

# Chapter 1

## Introduction

This thesis contributes to the field of multi-agent systems research. In this field, software systems are analysed and designed as if they are societies of autonomous, rational actors or *agents*. Section 1.1 defines multi-agent systems and discusses the notion of an agent. Section 1.2 presents some potential benefits of multi-agent systems. Section 1.3 discusses a number of issues that are encountered in the current state of the art. Section 1.4 describes the specific research aim of this thesis. Section 1.5 presents an outline of this thesis.

### ***1.1 Multi-Agent Systems***

A multi-agent system is defined as a system consisting of a number of agents that share a common environment. The definition of an agent is more involved. During the past years, quite a few researchers have attempted to define the notion of an agent (see (Franklin & Graesser, 1997) for an overview). The most influential definition is probably the ‘weak notion of agency’ defined by Wooldridge and Jennings (1995b). This notion is presented in Section 1.1.1 below. Wooldridge and Jennings have also defined, in the same paper, a stronger notion of agency, which is presented in Section 1.1.2. A perspective on the various notions of agency is presented in Section 1.1.3.

#### ***1.1.1 A Weak Notion of Agency***

In their influential paper, Wooldridge and Jennings (1995b) propose two notions of agency, called the weak notion and the strong notion of agency. The weak notion of agency defines an agent as a hardware- or software-based computer system that is *autonomous*, *reactive*, *pro-active* and has *social ability*. The most important property is autonomy: an agent is able to set its own goals and to choose a way to achieve those goals. Autonomy is only of interest for *situated* agents: agents that are situated in an environment, possibly shared with other agents, and that are able to observe their environment and carry out actions in the environment. (Wooldridge

## 1.1: Multi-Agent Systems

recently<sup>1</sup> refined the weak notion of agency by defining an agent as an *autonomous, embodied* hardware or software system, with *reactivity, pro-activeness* and *social ability* as additional properties.) The properties reactivity, pro-activeness and social ability together determine the *flexibility* of an agent:

- Reactivity of an agent refers to its ability to react, presumably in a sensible way, to unexpected situations arising in its environment;
- An agent is, by its autonomy, not only able to set its own goals, its pro-activeness ensures that it will actually do so. Thus, an agent *takes the initiative* and creates opportunities to pursue its goals instead of merely reacting to its environment;
- An agent's social ability enables it to co-operate with other agents in its environment. Co-operation is almost always needed to achieve the goals an agent has set.

The weak notion of agency of Wooldridge and Jennings raises a number of interesting issues:

- As the Latin stem of the word 'agent' (*agere*, to do) indicates, it is essential that an agent is an *active* software or hardware system. Dictionary definitions of an agent often put most emphasis on this property. (E.g., an agent is 'someone or something that acts'. Emphasis is also put on the social ability of an agent by indicating that an agent is 'someone or something that acts on behalf of someone else').
- The adjective 'active' might appear to be superfluous, especially for software systems. This adjective, however, emphasises that agents are *not* merely data structures. Objects in object-oriented development are often seen as active data structures: operations on data are encapsulated with the data itself. However, not every object is an agent. The distinction between objects and agents is best formulated by Wooldridge (1999) as follows. An *object* has no control over the execution of the operations it encapsulates (usually called its *methods* in object-oriented development). E.g., if object 1 invokes one of object 2's operations, this operation is executed regardless of the state of object 2. An *agent*, by definition (an agent is autonomous), has complete control over its own actions. Other agents may transfer requests to perform specific actions to an agent, comparable to operation invocation in object-oriented systems. The agent itself, however, decides whether the request will be granted. A rational agent bases this decision on an evaluation of the relation between the requests and its own goals, commitments and agenda.
- Wooldridge and Jennings do *not* require an agent to be mobile. Especially in the popular press, agents are depicted as software components that roam the

---

<sup>1</sup> In a lecture delivered at the SIKS Annual Meeting in Amsterdam, November 1, 1999.

Internet, collecting information for their dispatchers on the go. According to this concept of agency, mobility is essential. However, the notion of autonomous, embodied (software) components, or agents for short, can be applied to non-mobile systems as well, and this is also beneficial. Moreover, the primary focus of much research in the field of mobile agents is on implementation aspects of mobile agents, such as transportation, security, operating system support, and APIs (Application Programmer's Interfaces). Although the properties of agency distinguished by Wooldridge and Jennings are often mentioned (especially autonomy), the question how, on the one hand, mobility, and, on the other hand, autonomy, reactivity, pro-activeness and social ability interact is not addressed. Such research (Gray, Cybenko, Kotz & Rus, 1997; Johansen, Renesse & Schneider, 1997) is closer to research in the fields of object orientation or component based development than to research in multi-agent systems.

