptivity	Changes its behaviour based on experiences.
anthropomorphism	Behaves like a human. A society of agents can similarly be thought of as being akin to a human society.
autonomy	Operates on its own, without human guidance.
character	Believable personality and possibly emotional state (relevant in human computer interaction).
continuity	Continuously running process rather than a one off computation.
cooperativity	Cooperates with other agents to share resources, resolve conflicts or collectively solve problems.
interactivity	Communicates with humans and/or other agents.
mobility	Can transport itself to another site in a network.
proactivity	Takes initiatives to satisfy goals.
rationality	Works out (intelligently) how to achieve goals.
reactivity	Responds to changes in environment.
reflectivity	Exhibits self awareness by introspection of its own internal state.

Table 1.1: Some Agent Characteristics (Mamdani, 1997).

networks, fuzzy logic and genetic algorithms. New issues arise when these technologies are applied to agents. For example, the problem of planning for an agent whose architecture is based on beliefs, desires and intentions when the agent is situated in a dynamic environment where strategies may need to be altered. The anthropomorphic nature of agents means that psychology and cognitive science are also relevant. Research on agent societies and group behaviour includes research on social relations between agents, notions of power, trust, norms and institutions (Singh, 1999; Artikis et al., 2001). These considerations go hand in hand with an agent communication language since it is the messages exchanged that will modify social relations according to the conventions of a society. Linguistics, sociology and legal systems have analysed aspects of human social behaviour and these can provide valuable insights for agent societies.

Agents are an attractive programming metaphor because they allow us to deal with high level tasks and human level mental abstractions such as beliefs, desires and intentions as well as social abstractions like commitment. In order to mathematically prove properties of such systems we must be aware of the underlying computational processes. This thesis makes a contribution towards bridging the gap between the high level abstractions provided by agents and the low level computational processes which implement them.

1.2 Agent Communication

In a typical multi-Agent system the knowledge and functionality of the system is distributed among the constituent agents. Agents must have methods for sharing knowledge and taking advantage of each others capabilities as needed. It would not be feasible for every agent to have complete knowledge of the entire system, nor to have all possible capabilities. In order to cooperate effectively agents need to have standardised methods for exploiting each others resources. When an agent desires to achieve a goal which it cannot satisfy alone, it needs to be able to find an agent (or agents) which may be able to help and a method by which it can expect to get that help. That is, it must communicate its needs using a standardised language and the recipient must be able to understand the request, and respond appropriately. The language used needs to be sufficiently expressive to allow agents to transmit complex information and goals, possibly programming each other (Genesereth and Ketchpel, 1994). Some agent communication languages have already been developed but none has been widely adopted as a standard.

One of the key questions in agent communication is how to describe the *meaning* of a communication: in other words what *semantic definition* do we use. Consider, for example, a simple commercial transaction; one agent sends a *request(price of x)* message to another agent, what does this mean?[1] Some would say it means that the sender *desires* to *know* the price of *x*. Some would say it means that the receiver must reply with either a "*don't know*" or "*the price of x is 4*". It is the contention of this thesis that the meaning can be best described as *an expressed desire of the sender to know the price of x*. These three approaches to semantic definition are described in the following paragraphs.

Mental Approach: The first wave of research in agent communication used the mental attitudes of the agents as a basis for describing the semantics of communication. This approach arose from work whose original aim was to develop a theory of human intention (Cohen and Levesque, 1990); it was later applied to agent communication (Cohen and Levesque, 1995). There are other closely related approaches (Labrou and Finin, 1994). Essentially the semantics of a communication is described in terms of the beliefs, desires and intentions of the communicating agents. As a simple example, if a speaker makes a *request* for directions, this approach would describe the meaning as the speaker's *desire to know* the direction. Semantics of these mental attitudes is given in terms of the possible states of the world (modal logic), given the agent's current mental state. The theory assumes that agents can deduce the implications of all their beliefs; in practise this may not be possible because of computational limitations (Wooldridge, 1992). These semantics are also not grounded in a computational model; i.e. it is not clear how the modal logic relates to the computational processes implementing the agents. A

[1] We are only concerned with the *wrapper* language here: the *request* part; we do not deal with the *content* language.

further difficulty relates to the assumed sincerity of participants; i.e. the semantics does not account for a situation where an agent might lie and not really hold the mental attitude described by the semantics.

