Prologue

Multiagent Systems and Distributed Artificial Intelligence

Since its inception in the mid to late 1970s distributed artificial intelligence (DAI) evolved and diversified rapidly. Today it is an established and promising research and application field which brings together and draws on results, concepts, and ideas from many disciplines, including artificial intelligence (AI), computer science, sociology, economics, organization and management science, and philosophy. Its broad scope and multi-disciplinary nature make it difficult to precisely characterize DAI in a few words. The following definition is intended to serve as a starting point for exploring this arena and as a constant point of reference for reading through this book:

> *DAI is the study, construction, and application of multiagent systems, that is, systems in which several interacting, intelligent agents pursue some set of goals or perform some set of tasks.*

An agent is a computational entity such as a software program or a robot that can be viewed as perceiving and acting upon its environment and that is autonomous in that its behavior at least partially depends on its own experience. As an intelligent entity, an agent operates flexibly and rationally in a variety of environmental circumstances given its perceptual and effectual equipment. Behavioral flexibility and rationality are achieved by an agent on the basis of key processes such as problem solving, planning, decision making, and learning. As an interacting entity, an agent can be affected in its activities by other agents and perhaps by humans. A key pattern of interaction in multiagent systems is goal- and task-oriented coordination, both in cooperative and in competitive situations. In the case of cooperation several agents try to combine their efforts to accomplish as a group what the individuals cannot, and in the case of competition several agents try to get what only some of them can have. The long-term goal of DAI is to develop mechanisms and methods that enable agents to interact as well as humans (or even better), and to understand interaction among intelligent entities whether they are computational, human, or both. This goal raises a number of challenging issues that all are centered around the elementary question of *when and how to interact with whom*.

Two main reasons to deal with DAI can be identified, and these two reasons are the primary driving forces behind the growth of this field in recent years. The first is that multiagent systems have the capacity to play a key role in current and future computer science and its application. Modern computing platforms and information environments are distributed, large, open, and heterogeneous. Computers are no longer stand-alone systems, but have became tightly connected both with each other and their users. The increasing complexity of computer and information systems goes together with an increasing complexity of their applications. These often exceed the level of conventional, centralized computing because they require, for instance, the processing of huge amounts of data, or of data that arises at geographically distinct locations. To cope with such applications, computers have to act more as "individuals" or agents, rather than just "parts." The technologies that DAI promises to provide are among those that are urgently needed for managing high-level interaction in and intricate applications for modern computing and information processing systems.

The second reason is that multiagent systems have the capacity to play an important role in developing and analyzing models and theories of interactivity in human societies. Humans interact in various ways and at many levels: for instance, they observe and model one another, they request and provide information, they negotiate and discuss, they develop shared views of their environment, they detect and resolve conflicts, and they form and dissolve organizational structures such as teams, committees, and economies. Many interactive processes among humans are still poorly understood, although they are an integrated part of our everyday life. DAI technologies enable us to explore their sociological and psychological foundations.

Intelligent Agents that Interact

To make the above considerations more concrete, a closer look has to be taken on multiagent systems and thus on "interacting, intelligent agents":

- "Agents" are autonomous, computational entities that can be viewed as perceiving their environment through sensors and acting upon their environment through effectors. To say that agents are computational entities simply means that they physically exist in the form of programs that run on computing devices. To say that they are autonomous means that to some extent they have control over their behavior and can act without the intervention of humans and other systems. Agents pursue goals or carry out tasks in order to meet their design objectives, and in general these goals and tasks can be supplementary as well as conflicting.
- "Intelligent" indicates that the agents pursue their goals and execute their tasks such that they optimize some given performance measures. To say that agents are intelligent does not mean that they are omniscient or omnipotent, nor does it mean that they never fail. Rather, it means that they operate flexibly and rationally in a variety of environmental circumstances, given the information they have and their perceptual and effectual capabilities. A major

focus of DAI therefore is on processes such as problem solving, planning, search, decision making, and learning that make it possible for agents to show flexibility and rationality in their behavior, and on the realization of such processes in multiagent scenarios.

- "Interacting" indicates that the agents may be affected by other agents or perhaps by humans in pursuing their goals and executing their tasks. Interaction can take place indirectly through the environment in which they are embedded (e.g., by observing one another or by carrying out an action that modifies the environmental state) or directly through a shared language (e.g., by providing information in which other agents are interested or which confuses other agents). DAI primarily focuses on coordination as a form of interaction that is particularly important with respect to goal attainment and task completion. The purpose of coordination is to achieve or avoid states of affairs that are considered as desirable or undesirable by one or several agents. To coordinate their goals and tasks, agents have to explicitly take dependencies among their activities into consideration. Two basic, contrasting patterns of coordination are cooperation and competition. In the case of cooperation, several agents work together and draw on the broad collection of their knowledge and capabilities to achieve a common goal. Against that, in the case of competition, several agents work against each other because their goals are conflicting. Cooperating agents try to accomplish as a team what the individuals cannot, and so fail or succeed together. Competitive agents try to maximize their own benefit at the expense of others, and so the success of one implies the failure of others.

It has to be stressed that there is no universally accepted definition of agency or of intelligence, and the above explanations are just intended to show how these terms are generally understood and what is generally considered as essential for an entity to be an intelligent agent. The concept of an intelligent agent that interacts allows various degrees of degradation, and is perhaps best viewed as a "guideline" for designing and analyzing systems rather than an "instruction" that allows no variation, or a precise "criterion" that always allows one to determine whether an object does or does not fulfill it. A useful catalog of agent theories and systems is provided in [45]. Another popular text on agents is [38, Chapter 2]. A recent overview of key themes in agent research is given in [22].

In [25] the following major characteristics of multiagent systems are identified:

- each agent has just incomplete information and is restricted in its capabilities;
- system control is distributed;
- data is decentralized; and
- computation is asynchronous.