- It goes without saying that the weak notion of agency applies to human beings to a large extent: humans are generally assumed to be autonomous, situated in their society and socially able. This is of interest for agent-based approaches to requirements engineering: the same concepts can be used both for the analysis of (existing) human procedures and the analysis of automated systems consisting of hardware or software agents.

### 1.1.2 A Strong Notion of Agency

Wooldridge and Jennings (1995b) also define a stronger, more specific notion of agency. Literally, this stronger notion is defined as follows:

“... a computer system, that, in addition to having the properties identified above [autonomy, reactivity, pro-activeness and social ability, PvE], is either conceptualised or implemented using concepts that are more usually applied to humans. For example, it is quite common in AI to characterise an agent using *mentalist* notions, such as knowledge, belief, intention, and obligation (Shoham, 1993).”

Under the strong notion of agency, an agent is ascribed a mental state, consisting of e.g. belief, knowledge, intentions, goals, commitments and obligations. The behaviour of the agent is described using these mentalistic notions.

According to McCarthy (1987), the use of notions more usually applied to humans is both legitimate and often useful. It is legitimate if these notions express the same information for both humans and agents. Seel (1989) argues that this is possible for all kinds of automata. Using notions normally used for humans is useful in the sense that it provides a high-level description of the behaviour of an agent, instead of a description in terms of the actual mechanisms with which its behaviour is generated. Often, a description in terms of mentalistic notion is the only description available, or the only description simple enough to be

## *1.2: The Promise of Multi-Agent Systems*

comprehended. As an aside, the strong notion of agency only plays a minor role in this thesis.

### *1.1.3 Other Notions of Agency*

The notions of agency developed by Wooldridge and Jennings are not without criticism. In fact, almost every introductory text on multi-agent systems proposes a new definition for the notion of an agent. Franklin and Graesser (1997) survey ten alternative definitions found in the agent literature and then propose their own definition:

“An autonomous agent is a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future.”

Franklin and Graesser thus emphasise situatedness and continuity of an agent, and require that there is some kind of feedback: the actions of an agent should affect future observations of the agent. Both continuity as well as the feedback requirement help to distinguish agents from, in Franklin’s and Graesser’s terms, ‘ordinary programs’.

The lack of one single accepted definition of the concept of an agent is a source of much criticism from other research areas. It is not uncommon that central notions in a newly established research field are subject to debate within the field for a long time. Often, valuable research results are obtained as a side effect of these debates. However, such debates should not hide the central focus of the field of multi-agent systems: the creation of software systems that are designed for flexibility. An agent is autonomous and is able to devise its own goals and pursue them in a rational way, all by itself. Unexpected events in its environment should not hinder the agent in the completion of its tasks. The central issue in the field of multi-agent systems research is to answer the question how software with this level of flexibility can be engineered.

## *1.2 The Promise of Multi-Agent Systems*

In the field of multi-agent systems research, software systems are analysed and designed as if they are societies of autonomous agents. The benefits of this focus are twofold.

First, autonomous agents are a natural metaphor for today’s complex, networked software systems. In such systems, there is no central authority to which the software components are enslaved. In other words, such components are autonomous. Network connections between the components enable them to cooperate and to act upon other components, but the precise behaviour of other components is beyond their control. The subject of multi-agent systems research precisely matches this context: autonomous agents that are situated in an environment in which they may influence, and are influenced themselves, by other

agents. This metaphor naturally leads to decentralised systems, which, together with the flexibility of an agent—its reactivity, pro-activeness and social ability—is needed for robustness: the possibilities to cope with unexpected events in the environment. It is possible to design decentralised, robust systems using conventional approaches. However, in conventional approaches, there is no inherent support for agent-like properties. In agent-oriented approaches, such support is the norm.