Behavioural Approach: The second approach defined semantics behaviourally: an ordered sequence of message sending was described by means of a finite state diagram (Akkermans et al., 1998) also called a conversation policy (Greaves et al., 2001). The meaning of a communication was defined behaviourally in the context of a conversation in that only certain responses were deemed appropriate. This was attractive from an implementation point of view since it was easy to ensure that agents followed the prescribed sequences of messages; i.e. there was a very direct relationship between the theory and the implemented system. The main criticism of this approach is that it is too low level, agents should not be treated as simple communicating objects following a predefined behaviour diagram. Such predefined behaviours would constrain the autonomy of the agent excessively. As a human, if we are asked for directions to the nearest shop, we infer the desire of the speaker. We would assume that the speaker desires to purchase something there, and therefore we would not be likely to direct them to a shop which we know to be shut. This kind of inference is only possible if we understand the meaning of the request at a high level. If a request means nothing more than the appropriate responses (we must respond with the answer or a refusal) then higher level reasoning is not possible.

Social Approach: More recently a third approach has taken a social perspective on communication (Singh, 2000) and recognises the importance of context. The semantics of a communication can be described in terms of the change it causes in the social relations existing between the conversational participants. For example it may create or modify commitments. This avoids the problems of previous approaches and is more in line with theories of human communication which recognise that communicative actions have their origins in social practices (Clark, 1996, p. 139). The approach of this thesis takes a social perspective and also introduces the notion of *expressed mental attitudes* to describe what is conventionally expressed by a communication without assuming sincerity of the speaker. A speaker may express an attitude without necessarily having that attitude internally. These social meanings are at a high level, such that the *expressed desire* of the speaker in the *shop-query* example above could be recognised.

1.3 The Open Agent Society

An open agent society is one where the constituent agents may be developed and owned by different individuals or organisations who may have conflicting interests. Hence the internals (i.e. program and state) of agents are not public and so notions of trust and deception are relevant. It is proposed that such systems will be used in scenarios where legally binding contracts (with effects in the real world) are made.

Agents will also engage in financial transactions on behalf of their owners; for example, they may be required to pay for the services of other agents (Genesereth and Ketchpel, 1994). Therefore there must be an explicit delegation of responsibility from a human or an organisation responsible legally for the agent's actions (Mamdani and Pitt, 2000). Not surprisingly, agent owners can be expected to be reluctant to delegate tasks involving potentially detrimental outcomes to an agent unless they can be assured that the system has certain desirable properties. For example, does the system guarantee that my agent will not be discriminated against in favour of a competing agent? Or that my competitor cannot benefit by lying to my agent? Or that I get the optimal price? It may be impossible to guarantee the most desirable outcomes for all participants, but the system should be at least as good as the best non-agent alternative. The best alternatives are real market mechanisms. Self interested rational agents in an open society can be treated in much the same way as humans playing games or participating in markets. Solutions from game theory and economics (Binmore, 1992) allow us to design *mechanisms for interactions* which have the properties we desire.

A mechanism for an interaction is a set of rules by which the interaction will be conducted. A good example is given by Rosenschein and Zlotkin (1994) in relation to airport landing charges. Airplanes are coming to land at an airport and must enter a queue awaiting a free runway. The airplanes report how much fuel they have remaining to the control tower and planes with less fuel are promoted in the queue. It is in the interest of planes to land early as they will expend less fuel circling in the queue and their passengers will not be delayed. Thus it is in the interest of the pilot of a plane to under-report the amount of fuel remaining. To discourage this practice the ground crew will check the fuel level when the plane lands to determine if the pilot was truthful. However, this is difficult to determine since the plane will have used some fuel after the last report and in landing. The amount used is not easily determined and may be weather dependent. Thus it becomes a game between airline and airport to under-report just enough not to be caught out. Game theorists would propose the design of a mechanism in this scenario such that all airplanes would pay a tax on landing and that tax would be greater for planes landing earlier. In the case of planes landing very late the airport might be the one to pay the airline. The tax would be set at a rate just high enough to cancel the benefit of landing early. Thus it would not be in the interest of the airlines to under-report their fuel. Such a mechanism would be called *incentive compatible* since the players have an incentive to tell the truth. An additional benefit of such a mechanism is that players will not waste valuable resources with airlines trying to determine the maximum level of under-reporting they can get away with and airports trying to determine who is under-reporting.