Multiagent systems can differ in the agents themselves, the interactions among the agents, and the environments in which the agents act. The following table gives an overview of some attributes of multiagent systems, together with their potential range (an extensive overview is offered in [22]):

	attribute	**range**
agents	number	from two upward
	uniformity	homogeneous ... heterogeneous
	goals	contradicting ... complementary
	architecture	reactive ... deliberative
	abilities (sensors, effectors, cognition)	simple ... advanced
interaction	frequency	low ... high
	persistence	short-term ... long-term
	level	signal-passing ... knowledge-intensive
	pattern (flow of data and control)	decentralized ... hierarchical
	variability	fixed ... changeable
	purpose	competitive ... cooperative
environment	predictability	forseeable ... unforseeable
	accessibility and knowability	unlimited ... limited
	dynamics	fixed ... variable
	diversity	poor ... rich
	availability of resources	restricted ... ample

Traditionally two primary types of DAI systems have been distinguished [2]: multiagent systems in which several agents coordinate their knowledge and activities and reason about the processes of coordination; and distributed problem solving systems in which the work of solving a particular problem is divided among a number of nodes that divide and share knowledge about the problem and the developing solution. Whereas initially the emphasis of work on multiagent systems was on behavior coordination, the emphasis of work on distributed problem solving systems was on task decomposition and solution synthesis. The modern concept of multiagent systems as described above covers both types of systems. For that reason, and in accordance with contemporary usage, in this book no explicit distinction is made between multiagent systems and distributed problem solving systems, and the terms multiagent system and DAI system are used synonymously.

The role that the concept of a multiagent system plays in DAI is comparable to the role that the concept of an individual agent plays in traditional AI (see, e.g., [33, 36, 38]). Broadly construed, both DAI and traditional AI deal with computational aspects of intelligence, but they do so from different points of view and under different assumptions. Where traditional AI concentrates on agents as "intelligent stand-alone systems" and on intelligence as a property of systems that act in isolation, DAI concentrates on agents as "intelligent connected systems" and on intelligence as a property of systems that interact. Where traditional AI focuses on "cognitive processes" within individuals, DAI focuses on "social processes" in groups of individuals. Where traditional AI considers systems having a single locus of internal reasoning and control and requiring just minimal help from others to act successfully, DAI considers systems in which reasoning and control is distributed and successful activity is a joint effort. And where traditional AI uses psychology and behaviorism for ideas, inspiration, and metaphor, DAI uses sociology and economics. In this way, DAI is not so much a specialization of traditional AI, but a generalization of it.

Challenging Issues

To build a multiagent system in which the agents "do what they should do" turns out to be particularly difficult in the light of the basic system characteristics mentioned above. The only way to cope with these characteristics is to enable the agents to interact appropriately, and thus the elementary question faced by DAI is *When and how should which agents interact—cooperate and compete—to successfully meet their design objectives?* Based on the common distinction between the *"micro"* or agent level and the *"macro"* or group level (e.g., see [31]), in principle one can follow two different routes to answer this question:

- bottom up: to search for specific agent-level capabilities that result in appropriate interaction at the overall group level; or
- top down: to search for specific group-level rules—called conventions, norms, and so on—that appropriately constrain the interaction repertoire at the level of the individual agents.

(The question how agent-level—individual—activity and group-level—societal—rules and structures are related to each other is known as the micro-macro problem in sociology.) No matter which route is chosen, this question raises several challenging, intertwined issues (items 1 to 5 were first mentioned in [2], and item 6 and items 7 and 8 were additionally formulated in [31] and [25], respectively):

1. How to enable agents to decompose their goals and tasks, to allocate sub-goals and sub-tasks to other agents, and to synthesize partial results and solutions.
2. How to enable agents to communicate. What communication languages and protocols to use.
3. How to enable agents to represent and reason about the actions, plans, and knowledge of other agents in order to appropriately interact with them.
4. How to enable agents to represent and reason about the state of their interaction processes. How to enable them to find out whether they have achieved progress in their coordination efforts, and how to enable them to improve the state of their coordination and to act coherently.

5. How to enable agents to recognize and reconcile disparate viewpoints and conflicts. How to syntheze views and results.
6. How to engineer and constrain practical multiagent systems. How to design technology platforms and development methodologies for DAI.
7. How to effectively balance local computation and communication.
8. How to avoid or mitigate harmful (e.g., chaotic or oscillatory) overall system behavior.
9. How to enable agents to negotiate and contract. What negotiation and contract protocols should they use.
10. How to enable agents to form and dissolve organizational structures—teams, alliances, and so on—that are suited for attaining their goals and completing their tasks.
11. How to formally describe multiagent systems and the interactions among agents. How to make sure that they are correctly specified.
12. How to realize "intelligent processes" such as problem solving, planning, decision making, and learning in multiagent contexts. How to enable agents to collectively carry out such processes in a coherent way.

To provide solutions to these issues is the core request of DAI.

Applications

Many existing and potential industrial and commercial applications for DAI and multiagent systems are described in the literature (e.g., see [23, 24] and also [26]). Basically following [25] (here the readers find a number of pointers to specific work), examples of such applications are:

- Electronic commerce and electronic markets, where "buyer" and "seller" agents purchase and sell goods on behalf of their users.
- Real-time monitoring and management of telecommunication networks, where agents are responsible, e.g., for call forwarding and signal switching and transmission.
- Modelling and optimization of in-house, in-town, national- or world-wide transportation systems, where agents represent, e.g., the transportation vehicles or the goods or customers to be transported.
- Information handling in information environments like the Internet, where multiple agents are responsible, e.g., for information filtering and gathering.
- Improving the flow of urban or air traffic, where agents are responsible for appropriately interpreting data arising at different sensor stations.
- Automated meeting scheduling, where agents act on behalf of their users to fix meeting details like location, time, and agenda.

- Optimization of industrial manufacturing and production processes like shop-floor scheduling or supply chain management, where agents represent, e.g., different workcells or whole enterprises.
- Analysis of business processes within or between enterprises, where agents represent the people or the distinct departments involved in these processes in different stages and at different levels.
- Electronic entertainment and interactive, virtual reality-based computer games, where, e.g., animated agents equipped with different characters play against each other or against humans.
- Design and re-engineering of information- and control-flow patterns in large-scale natural, technical, and hybrid organizations, where agents represent the entities responsible for these patterns.
- Investigation of social aspects of intelligence and simulation of complex social phenomena such as the evolution of roles, norms, and organzational structures, where agents take on the role of the members of the natural societies under consideration.