Second, analysing and designing software systems as if they are societies of autonomous agents paves the way for the application of theories from the Social Sciences to software systems. A multi-agent system is a system that consists of a number of agents that share a common environment. Just like humans and their society, agents both have the potential to mutually benefit from co-operation and are constrained by interdependence. The Social Sciences (e.g., Economics, Management Science, Sociology, Political Science, and Social Psychology) investigate how social functions (i.e., specific behaviour of a society as a whole, such as, for instance, the occurrence of a pareto-optimal market equilibrium or a traffic jam) relate to the individual behaviour of (human) agents. Results obtained in the Social Sciences can be used to support the design of multi-agent systems, in which the overall functionality of the multi-agent system emerges from the individual behaviour of rational agents that pursue their own goals.

On the one hand, solutions inspired by the Social Sciences are implemented in current-day multi-agent systems, the most well-know example being the contract net protocol for delegation of tasks (Smith, 1980). On the other hand, it seems that the Social Sciences themselves currently do not possess the definite answer to the question of how social functions emerge from the behaviour of individual agents. Moreover, it may be the case that, to obtain such an answer, the Social Sciences need to rely on advances in the area of multi-agent systems. The relation between the Social Sciences and multi-agent systems research thus seems to be a beneficial one for both areas (Eck, 1998).

From an Artificial Intelligence point of view, the two promises together offer a new level of abstraction for communication and co-ordination in knowledge-intensive distributed systems comparable to the knowledge level (Newell, 1984): the social level. The Social Sciences may become an additional source of inspiration for Artificial Intelligence, complementing Psychology (Cognitive Science).

### **1.3 The Practise of Multi-Agent Systems**

Practitioners in the area of multi-agent systems research currently face a number of issues that need to be solved:

- First of all, the notion of an agent is used both as a metaphor and as a reference to a type of technology. This is a source of much confusion in the field of multi-agent systems, and of even more confusion in related research areas. In the field of agent technology, the main focus is on the development

### 1.3: The Practise of Multi-Agent Systems

of software components that are agents. Such software components are programmed to provide autonomy, reactivity, pro-activity, social ability and (often) mobility. Software engineers who employ agent-based technology only have to refine the components by adding task-specific functionality. Agent technology is best viewed as an extension of object technology. Much research in the area of multi-agent systems, however, is not directly concerned with the development of software components that are agents. Instead, the notion of an agent is used as a metaphor: software systems are analysed and designed as if they consist of autonomous, rational agents. Research topics include the development of theories about the behaviour of such agents, about behaviour that emerges from the interplay between agents in a multi-agent system, and generic architectures for such agents. However, there is no commitment to a specific technology for implementing multi-agent systems, although, of course agent technology is often the most promising candidate.

- Currently, there is no solution to the problem of how the behaviour of a complex multi-agent system can be predicted based on the goals of individual agents. In terms of the Social Sciences, there is no *middle ground theory* (Castelfranchi & Conte, 1996). The promise that multi-agent systems research will enable the Social Sciences to develop such a theory has not yet been met.
- As stated in Section 1.1.1, autonomy is an essential property of an agent. However, currently there is no philosophically satisfactory definition of autonomy. (See (Dennett, 1984) for a contemporary philosophical treatment of autonomy.) The field of multi-agent systems research is often criticised for its reliance on such a vague notion. For practitioners, it is difficult to determine if a specific software component is an agent, or “just a program” (Franklin & Graesser, 1997).
- A multi-agent system consists of agents that are active simultaneously. Consequently, the analysis and design of multi-agent system also involves the analysis and design of concurrent behaviour. The central focus in the area of concurrent behaviour is how to relate the different time scales (or clocks, or, in other words, behaviour) of the different simultaneous processes. In multi-agent systems, this problem has to be solved without making (implicit or explicit) assumptions about synchronicity that are not compatible with the autonomy of agents. As explained in the next section, this issue is closely related to the research aim of this thesis.
- Only recently (Jennings, 1999) has the multi-agent systems research community realised that the development of a multi-agent system not only involves state-of-the-art Artificial Intelligence and results in agent theory, but also complex software engineering. Software engineering aspects of

multi-agent system development have therefore only recently been addressed, and results are preliminary.

Wooldridge and Jennings (1998) present a list of many more, as they call it, pitfalls of agent-oriented software development.

This thesis develops an approach for the representation of the behaviour of agents and of multi-agent systems in a precise, mathematical way. This approach facilitates the development of solutions for the issues listed above.

## **1.4 Research Aims, Method, Relevance, Context and Demarcation**

This section discusses the aims of the research presented in this thesis, as well as the relevance of the research aims and the method followed. The context in which the research is placed is briefly discussed and a demarcation of the performed research is sketched.