In agent systems we usually call this type of mechanism a *protocol*. It is a set of "public rules by which agents will come to agreements" (Rosenschein and Zlotkin, 1994). When participating in a protocol, individual agents will use private *strategies*. The protocol can be designed to influence the optimal strategy of the partici-

pating agents; for example if the protocol has the property of *incentive compatibility* then it will be optimal for the agent strategy to take the actions that the protocol designer is intending to induce. In the airplane landing scenario the mechanism was designed to induce truth telling. Many mechanisms for interactions have been researched in economics, for example designing auctions to induce an optimal outcome for the seller (Vickrey, 1961; Myerson, 1981). A lot of work has also been done on applying this work to problems in agent systems. Sandholm (1996) has done work on the high level protocols required for self-interested computationally limited agents to negotiate. If we have designed a protocol for agent interactions, how should we write it down? Could we write it down in some language that an agent could read? In the human world, a person who has never been at an auction can read the rules in a book and can then go and participate. Any observer at the auction can determine if any participant in the auction is not following the rules. How can we do this in the agent world?

To specify a protocol for agents we need a formal language; natural language is too ambiguous. Given that we develop such a formal language, how can we guarantee that a system of agents using a protocol does indeed have the desired properties? Such guarantees are important to agent owners who may delegate a task to an agent and also to agent designers who will design the agents' strategies. For example, if truth telling is proved to be optimal, then the agent designer need not consider deceptive strategies. The mechanism must be specified publicly in the form of a protocol with a procedure for carrying out mathematical proofs so that all participants can verify its properties for themselves. There must also be some means of enforcement so that rogue agents will not be permitted to damage the system's properties. This requires a method for determining if agents are not following the protocol rules.

At a higher level than the individual protocols in a system, we may need certain global rules possibly enforced by institutions which would ensure that the agent society does not become dysfunctional. These rules could be analogous to the social conventions mandating politeness and helpfulness in the human world. These concerns are inherently tied to a communication language, as with human languages where many of the conventions of society are present in the language.

> "One can think of the complicated system of laws and conventions as a kind of social engineering, intended to produce certain behaviour among people. We are interested in social engineering for machines." (Rosenschein and Zlotkin, 1994)

1.4 Mathematical Tools

In order to formally analyse and reason about multi-agent systems we need a precise mathematical description of every component of the system including the com-

munication language. This thesis avoids the types of ungrounded logics used in the first agent communication languages (described in section 1.2); instead it uses well established theories of computation to develop a grounded framework within which higher level aspects (such as mental attitudes expressed by agents) can be specified.

Agents on the internet will engage in many different e-commerce applications and different applications will need different languages and protocols. This is analogous to language use in the human world where conventions are different in courthouses, auction houses or normal conversation. Therefore this thesis focuses on developing a framework rather than a single language. The framework provides a specification language within which an agent communication language can be written. The specification language provides a well defined meaning for a communication language. In much the same way as a compiler converts a high level program into executable code, the specification language converts the communication language into a function which an agent can use at run time to interpret messages (see figure 1.1). The mathematical description of this converter is given by denotational semantics.