What these applications have in common is that they show one or several of the following features [2]:

- *Inherent Distribution* – They are inherently distributed in the sense that the data and information to be processed
  - arise at geographically different locations ("spatial distribution");
  - arise at different times ("temporal distribution");
  - are structured into clusters whose access and use requires familiarity with different ontologies and languages ("semantic distribution"); and/or
  - are structured into clusters whose access and use requires different perceptual, effectual, and cognitive capabilities ("functional distribution").
- *Inherent Complexity* – They are inherently complex in the sense that they are too large to be solved by a single, centralized system because of limitations available at a given level of hardware or software technology. To enlarge a centralized system such that it meets the requirements of inherently complex applications usually is very difficult, time-consuming, and costly. Moreover, such an enlargement often results in solutions that are brittle and that become useless as soon as the application requirements change only slightly.

Solving inherently distributed and complex applications in a centralized way is obviously not only counter-intuitive, but often not even possible at all. The alternative is to distribute the solution process across multiple entities capable of intelligent coordination—and DAI aims at developing technologies and methodologies for realizing this alternative in a very natural way [15].

Rationales for Multiagent Systems

The two major reasons that cause people to study multiagent systems are:

- *Technological and Application Needs* – Multiagent systems offer a promising and innovative way to understand, manage, and use distributed, large-scale, dynamic, open, and heterogeneous computing and information systems. The Internet is the most prominent example of such systems; other examples are multi-database systems and in-house information systems. Computers and computer applications play an increasingly important and influencing part in our everyday life, as they become more powerful and more tightly connected both with each other through long-range and local-area networks and with humans through user-interfaces. These systems are too complex to be completely characterized and precisely described. As their control becomes more and more decentralized, their components act more and more like "individuals" that deserve attributes like autonomous, rational, intelligent, and so forth rather than just as "parts." DAI does not only aim at providing know-how for building sophisticated interactive systems from scratch, but also for interconnecting existing legacy systems such that they coherently act as a whole. Moreover, like no other discipline, DAI aims at providing solutions to inherently distributed and inherently complex applications. As we saw above, these applications are hard to solve with centralized computing technology. Many real world applications, if not most, fall into this class, and they are present in many domains such as scheduling, manufacturing, control, diagnosis, and logistics.
- *Natural View of Intelligent Systems* – Multiagent systems offer a natural way to view and characterize intelligent systems. Intelligence and interaction are deeply and inevitably coupled, and multiagent systems reflect this insight. Natural intelligent systems, like humans, do not function in isolation. Instead, they are at the very least a part of the environment in which they and other intelligent systems operate. Humans interact in various ways and at various levels, and most of what humans have achieved is a result of interaction. DAI can provide insights and understanding about poorly understood interactions among natural, intelligent beings, as they organize themselves into various groups, committees, societies, and economies in order to achieve improvement.

In addition, multiagent systems, as distributed systems, have the capacity to offer several desirable properties [2]:

- *Speed-up and Efficiency* – Agents can operate asynchronously and in parallel, and this can result in an increased overall speed (provided that the overhead of necessary coordination does not outweigh this gain).
- *Robustness and Reliability* – The failure of one or several agents does not necessarily make the overall system useless, because other agents already available in the system may take over their part.

- *Scalability and Flexibility* – The system can be adopted to an increased problem size by adding new agents, and this does not necessarily affect the operationality of the other agents.
- *Costs* – It may be much more cost-effective than a centralized system, since it could be composed of simple subsystems of low unit cost.
- *Development and Reusability* – Individual agents can be developed separately by specialists (either from scratch or on the basis of already available hardware and/or software facilities), the overall system can be tested and maintained more easily, and it may be possible to reconfigure and reuse agents in different application scenarios.

The available computer and network technology forms a sound platform for realizing these systems. In particular, recent developments in object-oriented programming, parallel and distributed computing, and mobile computing, as well as ongoing progress in programming and computing standardization efforts such as KSE (e.g., http://www.cs.umbc.edu/kse/), FIPA (e.g., http://drogo.cselt.stet.it/fipa/), and CORBA (e.g., http://www.rhein-neckar.de/~cetus/oo_corba.html and http://industry.ebi.ac.uk/~corba/) are expected to further improve the possibilities of implementing and applying DAI techniques and methods.

A Guide to This Book

The Chapters

The book is divided into two parts. Part I contains nine chapters, each treating a core theme in the field of multiagent systems and DAI:

- Chapter 1 concentrates on agents—the “micro” level referred to above.
- Chapter 2 expands the considerations of Chapter 1 by focusing on systems of agents and the computational infrastructure required for interaction—the “macro” level referred to above.
- Chapters 3 to 6 address elementary “intelligent activities” and their realization in multiagent systems, namely,
  - problem solving and planning,
  - search,
  - decision making, and
  - learning.
- Chapter 7 shows how processes of organizing, as they occur among agents and humans, can be computationally modelled.

- Chapter 8 describes formal methods for studying and constructing agents and multiagent systems.
- Chapter 9 concentrates on applications of agent and multiagent system technology.

Part II includes chapters on closely related, selected themes from computer science and software engineering:

- Chapter 10 focuses on groupware and computer supported cooperative work.
- Chapter 11 concentrates on distributed decision support systems.
- Chapter 12 discusses various issues of concurrent programming.
- Chapter 13 describes distributed control algorithms.

The relevance of these themes for the field can be easily seen. Agents in a multiagent system often have to coordinate their activities, and so there is a need for technologies that support them in acting coherently as a group; additionally, groupware and computer supported cooperative work constitute an important application domain for multiagent systems. Agents in a multiagent system often have to jointly make decisions, and so there is a need for technologies that support them in their distributed decision processes; moreover, distributed decision making is another obvious application domain for multiagent systems. There is a need for powerful concurrent programming techniques that allow to efficiently implement multiagent systems as parallel and distributed systems. And finally, there is an obvious need for mechanisms and methods that enable agents to control their distributed computations.

In the following, the individual chapters and their themes are motivated in more detail.