### **1.4.1 Research Aims**

The research aim of this thesis is to develop a *formal, compositional, semantic structure for multi-agent systems dynamics*. This research aim is characterised in more detail as follows:

- As indicated at the beginning of this chapter, agents are first and foremost active entities. Therefore, the primary characteristic of a multi-agent system is that different and simultaneous processes take place. These processes result in a continuous change of properties of the agents in a multi-agent system. The focus of this thesis is on these continuous processes of change, or, in other words, on the *dynamics* of a multi-agent system. The activity of a multi-agent system or of an individual agent is also called the *behaviour* of the multi-agent system or agent. In this thesis, behaviour, activity and dynamics are used as synonyms.
- Design of a multi-agent system is based on a model in which different aspects of the multi-agent system, e.g. the behaviour, knowledge, intentions and goals of the agents, are explicitly distinguished. Which aspects of a multi-agent system are covered by a model depend on a number of factors, such as, for instance, the intended usage of the model and the level of abstraction. A model of a multi-agent system is itself intangible. However, a model can be given a tangible form by a description, or *specification*, of the model. A description of aspects of the dynamics of a multi-agent system is called a specification of multi-agent systems dynamics.
- A specification is an expression formulated in a (natural or formal) language, called the specification language. As such, a specification itself is merely a syntactic entity, or, in other words, a string of symbols. The *raison d'être* of the specification is its *intended meaning*: the model it denotes. Thus,

#### 1.4: Research Aims, Method, Relevance, Context and Demarcation

the need arises to express the intended meaning of a specification. A *semantic structure* provides a set of constructs and relations between these constructs with which the intended meaning of a specification can be defined.

- The semantic structure is identified as a *compositional* semantic structure, which indicates the two most important properties of the semantic structure. First, the primary construct in the semantic structure is the *component*: the basic building block for models of multi-agent systems. In this thesis, multi-agent systems are assumed to be modelled as *compositional systems*. Second, the semantic structure supports the principle of *compositionality*: the dynamics of a system composed of a number of components is defined by a composition relation in terms of the dynamics of these components.
- The semantic structure is also identified as a *formal* semantic structure. The constructs provided by the structure to define the meaning of a specification are mathematical constructs. The use of mathematical constructs enables a precise and unambiguous definition of the intended meaning of a specification.

To summarise, the primary research aim of this thesis is *to develop a set of mathematical constructs and relations between these constructs that can be used to define the intended meaning of a specification for a multi-agent system*. Such a set of constructs is called a semantic structure. As a specification is an expression formulated in a specific language (the *specification language*), a semantic structure can also be used to define the intended meaning (semantics) of the specification language. The semantics of a language, by definition, consists of the semantics of expressions in that language: the semantic structure serves as the *semantic domain* of the specification language.

To illustrate possible applications of the semantic structure, in this thesis, two example multi-agent systems are modelled. First, a model for co-ordinating exclusive access to resources in a multi-agent system is presented in Chapter 10. Second, in Chapter 11, a society of a relatively large number of relatively simple agents is modelled, with the purpose of studying emergent social behaviour.

##### 1.4.2 Method

The method employed in this thesis to develop the semantic structure consists of the following steps:

- The starting point for the development of a semantic structure consists of a commitment to a set of constructs and relations between these constructs. Together, the set of constructs and relations constitute the semantic structure. To establish such a commitment, different possibilities for specific sets of constructs and relations are identified. This commitment is made by the developers of the semantic structure and is fixed for all applications of

the semantic structure. The primary commitment in this thesis is the commitment to components as the basic construct. This commitment supports the basic assumption mentioned above: multi-agent systems are modelled as compositional systems.