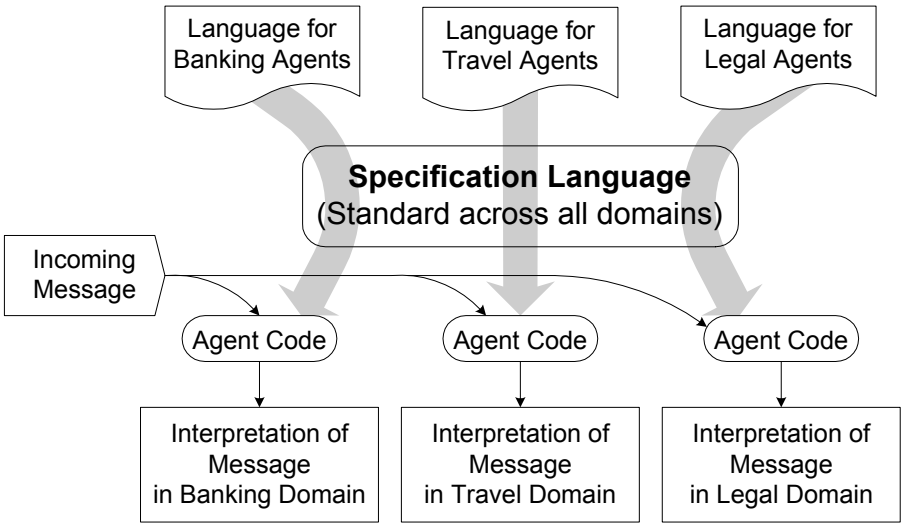

Figure 1.1: Application Specific Languages Using a Standard Specification Language.

To mathematically prove that certain properties hold for agent systems we need to examine the low level details of their implementation. A multi-Agent system may be implemented by computers running several agent programs in parallel. Mathematical proofs of properties of systems of concurrently executing computer programs are provided by computing theories. We formally treat an agent program as a reactive program.

> "A reactive system is a system that maintains an ongoing interaction with its environment, as opposed to computing some value on termination." (Manna and Pnueli, 1995)

Verifying compliance for reactive programs is done by temporal verification, using a linear temporal logic to specify a desired property of the program.

These theoretical tools will allow us to prove that properties hold for a system of agents.

1. As an agent designer we can verify that an agent's code complies with the rules specified by the language.
2. As an agent system administrator we can observe a history of transactions in our system and determine if any agent is not following the public rules specified by the language.
3. As a protocol designer we can determine the set of possible outcomes for a system of agents which complies with the rules of our protocol. If all the outcomes in the set satisfy a certain property then we have proved that the protocol has that property. Properties to be proved can be those developed for mechanisms in economics, for example incentive compatibility as described in the previous section.
4. An agent designer can design a private strategy for an agent participating in a protocol. It will then be possible to determine the set of possible outcomes for that agent provided that it is operating in a system of compliant agents. Finally properties can be proved for this set of outcomes; for example that all outcomes are optimal.

1.5 Contribution of this Thesis

The contribution of this thesis is in the area of formal tools to specify, analyse and reason about multi-agent systems with particular emphasis on agent communication languages (ACLs). In detail the contributions are as follows:

- A critical analysis of existing ACLs (with particular attention to application in open systems) which allows a set of requirements to be drawn up for a language for use in open systems.
- A computational model for multi-agent systems which can represent the observable states of the system.
- A rich description of social states using *expressed mental attitudes*, commitments, role relationships and state variables. The states are grounded in

the computational model and hence the framework shows that agent communication can be given a high level semantics which is verifiable if social phenomena are used.

- A protocol specification method which uses information in the social state to control the flow of a conversation. This is more efficient than the more traditional finite state description of a protocol where each unique state must be enumerated.

- A general framework within which agent communication languages can be specified and several different notions of verification can be defined. This leads on to an analysis of what types of verification are possible given limited information, with particular attention to an open system where agent internals may not be accessible.

- A more specialised framework for specifying verifiable languages in an open system. The framework is based on a specification language for ACLs which allows semantics to be defined as a change in the observable state of the system. Since the specification language allows for different ACLs to be used in different applications it could be a useful basis for a standard. That is, the specification language would be standardised and any ACL could be *plugged in*.

- A language for specifying the social conventions for a system, i.e. the changes induced in social relations by communication. A further language is provided for specifying the semantics of social relations. This allows agents to communicate commitments and gives them social awareness. This could be used for the formal specification of norms and institutions, though this thesis does not investigate these topics, it provides the low level formalism which could facilitate their specification.

- A method for publicly specifying the inference rules which a socially aware agent is expected to use; these are specified in a procedural fashion which is easily implemented. This simple mechanistic approach avoids the problem of logical omniscience associated with more traditional approaches.

- The application of a model checking algorithm to verify compliance with the ACL and to prove properties of protocols.

- A development method for ACLs which is demonstrated by specifying an ACL which includes semantics for all the major categories of communicative acts and some common protocols.

- A demonstration of the application of game theoretic concepts to the specification and analysis of protocols for multi-agent systems.

- The formal framework provides a step towards bridging the gap between the high level abstractions provided by agents and the low level computational processes which implement them. In particular, it provides a specification for the low level implementations.