Chapter 1, "Intelligent Agents" by Michael Wooldridge – This chapter aims to introduce the reader to the basic issues surrounding the design and implementation of intelligent agents. It begins by motivating the idea of an agent, presents a definition of agents and intelligent agents, and then discusses the relationship between agents and other software paradigms (in particular, objects and expert systems). The chapter then goes on to discuss four major approaches to building agents. First, *logic based architectures* are reviewed. In logic based architectures, decision-making is viewed as logical deduction: the process of deciding which action to perform is reduced to a theorem proving problem. Such architectures have the advantage of semantic clarity, and in addition allow us to bring to bear all the apparatus of logic and theorem proving that has been developed in AI and computer science over the years. However, such architectures suffer from a number of drawbacks, not the least of which being that purely logical architectures do not seem well suited to domains that are subject to real time constraints. Second, *reactive architectures* are discussed. The characteristic of such architectures is that they eschew symbolic representations and reasoning in favour of a closer relationship between agent perception and action. Such architectures are more economical in computational terms, making them well-suited to episodic environments that require real-time performance. However, the process of engineering such architectures is not well understood. Third, *belief-desire-intention* architectures are discussed. In such architectures, decision making is viewed as *practical reasoning* from beliefs about how the world is and will be to the options available to an agent, and finally to intentions and actions. The process is somewhat similar to the kind of "folk reasoning" that humans use every day in deciding what to do. Belief-desire-intention architectures also have an attractive formalization, discussed elsewhere in this book. Fourth, layered agent architectures are reviewed. In such architectures, decision making is partitioned into a number of different decision making layers, each dealing with the agent's environment at a different level of abstraction. Layered agent architectures provide a natural way of decomposing agent functionality, and are currently a popular approach to agent design. In particular, the *horizontally layered* TOURINGMACHINES architecture and the *vertically layered* INTERRAP architecture are discussed. Finally, some prototypical agent programming languages are reviewed: Shoham's AGENT0 language, and Fisher's Concurrent METATEM language.

Chapter 2, "Multiagent Systems and Societies of Agents" by Michael N. Huhns and Larry M. Stephens – Agents operate and exist in some environment, which typically is both computational and physical. The environment might be open or closed, and it might or might not contain other agents. Although there are situations where an agent can operate usefully by itself, the increasing interconnection and networking of computers is making such situations rare. In Chapter 2, environments in which agents can operate effectively and interact with each other productively are analyzed, described, and designed.

The environments provide a computational infrastructure for such interactions to take place. The infrastructure includes communication protocols, which enable agents to exchange and understand messages, and interaction protocols, which enable agents to have conversations—structured exchanges of messages. For example, a communication protocol might specify that the messages for a particular course of action to be exchanged between two agents are of the types Propose, Accept, Reject, and Counterpropose. Based on these message types, two agents might use the following interaction protocol for negotiation: Agent1 proposes a course of action to Agent2; Agent2 evaluates the proposal and sends a counterproposal to Agent1; Agent1 accepts the counterproposal.

Interaction protocols enable agents to coordinate their activities, which can then be performed more efficiently. The degree of coordination is the extent to which they avoid extraneous activity by reducing resource contention, avoiding livelock and deadlock, and maintaining applicable safety conditions. Cooperation is coordination among nonantagonistic agents, while negotiation is coordination among competitive or simply self-interested agents. Chapter 2 describes protocols for coordination, cooperation, and negotiation.

Chapter 2 also shows how environments in which large numbers of agents exist must have different interaction protocols, based on social commitments, laws, and conventions.

Chapter 3, "Distributed Problem Solving and Planning" by Edmund H. Durfee – The interaction protocols introduced in Chapter 2 provide a means for agents to communicate about working together to solve problems, including coordination problems. Chapter 3 focuses on strategies for using protocols and reasoning capabilities to realize the benefits of cooperation. *Distributed problem solving* focuses on techniques for exploiting the distributed computational power and expertise in a MAS to accomplish large complex tasks. Of particular interest are strategies for moving tasks or results among agents to realize the benefits of cooperative problem solving. One main thread of work is the development of *task-passing techniques* to decide where to allocate subtasks to exploit the available capabilities of agents when large tasks initially arrive at a few agents. A second main thread of work is the study of *result-sharing strategies* to decide how agents that might be working on pieces of larger task can discover the relationships among their activities and coordinate them.

Coordinating problem-solving activities can involve anticipating the activities being undertaken by various agents and modifying those *plans* to make them more coordinated. Solving this planning problem is thus both a means to an end (distributed problem solving) and a distributed problem to be solved in its own right. The specific requirements and representations of planning problems, however, allow us to identify techniques that are particularly suited for *distributed planning*. We distinguish between the planning process and the execution of plans, and recognize that either, or both, of these can be distributed. We can then consider techniques for each. An interesting issue arises as to whether the coordination process should precede or succeed the planning processes of the agents; different decisions lead to different flavors of distributed planning, and a perspective is presented that allows these approaches to be seen as extremes of a more general process. It is also considered how throwing execution into the mix of planning and coordination can complicate matters, and algorithms for interleaving planning, coordination, and execution for dynamic applications are presented.

Chapter 4, "Search Algorithms for Agents" by Makoto Yokoo and Toru Ishida – This chapter deals with search algorithms for agents. Search is an umbrella term for various problem solving techniques in AI, where the sequence of actions required for solving a problem cannot be known *a priori* but must be determined by a trial-and-error exploration of alternatives. Search problems are divided into three classes: *(i)* path-finding problems, where the objective is to find a path from an initial state to a goal state, *(ii)* constraint satisfaction problems, where the objective is to find a combination of variable values that satisfies the given constraints, and *(iii)* two-player games such as chess and checkers. While two-player games deal with situations in which two *competitive* agents exist, most algorithms for the other two classes (constraint satisfaction and path-finding) were originally developed for single-agent problem solving. Various *asynchronous search* algorithms for these two classes are described. These algorithms are useful for cooperative problem solving by multiple agents each with *limited rationality*, since in these algorithms, a problem can be solved by accumulating local computations for each agent, and the execution order of these local computations can be arbitrary or highly flexible. More specifically, with respect constraint satisfaction problems the following asynchronous search algorithms are presented: the filtering algorithm, the hyper-resolution-based consistency algorithm, the asynchronous backtracking algorithm, and the asynchronous weak-commitment search algorithm. With respect to path-finding problems, first asynchronous dynamic programming as the basis for other algorithms is introduced. Then the Learning Real-time A* algorithm, the Real-time A* algorithm, the Moving Target Search algorithm, Real-time Bidirectional Search algorithms, and real-time multiagent search algorithms as special cases of asynchronous dynamic programming are described. With respect to two-player games, the basic minimax procedure and the alpha-beta pruning method to speed up the minimax procedure are presented.