- The specific set of constructs and relations between these constructs, which is fixed for the semantic structure, is designed to be as general as possible. As a consequence, for each application of the semantic structure, a number of additional choices have to be made that determine how the semantic structure is applied for the specific application. These choices are made by a user of the semantic structure and are fixed for each specific application. After establishing the set of constructs and relations between these constructs, this thesis explores different choices for applications of the semantic structure.
- The set of constructs, properties and relations is described in a detailed, mathematical way. (This constitutes the main part of this thesis.)
- The semantic structure is applied to provide a semantic domain for the compositional multi-agent systems development method DESIRE. (See (Brazier, Jonker & Treur, 1998) for an overview of the principles behind DESIRE. A generic agent model modelled in DESIRE is described in (Brazier, Jonker & Treur, 2000). The generic agent model has been applied in many domains including electricity transportation management (Brazier, Dunin-Keplicz, Jennings & Treur, 1997), electricity load balance management (Brazier, Cornelissen, Gustavsson, Jonker, Lindeberg, Polak & Treur, 2000) and as a basis for the co-operative agent model which has been applied, for example, in distributed call centre support (Brazier, Cornelissen, Jonker & Treur, 2000). An earlier version of DESIRE is described in (Langevelde, Philips & Treur, 1992)).
- The last step consists of the analysis of two example multi-agent systems modelled in DESIRE. The first system multi-agent system consists of agents that have to co-operate to obtain mutually exclusive access to a shared resource. The second multi-agent system represents a society of 30 simple agents used for experimental Social Science research.

### **1.4.3 Relevance of the Research Aim**

As explained above, dynamics are an important aspect of every multi-agent system. The dynamics of a multi-agent system emerge from complex mutual influence between concurrent processes in a system. Wooldridge (1996, Section 5) acknowledges the importance of concurrency in multi-agent systems as follows: “The issue of concurrency has scarcely been addressed in formal treatments of DAI, and yet concurrency is at the very heart of the area.”

#### 1.4: Research Aims, Method, Relevance, Context and Demarcation

The formal representation of concurrency has long been studied in mainstream Theoretical Computer Science. However, the results obtained are not directly applicable to concurrency in multi-agent systems due to several specific characteristics of multi-agent systems and the individual agents of which a multi-agent system is constituted. These special characteristics can be described in the form of requirements for a semantic structure for multi-agent systems dynamics as follows:

- A multi-agent system consists of a number of agents and the environment shared by these agents. The agents deliberately and mutually influence one another by executing actions in the environment, observing the environment, and by communicating with one another. Communication, action execution, and observation are together called interaction. Individual agents have to deliberate to evaluate information received from other agents and to determine which actions to execute and information to exchange to pursue their own goals. The dynamics of a multi-agent system thus consists of *deliberation and interaction*. A semantic structure for multi-agent systems dynamics should therefore provide a natural representation of deliberation and interaction.
- An important phenomenon of multi-agent systems is that a multi-agent system is composed of a number of heterogeneous agents: e.g., agents may have completely different structure and they may perform different tasks, either collaboratively or in competition. A multi-agent system is often not designed by a single designer. In general, agents in a multi-agent system are themselves complex systems consisting of a number of components, such as e.g. planning components and knowledge base components. Analysis and design of such heterogeneous systems is greatly improved by the *compositionality principle*. A semantic structure for agent dynamics should support this compositionality principle.
- In, for instance, the strong notion of agency put forward by Wooldridge and Jennings mentioned at the beginning of this chapter, an agent is characterised as an (active, computational) entity whose *state* is described by mentalistic notions. This characterisation suggests that the dynamics of multi-agent systems should be modelled in terms of *states and state transformations*. This view may be contrasted with a number of common semantic structures for dynamics in mainstream Theoretical Computer Science, where dynamics is often modelled as partial or total orders of named actions<sup>2</sup> (also called events), without an explicit notion of state and

---

<sup>2</sup> There seems to be a difference between the notion of action in Theoretical Computer Science and Multi-Agent Systems. In Theoretical Computer Science, any observable activity of an agent is called an action, whether or not this activity influences any other part of the

state transformations. (Nevertheless, Burkhard (1993) chooses to use events as the basis for his semantic structure for multi-agent systems.) In this thesis, the dynamics of a multi-agent system are defined in terms of state and state transformation.

- The semantic structure should model dynamics as a composition of *local* behaviours of agents and agent components, together with interrelationships between these local behaviours. This *locality* is motivated by the following two phenomena. In the first place, as already mentioned, most multi-agent systems consist of a heterogeneous collection of agents. Therefore, a global view is difficult to obtain, while a collection of local views with interrelationships is more natural. In the second place, the agents in a multi-agent system are often widely separated. Due to this wide-area distribution, communication is asynchronous and cannot be assumed to be durationless. Under these conditions, as explained in Chapter 7, global time is not available. Consequently, if a notion of global state is desired, it has to be defined such that the existence of global time is not implied by the definition. In this thesis, a notion of global state is defined using dependence relations between local states (a form of locality). Most common semantic structures for dynamics in Theoretical Computer Science (often implicitly) assume that global time is available and thus do not support this form of locality.