This thesis is not trying to advance game theory, social engineering, philosophy of language, program semantics or verification of programs. Instead its purpose is to show how all these can be used to solve the problem of specifying languages for agent systems. This thesis is one contribution towards the vision of an open agent society described at the beginning of this chapter.

1.6 Some Open Questions

The thesis focuses on the external specifications for an agent system; these are the public rules governing communication. It does not devote much attention to discussing how agents might be built to operate in such systems (although some example agents are given). The private inferences an agent makes upon receiving a communication and the agent's strategy for selecting new messages to send are left open by the proposed languages. It is easy to imagine a human agent designer being able to inspect a published language and then to design an agent to use the language. However it is not so easy to imagine how an agent itself might inspect a specification for a language and come up with a strategy for using the language. This is the "holy grail" of interoperability for open systems of heterogeneous agents: sharing protocols and languages and knowing how to use them without any intervention on the part of a human. Given that we have a high level semantics for communicative acts, it should be possible for agents to formulate their own strategies for use with new protocols or languages. To do this we need to make our agents understand something of the human level of the meaning of communication. We run into some of the real problems in artificial intelligence here; more on this can be found in chapter 8. Some of the open questions include how best to specify the meaning of individual messages, how to represent an agent's internal state and how to specify social relations (norms and institutions for example).

1.7 Thesis Outline

Here is a brief overview of the material presented in each of the following chapters:

Chapter 2 is a literature review beginning with research on human communication including conversation analysis (Sacks, 1972) and speech acts (Searle, 1965). The insights gained from human communication are the main inspiration for the framework developed in the next chapter; in particular we try to emulate the rich context

which is built up as a human conversation progresses and is exploited by the participants to convey their meaning efficiently. We also bear in mind the social nature of communication and the fact that it is based on a system of conventions. The remainder reviews research on agent communication languages including languages based on mental attitudes, behaviour based languages and languages taking a social perspective. The chapter evaluates various ACLs and their various approaches to semantic definition, coming up with desiderata for a good definition. The most promising approach found is that of Singh (2000).

Chapter 3 develops a general agent communication framework within which various different notions of verification can be discussed. The framework is used to see how verification is possible with various different languages. The chapter then identifies a more specialised framework for use in open systems where agent internals are not accessible. It is this specialised framework which is used in the remainder of the thesis. The framework features an explicit representation of social context and uses a multi-variable state representation for protocols.

Chapter 4 presents the four languages on which the framework relies. The main body of the chapter is devoted to the specification language for ACLs. Much of the material in this section is a revision of material published in Guerin and Pitt (2001). A familiarity with denotational semantics is beneficial here, a brief introduction is provided in the chapter, but a more comprehensive introduction can be found in Schmidt's book (Schmidt, 1986, chapters 1 through 5). The other languages include a language for social relations, a language for agent programs and a language of temporal logic. These last two languages being summarised from Manna and Pnueli (1995).

Chapter 5 demonstrates the verification methods outlined in chapter 3 by proving some properties for a simple system of communicating agents. A model checking verification algorithm is used. This makes use of the theory of temporal verification for reactive systems (Manna and Pnueli, 1995), some of which is summarised in the chapter, a more comprehensive account can be found in Manna and Pnueli's second book (Manna and Pnueli, 1995, chapter 0).

Chapter 6 presents an ACL specified within the framework of the thesis. The ACL includes semantics for all the major categories of speech acts and some common protocols. This is not an attempt to design definitive ACL, it is the contention of this thesis that such an undertaking is not feasible as different applications will need different ACLs. This chapter shows how ACLs can be specified in a modular fashion so that the specification of useful, flexible communication primitives simplifies the task of protocol design. It also shows how social facts can be used in protocol design to describe the protocol state.

Chapter 7 discusses the application of game theoretic results to protocol design and gives an example of how useful properties can be specified and proved for a system of ACL compliant agents following a protocol. In an open agent system we cannot force the agents to adopt a particular architecture or to make their strategies

public, but we can design protocols so that it is in the agents' own interest to behave as we desire.

Chapter 8 concludes and outlines areas of future work. It includes a discussion of how agents might be built to understand new languages without the intervention of a human user. We also describe tools which would be useful for the development and analysis of agent systems.