Chapter 5, "Distributed Rational Decision Making" by Tuomas W. Sandholm – Multiagent systems consisting of self-interested agents are becoming increasingly important. One reason for this is the technology push of a growing standardized communication infrastructure over which separately designed agents belonging to different organizations can interact in an open environment in real-time and safely carry out transactions. The second reason is strong application pull for computer support for negotiation at the operative decision making level. For example, we are witnessing the advent of small transaction electronic commerce on the Internet for purchasing goods, information, and communication bandwidth. There is also an industrial trend toward virtual enterprises: dynamic alliances of small, agile enterprises which together can take advantage of economies of scale when available—e.g., respond to more diverse orders than individual agents can—but do not suffer from diseconomies of scale. Automated negotiation can save labor time of human negotiators, but in addition, other savings are possible because computational agents can be more effective at finding beneficial short-term contracts than humans are in strategically and combinatorially complex settings.

This chapter discusses methods for making socially desirable decisions among rational agents that only care of their own good, and may act insincerely to promote it. The techniques covered include

- voting,
- auctions,
- bargaining,
- market mechanisms,
- contracting, and
- coalition formation.

The chapter cites results from microeconomics—especially game theory—but it is not a general overview of those topics. Instead it deals relatively deeply with some of the topics which are particularly relevant to the design of computational multiagent systems. Special emphasis is placed on the implications of limited computation on the classic results. This is one area where game theory and computer science fruitfully blend within the field of DAI.

Chapter 6, "Learning in Multiagent Systems" by Sandip Sen and Gerhard Weiss – Multiagent systems typically are of considerable complexity with respect to both their structure and their function. For most application tasks, and even in environments that appear to be more or less simple at a first glance, it is extremely difficult or even impossible to correctly specify the behavioral repertoires and concrete activities of multiagent sytems at design time. This would require, for instance, that it is known in advance which environmental requirements will emerge in the future, which agents will be available at the time of emergence, and how the available agents have to interact in response to these requirements. Obviously, often the only feasible way to cope with this kind of problems is to endow the agents themselves with the ability to learn appropriate activity and interaction patterns. This chapter focuses on important aspects of learning in multiagent systems.

The chapter starts with a more general characterization of learning in multiagent systems. This includes an identification of *principle categories* of this kind of learning, an overview of *differencing features* that help to structure the broad variety of forms of learning that may occur in multiagent systems, and (from the point of view of multiagent systems) a description of the basic learning problem known as the *credit-assignment problem*. Then several typical learning approaches are described and illustrated. These approaches are ordered according to their main focus:

- learning and activity coordination;
- learning about and from other agents; and
- learning and communication.

The chapter also offers a brief guide to relevant related work from machine learning, psychology, and economics, and shows potential directions of future research.

Chapter 7, "Computational Organization Theory" by Kathleen M. Carley and Les Gasser – Chapter 7 provides an overview of the emergent field of Computational Organization Theory (COT). Researchers in COT use mathematical and computational models to theorize about and analyze organizations and the processes of organizing. Research in this area blends some of the traditional concerns of AI and distributed computing with work by organizational and social theorists, to develop a more comprehensive understanding. In most of this work, organizations are characterized as multiagent systems in which agents are embedded in particular social roles, have particular cognitive capabilities, and are engaged in specific organizationally-relevant tasks. Using computationally intensive techniques and empirical data, researchers are examining how organizations composed of people, artificial agents (such as webbots, robots, or other information technologies), or both, should be coordinated and how work should be distributed within and across such systems. Much of the work in this area focuses on issues of representation, organizational design, knowledge sharing, learning, and adaptivity. Some issues currently being addressed include:

- What is the nature of coordination and how can it be made most effective?
- How do organizations of people and organizations of automated agents differ? Should they be coordinated in similar ways?
- How socially intelligent do artifical agents need to be to communicate effectively with people during a team decision task?

and so on. In general, the aim of research in this area is to build new concepts, theories, and knowledge about organizing and organization in the abstract, to develop tools and procedures for the validation and analysis of computational organizational models, and to reflect these computational abstractions back to actual organizational practice through both tools and knowledge. This chapter reviews the dominant approaches and models in this area, potential toolkits, new findings, directions, and trends.

Chapter 8, "Formal Methods in DAI" by Munindar P. Singh, Anand S. Rao, and Michael P. Georgeff – As DAI moves into larger and more critical applications, it is becoming increasingly important to develop techniques to ensure that DAI systems behave appropriately. Safety and assurance can be addressed by development methodologies, as in traditional software engineering. But for methodologies to be effective in improving safety and correctness, they must be founded upon rigorous characterizations of the architecture and behavior of the given class of systems. In the case of DAI, this means that we develop formal bases for the abstractions and constructions that arise in the study of agents and multiagent systems.

Chapter 8 studies precisely such formalizations. It begins with background material on some logics that are commonly used in traditional computer science, especially in the verification of concurrent programs. It presents DAI-specific enhancements to these logics, covering the concepts of knowledge, beliefs, desires, goals, intentions, and know-how. Such cognitive concepts have long been informally studied in the context of agents, because they offer high-level specifications of the agents' design and behavior that are independent of most implementation details. In order to give a flavor of how the formal techniques might be applied, this chapter also describes how the above concepts may be realized in a practical interpreter.

Next, this chapter discusses a range of additional phenomena, such as coordination, teamwork, interagent communications, and social primitives. In conjunction with concepts such as joint and group intentions, which lift single-agent primitives to multiagent systems, these topics provide the essential conceptual basis for multiagent systems.

The chapter concludes with a discussion of tools and systems that either directly implement the associated DAI-specific formal theories, are inspired by those theories, or bring in traditional formal approaches.