The use of dependence relations to define a notion of global state is inspired by the *true concurrency* view on modelling distributed systems (see among others Pratt, 1986; Schwarz & Mattern, 1994), as opposed to the *atomic mutual exclusion*, or *interleaving*, view. Several authors (e.g., Pratt 1986) argue that truly concurrent processes are not properly modelled in the atomic mutual exclusion view, and point at disadvantages of this view. In the atomic mutual exclusion view, concurrency is modelled as nondeterministic global choice between different sequences of atomic actions of individual processors in a distributed system. The atomic mutual exclusion view implicitly assumes that a notion of global time is available, i.e., the notion of time implied by the (global) sequences of atomic actions. As concurrency is modelled as nondeterministic choice, made by a (sequential) global automaton representing an entire distributed system, the atomic mutual exclusion view does not support the principle of locality. Moreover, in the atomic mutual exclusion view, the behaviour of a multi-agent system may possibly change if actions are refined. (In the atomic mutual exclusion view, the sequences of atomic actions are not preserved under action refinement (Castellano *et al.*, 1987; Glabbeek & Goltz, 1989). For example, consider two actions *a* and *b* that are performed concurrently. Seen as atomic actions, in the interleaving view either

---

system. However, in the multi-agent systems discipline, usually only agent activities that attempt to change the state of the environment are called actions.

#### 1.4: Research Aims, Method, Relevance, Context and Demarcation

$a$  precedes  $b$  or  $b$  precedes  $a$ . However, if  $a$  is refined to the sequence of actions  $a_1a_2$ , a possible behaviour is  $a_1ba_2$ : neither  $a_1a_2$  precedes  $b$  nor the other way around.) Consequently, when applying the atomic mutual exclusion view, the behaviour of a multi-agent system always needs to be described at the most detailed level, or it has to be accepted that behaviour is not preserved if later on, a more detailed description is developed. The true concurrency view does not have these drawbacks.

The relevance of the research aim can also be indicated by reference to the literature. Wooldridge (1996, Section 5) acknowledges the special position of multi-agent systems (which he refers to as Distributed AI or DAI systems): "... DAI systems are *not* simply concurrent systems." The development of a formal semantic structure specifically designed for multi-agent systems dynamics should help to remedy this situation.

##### 1.4.4 Research Context

The research presented in this thesis is related to three disciplines: Artificial Intelligence, Theoretical Computer Science and Multi-Agent Systems. More specifically, the relations between these disciplines and the development of a semantic structure for multi-agent systems dynamics is as follows:

- In Artificial Intelligence, locality is identified as a concept in modelling common-sense contextual reasoning, as witnessed by e.g. (Giunchiglia, 1993).
- Theoretical Computer Science is used as a source of concepts for the representation of concurrency, e.g. (Lamport, 1978, 1986; Schwarz & Mattern, 1994). As an aside, the work of Lamport, Schwarz and Mattern referenced can also be classified in the area of Distributed Systems research.
- In the area of Multi-Agent Systems, a number of formal specification languages exist that have been specifically designed for this area. In Chapter 12, a comparison with a number of alternative approaches is provided. Surveys of this area have been published by for example Wooldridge and Jennings (1995b) and O'Hare and Jennings (1996).

##### 1.4.5 Demarcation

With respect to the demarcation of the research presented in this thesis, please note:

- This thesis approaches the subject of multi-agent systems from an Artificial Intelligence point of view. Artificial Intelligence is an engineering discipline: the focus is on *designing* systems that exhibit intelligent behaviour. Consequently, the subject of multi-agent systems is approached in this thesis with the ultimate goal of contributing to the development of engineering

methods that support the design of multi-agent systems. The thesis is not pursuing to contribute to the development of Sociology or other branches of the Social Sciences. However, the semantic structure developed in this thesis can be used to model multi-agent systems that simulate social phenomena. An example is provided in Chapter 11.