Chapter 9, "Industrial and Practical Applications of DAI" by H. Van Dyke Parunak – Successful application of agents (as of any technology) must reconcile two perspectives. The researcher (exemplified in Chapters 1 to 8) focuses on a particular capability (e.g., communication, planning, learning), and seeks practical problems to demonstrate the usefulness of this capability (and justify further funding). The industrial practitioner has a practical problem to solve, and cares much more about the speed and cost-effectiveness of the solution than about its elegance or sophistication. Chapter 9 attempts to bridge these perspectives. To the agent researcher, it offers an overview of the kinds of problems that industrialists face, and some examples of agent technologies that have made their way into practical application. To the industrialist, it explains why agents are not just the latest technical fad, but a natural match to the characteristics of a broad class of real problems. Chapter 9 emphasizes agent applications in manufacturing and physical control because good examples are available, the problems of interfacing agents to the environment are more challenging than in all-electronic domains, and the evidence of success or failure is clearer when a system must directly confront the laws of physics. The chapter begins by describing the main *industrial motivations* for choosing an agent architecture for a particular problem. It then explains the concept of a *system life cycle*, which is widely used in industry to manage the progress of a project toward its intended results. The life cycle serves as an organizing framework for two sets of case studies. The first shows where in the life cycle agent-based systems are used, while the second discusses the design and construction of an agent-based system in terms of the life cycle. The chapter includes a review of some *development tools* that will hasten deployment of agent technology in industry.

Chapter 10, "Groupware and Computer Supported Cooperative Work" by Clarence Ellis and Jacques Wainer – The explosive growth of internet, intranet, and related technologies is leading to an explosive growth of the interest in groupware. Within our society, we see technologies that appear to greatly advance the conditions for human life (e.g., water purification technology), and others that seem to be questionable in their societal effects (e.g., television technology). Convergence of computer and communications technologies makes the world a "global village." Groupware is an emerging technology that promises to conceptually bring people together. Whether people are in the same conference room or scattered around the world, groupware can potentially help them to coordinate, collaborate, and cooperate.

Chapter 10 provides an introduction to groupware and computer supported cooperative work. Groupware is defined as computing and communications technology-based systems that assist groups of participants, and help to support a shared environment. Computer supported cooperative work is defined as the study of how groups work, and how technology to enhance group interaction and collaboration can be implemented.

The chapter, which primarily emphasizes technical issues of groupware, offers a taxonomy of groupware that is based on four aspects. The first aspect, *keeper*, groups functionalities that are related to storage and access of shared data; the second aspect, *coordinator*, is related to the ordering and synchronization of individual activities that make up the group process; the third aspect, *communicator*, groups functionalities related to unconstrained and explicit communication among the participants; and the fourth aspect, *team-agents*, refers to intelligent or semi-intelligent software components that perform specialized functions and contribute as participants to the dynamics of the group. Most current groupware systems have functionalities that are covered by the first three aspects. However, the most promising aspect is the fourth one—and because this aspect is most closely related to DAI, particular attention is paid to it throughout the chapter.

Chapter 11, "Distributed Models for Decision Support" by Jose Cuena and Sascha Ossowski – Decision support systems (DSS) assist the responsible persons in generating action plans in order to influence the behavior of natural or artificial systems in a desired direction. Knowledge-based DSSs have shown to perform well in a variety of different domains, as they allow for a meaningful dialogue with the control personnel. Still, the growing complexity of todays decision support problems makes the design process of such systems increasingly difficult and cost intensive.

This chapter introduces the notion of distributed knowledge-based DSSs. Setting out from concepts described in Part 1 of this book, an agent-based decision support architecture is proposed. On the basis of this architecture, the possibilities of a distributed, agent-based approach to DSS design are discussed by means of three case studies taken from literature:

- Environmental Emergency Management – The objective of Environmental Emergency Management is to minimize the negative impact of natural disasters or industrial accidents. The architecture of a multiagent DSS is presented, in which each agent corresponds to a preestablished organizational entity. An example of the operation of this system is given within the frame of a flood management scenario.
- Energy Management – Energy Management aims to maintain high quality supply of electrical energy despite damages to transport and distribution networks caused by wind, icing, lightning etc. A multiagent decision support architecture for this task is described, that integrates both preexisting and purposefully designed agents. In an example, it is shown how these agents cooperate to perform fault diagnosis and service restauration in a distributed fashion.
- Road Traffic Management – Road Traffic Management is concerned with the smooth flow of traffic in a road network along the different rush hour demands and despite events such as accidents or road works. A multiagent architecture is presented, where each traffic agent is responsible for specific parts of the road

network. An example illustrates how the interaction between these agents leads to the coordinated proposals of traffic control actions.

Chapter 12, "Concurrent Programming for DAI" by Gul A. Agha and Nadeem Jamali – As processors and networks have become faster and cheaper, parallelism and distribution to achieve performance gains has become more attractive. This chapter describes the Actor model of concurrent computation and extends it to define mobile *agents*. Mobile agents may travel over a network of processors in search for resources that they need to achieve their goals.

An economic model is useful as a basis on which hosts could be provided incentives to allow agents to migrate and also to limit the resources that the agents consume. The chapter defines agents that are allocated limited units of a global currency which they can expend on purchasing physical resources needed for carrying out their activities on different hosts.

Reasoning about concurrent systems has traditionally been a challenging task. The chapter discusses ways of modifying semantics of Actor systems to support mobility and control of resource consumption. The semantics of Agent systems provides guidelines for designing systems of agents, for supporting non-intrusive monitoring of the system, allows the systematic use computational reflection, and enables agents to develop proofs of safe execution which may be offered to prospective hosts.

The dynamicity and uncertainty in the behavior of ensembles of agents poses challenging problems. The chapter describes how the context in which agents execute, and in which their interactions are mediated, may be dynamically customized. Programming constructs for naming in open systems and scalable communication are also described. The chapter also includes a number of programming examples and a discussion of open issues.

Chapter 13, "Distributed Control Algorithms for AI" by Gerard Tel – This chapter discusses a number of elementary problems in distributed computing and a couple of well-known algorithmic "building blocks," which are used as procedures in distributed applications. The chapter is not intended to be complete, as an enumeration of the many known distributed algorithms would be pointless and endless. The chapter is even not intended to touch all relevant sub-areas and problems studied in distributed computing, because they are not all relevant to DAI. Rather than an algorithm catalogue, the chapter aims to be an eye-opener for the possibilities of the distributed computing model, an introduction to designing and reasoning about the algorithms, and a pointer to some literature.

The chapter introduces the distributed model and illustrates the various possibilities and difficulties with algorithms to compute spanning trees in a network. It is shown how the communication and time complexities of the algorithms are evaluated. Then a more complicated, but relevant control problem is studied, namely termination detection. This study reveals how intricate it is to make information about a distributed global state available to a node locally. Termination detection occurs in distributed applications of all areas and is not specific for DAI.