- The semantic structure developed in this thesis is not biased towards specific processes in a multi-agent system, such as e.g. goal adaptation or emergence of norms. Instead, the semantic structure is general: it is applicable to all processes in a multi-agent system. Specific processes such as goal adaptation and emergence of norms may be studied in the context of applications of the semantic structure. The multi-agent systems analysed in Chapter 10 and Chapter 11 are examples of such studies.
- As stated in Section 1.4.3, the semantic structure is state-based, which conforms to the common approach of characterising agents in terms of their (mentalistic) state. However, the semantic structure abstracts from the contents of these states. In other words, questions of which (probably mentalistic) concepts to use to describe an agent's state are not addressed in this thesis.
- The semantic structure developed in this thesis can be compared to formalisms developed to study concurrency. The semantic structure developed in this thesis is specifically designed for the domain of dynamics of multi-agent systems. More general applicability of the semantic structure is not investigated.
- Chapter 9 presents a formal language (based on temporal logic) for precisely describing the intended dynamics of components. However, the thesis does not aim at developing a logic for reasoning about (the dynamics of) multi-agent systems. As a consequence, complexity, expressive power and inference relations of the language presented in Chapter 9 are not discussed in this thesis.

## **1.5 Outline of the Thesis**

The structure of this thesis is as follows:

Chapter 2 introduces compositional systems and presents a number of commitments with respect to specific properties of compositional systems that can be represented by the semantic structure.

Chapter 3 discusses how multi-agent systems can be represented as compositional systems. To apply the semantic structure in the area of multi-agent systems (which is the main application of the semantic structure), it is necessary to represent a

### *1.5: Outline of the Thesis*

multi-agent system as a compositional system, because compositions of components are the primary construct in the semantic structure.

Chapter 4 provides an overview of the semantic structure and introduces an example that is used in Chapter 5 to Chapter 9. The overview of the semantic structure summarises the central principles and the formal techniques employed in the semantic structure.

Chapter 5 formally defines the main constructs in the semantic structure. First, components and information transmission are formally described. After that, three views on the behaviour of a compositional system are defined, and relations between these views are discussed.

Chapter 6 discusses how different properties of information transmission, such as lossless information transmission and order-preserving information transmission, are formally represented in the semantic structure. The commitments made in Chapter 2 are formally defined in this chapter.

Chapter 7 develops a notion of a global state of a compositional system. This notion of global state does not assume that there is a global notion of time accessible to all components. Moreover, a global state is derived from the three views on the behaviour of a compositional system, which consist of local states. Thus, in the semantic structure, global states are derived from local states, instead of the other way around.

Chapter 8 develops additional constructs in the semantic structure for the representation of control. In almost every multi-agent system, some agents exercise control over other agents. The semantic structure presented in Chapter 5 and Chapter 6 enables the representation of control. Chapter 8 develops a refinement of the semantic structure that enables the representation of control in a separated, domain-independent way that supports reuse and maintainability.

Chapter 9 shows an application of the semantic structure, in which semantics are developed for the DESIRE modelling framework.

Chapter 10 presents the first example of a multi-agent system modelled with the DESIRE modelling framework. The multi-agent system in this example consists of agents that have to obtain mutually exclusive access to a shared resource.

Chapter 11 presents the second example of a multi-agent system modelled with the DESIRE modelling framework. The multi-agent system in this example represents a society of 30 simple agents. This society is used to replicate experimental Social Science research (Cesta, Miceli & Rizzo, 1996a).

Chapter 12 compares the semantic structure with a number of similar approaches. Additionally, conclusions and directions for further research are presented.

Discussions presented in Chapter 2, Chapter 8 and Chapter 9 of commitments made in the semantic structure, as well as alternatives to these commitments and

the modelling choices presented in Chapter 3 and Chapter 8, are adapted from (Brazier, Eck & Treur, 1996). A strongly abridged version of Chapters 4 to 7 can be found in (Brazier, Eck & Treur, 2001b). Details of the DESIRE modelling framework presented in Chapter 9 are taken from various publications of the DESIRE group at the Vrije Universiteit, Amsterdam. Figure 9.1 was designed by Niek Wijngaards. Chapter 10 is an extended version of (Brazier, Eck & Treur, 1997b). Chapter 11 is an extended version of (Brazier, Eck & Treur, 2001a), which also appeared in a much shorter version as (Brazier, Eck & Treur, 1997a). Figure 11.6 in Chapter 11 is taken from (Cesta, Micelli & Rizzo, 1996a). The author thanks Amadeo Cesta, Maria Miceli and Paola Rizzo for providing the raw data that is used in the experimentation. The description of OSL in Chapter 12 is taken from (Eck, Engelfriet, Fensel, Harmelen, Venema & Willems, in press), which also appeared in a much shorter version as (Eck, Engelfriet, Fensel, Harmelen, Venema & Willems, 1998).

### *1.5: Outline of the Thesis*