Application of some distributed control techniques is exemplified in the later sections in distributed computations for AI problems. A distributed implementation of Arc Consistency and Constraint Satisfaction is discussed, and it is shown how termination detection and distributed evaluation of functions play a role. The chapter finally presents a distributed graph algorithm, illustrating another termination detection principle, and providing an example of broadcast/convergecast and controller movement.

The Exercises

To enable the reader to gain practice in multiagent systems and DAI, a number of exercises of varying levels of difficulty are provided at the end of each chapter. The following rating system is applied to roughly indicate the amount of effort required for solving the exercises:

1. *[Level 1]* Exercises of Level 1 are solvable within a day (e.g., simple test of comprehension or a small program).
2. *[Level 2]* Solving exercises of Level 2 can take days or weeks (e.g., writting a fairly complex program). Usually the chapters provide all the information necessary for solving Level-1 and Level-2 exercises.
3. *[Level 3]* Exercises of Level 3 are even harder and their solution can take several weeks or months. Many of these exercises are related to "hot" topics of current research.
4. *[Level 4]* Exercises of Level 4 concern open research questions and could be topics of PhD theses. Solving Level-3 and Level-4 exercises typically requires to read further literature and/or to conduct extensive experiments.

It is recommend to do the Level-1 and Level-2 exercises, and to attack at least some of the exercises of Levels 3 and 4. Carefully working through Level-1 and Level-2 exercises will reward a reader with a real understanding of the material of the chapters, and solving Level-3 and Level-4 exercises will turn a reader into a real expert!

The Glossary

The glossary at the end of the book is the result of a joint effort of the chapter authors. It provides compact explanations of a number of terms used in the field of multiagent systems and DAI. This glossary is neither intended to be complete nor to offer "definitions" in the strict sense of this word. Instead, the focus is on key terms and on their common usage. The primary purpose of the glossary is to make it easier for the readers to get acquainted with basic terminology.

A Few Pointers to Further Readings

The number of publications on multiagent systems and DAI has grown rapidly in the past decade. The reader not familiar with the field and the available literature may find the following, by no means complete, list of pointers useful as an initial point of orientation:

- *Introductory texts, surveys, and overviews*:
  There are several general texts on multiagent systems and DAI (e.g., [2, 8, 20, 22, 25, 31, 40]), distributed problem solving (e.g., [10, 11, 17]), and agents (e.g., [5, 22, 45]).
- *Collections*:
  A detailed treatment of many key aspects of DAI is provided in [34]. A recent compendium that covers both agent and multiagent themes is [23]. A "classic" collection of DAI articles is [3]. Journal special issues on DAI and multiagent systems are, e.g., [9, 16, 46]. There is a number of proceedings of conferences and workshops on multiagent systems and DAI. For instance, the "International Conference on Multi-Agent Systems (ICMAS)" series resulted in three proceedings [12, 18, 30] that broadly cover the whole range of multiagent systems. The AAAI-sponsored "Workshop on DAI" series led to two other "classic" collections of DAI papers [19, 21]. The papers presented at the "European Workshop on Modelling Agents in a Multi-Agent World (MAAMAW)" series are published in [1, 7, 6, 13, 14, 35, 42, 43]. There are several conference and workshop series on agents. Among them are, for instance, the "International Conference on Autonomous Agents (Agents)" series [37, 41], the "International Workshop on Agent Theories, Architectures, and Languages (ATAL)" series [32, 39, 44, 47], and the "Cooperative Information Agents (CIA)" series [27, 28].
- *Bibliographies*:
  A useful list of pointers to published material on DAI and related areas is provided in [29]. A subject-indexed bibliography that comprehensively covers early DAI publications is [4].
- The first journal in the field is *Autonomous Agents and Multi-Agent Systems* (Kluwer Academic Publishers).

Pointers to papers that deal with specific aspects of multiagent systems are extensively included in the individual chapters.

References

1. M. Boman and W. Van der Velde, editors. *Decentralized Artificial Intelligence. Proceedings of the Eighth European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'97).* Lecture Notes in Artificial Intelligence, Vol. 1237. Springer-Verlag, Berlin, 1997.
2. A.H. Bond and L. Gasser. An analysis of problems and research in DAI. In A.H. Bond and L. Gasser, editors, *Readings in Distributed Artificial Intelligence*, pages 3–35. Morgan Kaufmann, San Mateo, CA, 1988.
3. A.H. Bond and L. Gasser, editors. *Readings in Distributed Artificial Intelligence.* Morgan Kaufmann, San Mateo, CA, 1988.
4. A.H. Bond and L. Gasser. A subject-indexed bibliography of distributed artificial intelligence. In A.H. Bond and L. Gasser, editors, *Readings in Distributed Artificial Intelligence*, pages 37–56. Morgan Kaufmann, San Mateo, CA, 1988.
5. J.M. Bradshaw. An introduction to software agents. In J.M. Bradshaw, editor, *Software Agents*, pages 3–46. AAAI Press/The MIT Press, 1997.
6. C. Castelfranchi and J.-P. Müller, editors. *Decentralized Artificial Intelligence. Proceedings of the Fifth European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'93).* Lecture Notes in Artificial Intelligence, Vol. 957. Springer-Verlag, Berlin, 1995.
7. C. Castelfranchi and E. Werner, editors. *Decentralized Artificial Intelligence. Proceedings of the Fourth European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'92).* Lecture Notes in Artificial Intelligence, Vol. 830. Springer-Verlag, Berlin, 1994.
8. B. Chaib-draa, B. Moulin, R. Mandiau, and P. Millot. Trends in distributed artificial intelligence. *Artificial Intelligence Review*, 6(1):35–66, 1992.
9. B. Chandrasekaran, editor. Special Issue on Distributed Artificial Intelligence of the *IEEE Transactions on Systems, Man, and Cybernetics.* Vol. SMC-11, 1981.
10. K.S. Decker. Distributed problem solving techniques: A survey. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-17:729–740, 1987.
11. K.S. Decker, E.H. Durfee, and V.R. Lesser. Evaluating research in cooperative distributed problem solving. In M.N. Huhns and L. Gasser, editors, *Distributed Artificial Intelligence, Volume 2*, pages 487–519. Pitman/Morgan Kaufmann, Cambridge, MA, 1989.
12. Y. Demazeau, editor. *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS-98).* IEEE Computer Society, 1998.
13. Y. Demazeau and J.-P. Müller, editors. *Decentralized Artificial Intelligence. Proceedings of the First European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'89).* North-Holland, 1990.
14. Y. Demazeau and J.-P. Müller, editors. *Decentralized Artificial Intelligence. Proceedings of the Second European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'90).* Elsevier Science, 1991.
15. E.H. Durfee. The distributed artificial intelligence melting pot. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-21(6):1301–1306, 1991.

16. E.H. Durfee, editor. Special Issue on Distributed Artificial Intelligence of the *IEEE Transactions on Systems, Man, and Cybernetics.* Vol. SMC-21, 1991.
17. E.H. Durfee, V.R. Lesser, and D.D. Corkill. Distributed problem solving. In S.C. Shapiro, editor, *Encyclopedia of Artificial Intelligence*, pages 379–388. John Wiley, 1992.
18. E.H. Durfee and M. Tokoro, editors. *Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS-96).* AAAI Press, 1996.
19. L. Gasser and M.N. Huhns, editors. *Distributed Artificial Intelligence, Volume 2.* Pitman/Morgan Kaufmann, 1989.
20. L. Gasser and M.N. Huhns. Themes in distributed artificial intelligence research. In L. Gasser and M.N. Huhns, editors, *Distributed Artificial Intelligence, Volume 2*, pages vii–xv. Pitman/Morgan Kaufmann, 1989.
21. M.N. Huhns, editor. *Distributed Artificial Intelligence.* Pitman/Morgan Kaufmann, 1987.
22. M.N. Huhns and M.P. Singh. Agents and multiagent systems: Themes, approaches, and challenges. In M.N. Huhns and M.P. Singh, editors, *Readings in Agents*, pages 1–23. Morgan Kaufmann, San Francisco, CA, 1998.
23. M.N. Huhns and M.P. Singh, editors. *Readings in Agents.* Morgan Kaufmann, San Francisco, CA, 1998.
24. N.R. Jennings, editor. *Cooperation in Industrial Multi-Agent Systems.* World Scientific, Singapore, 1994.
25. N.R. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1:7–38, 1998.
26. N.R. Jennings and M.J. Wooldridge, editors. *Agent Technology. Foundations, Applications, and Markets.* Springer-Verlag, Berlin, 1998.
27. P. Kandzia and M. Klusch, editors. *Cooperative Information Agents.* Lecture Notes in Artificial in Artificial Intelligence, Vol. 1202. Springer-Verlag, Berlin, 1997.
28. M. Klusch and G. Weiß, editors. *Cooperative Information Agents II.* Lecture Notes in Artificial in Artificial Intelligence, Vol. 1435. Springer-Verlag, Berlin, 1998.
29. D. Kwek and S. Kalenka. Distributed artificial intelligence references and resources. In G.M.P. O'Hare and N.R. Jennings, editors, *Foundations of Distributed Artificial Intelligence*, pages 557–572. John Wiley & Sons Inc., New York, 1996.
30. V.R. Lesser and L. Gasser, editors. *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95).* AAAI Press/The MIT Press, 1995.
31. B. Moulin and B. Chaib-Draa. An overview of distributed artificial intelligence. In G.M.P. O'Hare and N.R. Jennings, editors, *Foundations of Distributed Artificial Intelligence*, pages 3–55. John Wiley & Sons Inc., New York, 1996.
32. J.P. Müller, M. Wooldridge, and N.R. Jennings, editors. *Intelligent Agents III.* Lecture Notes in Artificial in Artificial Intelligence, Vol. 1193. Springer-Verlag, Berlin, 1997.
33. N.J. Nilsson. *Artificial Intelligence. A New Synthesis.* Morgan Kaufmann Publ., San Francisco, CA, 1998.
34. G.M.P. O'Hare and N.R. Jennings, editors. *Foundations of Distributed Artificial Intelligence.* John Wiley & Sons Inc., New York, 1996.
35. J.W. Perram and J.-P. Müller, editors. *Decentralized Artificial Intelligence. Proceedings of the Sixth European Workshop on Modelling Autonomous Agents in a*

Multi-Agent World (MAAMAW'94). Lecture Notes in Artificial Intelligence, Vol. 1069. Springer-Verlag, Berlin, 1996.

36. D. Poole, A. Machworth, and R. Goebel. *Computational Intelligence.* Oxford University Press, New York, 1998.
37. *Proceedings of the First International Conference on Autonomous Agents (Agents'97).* http://www.isi.edu/isd/Agents97/materials-order-form.html, 1997.
38. S.J. Russell and P. Norwig. *Artificial Intelligence. A Modern Approach.* Prentice Hall, Englewood Cliffs, New Jersey, 1995.
39. M.P. Singh, A. Rao, and M.J. Wooldridge, editors. *Intelligent Agents IV.* Lecture Notes in Artificial in Artificial Intelligence, Vol. 1365. Springer-Verlag, Berlin, 1998.
40. K. Sycara. Multiagent systems. *AI Magazine*, Summer:79–92, 1998.
41. K.P. Sycara and M. Wooldridge, editors. *Proceedings of the Second International Conference on Autonomous Agents (Agents'98).* Association for Computing Machinery, Inc. (ACM), 1998.
42. W. Van der Velde and J.W. Perram, editors. *Decentralized Artificial Intelligence. Proceedings of the Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'96).* Lecture Notes in Artificial Intelligence, Vol. 1038. Springer-Verlag, Berlin, 1996.
43. E. Werner and Y. Demazeau, editors. *Decentralized Artificial Intelligence. Proceedings of the Third European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'91).* Elsevier Science, 1992.
44. M. Wooldridge and N.R. Jennings, editors. *Intelligent Agents.* Lecture Notes in Artificial in Artificial Intelligence, Vol. 890. Springer-Verlag, Berlin, 1995.
45. M. Wooldridge and N.R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.
46. M. Wooldridge and N.R. Jennings, editors. Special Issue on Intelligent Agents and Multi-Agent Systems *Applied Artificial Intelligence Journal.* Vol. 9(4), 1995 and Vol. 10(1), 1996.
47. M. Wooldridge, J.P. Müller, and M. Tambe, editors. *Intelligent Agents II.* Lecture Notes in Artificial in Artificial Intelligence, Vol. 1037. Springer-Verlag, Berlin, 1996.