Cooperative Multiagent Systems: A Personal View of the State of the Art

Victor R. Lesser, *Member, IEEE*

Computer Science Department
University of Massachusetts at Amherst
V (413) 545-1322
F (413) 545-1249
Lesser@cs.umass.edu

Abstract—Scientific research and practice in multiagent systems focuses on constructing computational frameworks, principles, and models for how both small and large societies of intelligent, semi-autonomous agents can interact effectively to achieve their goals. This article provides a personal view of the key application areas for cooperative multiagent systems, the major intellectual problems in building such systems, the underlying principles governing their design, and the major directions and challenges for future developments in this field.

Index Terms—Multiagent systems, coordination, cooperation, distributed problem solving, distributed artificial intelligence, computational organizations.

To appear in *IEEE Trans. Knowledge and Data Engineering*, Jan.-Feb. Special Issue.

1. Introduction

Multiagent systems are computational systems in which two or more agents interact or work together to perform some set of tasks or to satisfy some set of goals. These systems may be comprised of homogeneous or heterogeneous agents. An agent in the system is considered a locus of problem-solving activity, it operates asynchronously with respect to other agents, and it has a certain level of autonomy. Agent autonomy relates to an agent's ability to make its own decisions about what activities to do, when to do them, what type of information should be communicated and to whom, and how to assimilate the information received. Autonomy can be limited by policies built into the agent by its designer, or as a result of an agent organization dynamically coming to an agreement that specific agents should take on certain roles or adopt certain policies for some specified period. Closely associated with agent autonomy is agent adaptability — the more autonomy an agent possesses the more adaptable it is to the emerging problem solving and network context. The degree of autonomy and the range of adaptability are usually associated with the level of intelligence/sophistication that an agent possesses.

Agents may also be characterized by whether they are benevolent (cooperative) or self-interested. Cooperative agents work toward achieving some common goals, whereas self-interested agents have distinct goals but may still interact to advance their own goals. In the latter case, self-interested agents may, by exchanging favors or currency, coordinate with other agents in order to get those agents to perform activities that assist in the achievement of their own objectives. For example, in a manufacturing setting where agents are responsible for scheduling different aspects of the manufacturing process, agents in the same manufacturing company would behave in a cooperative way while agents representing two separate companies where one company was outsourcing part of its manufacturing process to the other company would behave in a self-interested way.

Scientific research and practice in multiagent systems, which in the past has been called Distributed Artificial Intelligence (DAI), focuses on the development of computational principles and models for constructing, describing, implementing and analyzing the patterns of interaction and coordination in both large and small agent societies. Multiagent systems research brings together a diverse set of research disciplines and thus there is a wide range of ideas currently being explored. It is impossible to adequately address the full spectrum of issues and research perspectives of the field in such a short article.[1] Therefore, this article will of necessity be biased by my own research experience that has concentrated mainly on issues involved in cooperative interaction among sophisticated agents.

In the remainder of the article, I provide a personal view of some of the underlying principles governing the design of such multiagent systems, and the major directions and challenges of the field. The following three sections set the context for these discussions. The first section details the major application areas for multiagent systems and the potential benefits of structuring an application as a multiagent system. In the second section, a model of subproblem interaction is presented as the basis for cooperative interaction among agents. As part of this section, a number of examples of different types of subproblem interaction from implemented systems are analyzed. These examples are intended to motivate the need for coordination to effectively manage the cooperation necessary to solve interacting subproblems. Finally, the need for sophisticated quantitative-based coordination strategies to support effective cooperation among complex agents operating in open environments is discussed.

---

[1] A more comprehensive and historical view of the field can be obtained from the following special issues of journals [56, 61, 62], proceedings of the main conference of the field [63, 64, 73], collected sets of articles [57, 58, 65], recent books [44, 60], and a new journal *Autonomous Agents and Multiagent Systems*, Kluwer Academic Publishers.

2. Application of Multiagent Systems

Multiagent systems over the last few years have come to be perceived as crucial technology not only for effectively exploiting the increasing availability of diverse, heterogeneous, and distributed on-line information sources, but also as a framework for building large, complex, and robust distributed information processing systems which exploit the efficiencies of organized behavior. Multiagent systems also provide a powerful model for computing in the twenty-first century, in which networks of interacting, real-time, intelligent agents seamlessly integrate the work of people and machines, and dynamically adapt their problem solving to effectively deal with changing usage patterns, resource configurations and available sources of expertise and information. Application domains in which multiagent system technology is appropriate typically have a naturally spatial, functional or temporal decomposition of knowledge and expertise. By structuring such applications as a multiagent system rather than as a single agent, the system will have the following advantages: speed-up due to concurrent processing; less communication bandwidth requirements because processing is located nearer the source of information; more reliability because of the lack of a single point of failure; improved responsiveness due to processing, sensing and effecting being co-located; and finally, easier system development due to modularity coming from the decomposition into semi-autonomous agents.

Examples of application domains that have used a multiagent approach include:

- *Distributed situation assessment* which emphasizes how (diagnostic) agents with different spheres of awareness and control (network segments) should share their local interpretations to arrive at consistent and comprehensive explanations and responses (e.g., network diagnosis [50], information gathering on the Internet [11, 39], distributed sensor networks [3, 36]);

- *Distributed resource scheduling and planning* which emphasizes how (scheduling) agents (associated with each work cell) should coordinate their schedules to avoid and resolve conflicts over resources, and to maximize system output (e.g., factory scheduling [38, 41, 51], network management [1], and intelligent environments [74, 75]);

- *Distributed expert systems* which emphasize how agents share information and negotiate over collective solutions (designs) given their different expertise and solution criteria (e.g., concurrent engineering [32], network service restoral [6, 30]).

The next generation of applications will integrate characteristics of each of these generic domains. The need for a multiagent approach can also come from applications in which agents represent the interests of different organizational entities (e.g., electronic commerce [40] and enterprise integration [2]). Other emerging uses of multiagent systems are in layered systems architectures in which agents at different layers need to coordinate their decisions (e.g., to achieve appropriate configurations of resources and computational processing [53]), and in the design of survivable systems in which agents dynamically reorganize to respond to changes in resource availability, software and hardware malfunction, and intrusions. In general, multiagent systems provide a framework in which both the inherent distribution of processing and information in an application and the complexities that come from issues of scale can be handled in a natural way.

An example of this next-generation application is the WARREN system based on the RETSINA architecture [8, 9]. This multiagent system, which can be considered a multi-user, distributed information gathering system, (see Fig. 1) assists with the management of financial portfolios. Many of the features of the portfolio management domain are likely to become more common in the future: (1) an enormous amount of available information that is changing, unorganized, overlapping and possibly contradictory (e.g., market data, financial report data, technical models, analysts' reports, and breaking news), (2) a wide variety of analyses, each implemented as a separate agent by different designers, that can and should be brought to bear on the task, (3) analyses that can differ widely in their resource requirements, quality of results, and speed, (4) many sources of uncertainty and dynamic change in the environment, (5) time pressures that present agents with real-time deadlines for certain tasks, and (6) resource and cost constraints — since not all data and processing is available for free. Efficient performance of this information processing task requires dynamically locating appropriate expertise and information sources, high-level planning of how to decompose the overall task based on both user objectives and resource/agent availability, protocols for agents to come to consensus when they have conflicting viewpoints, careful scheduling of local activities and their interaction with activities of other agents so as to achieve coherent inter-agent behavior, and execution monitoring/adaptation of agent activities to guarantee that the overall task is accomplished in a cost-effective manner given the evolving state of network problem-solving and resources.

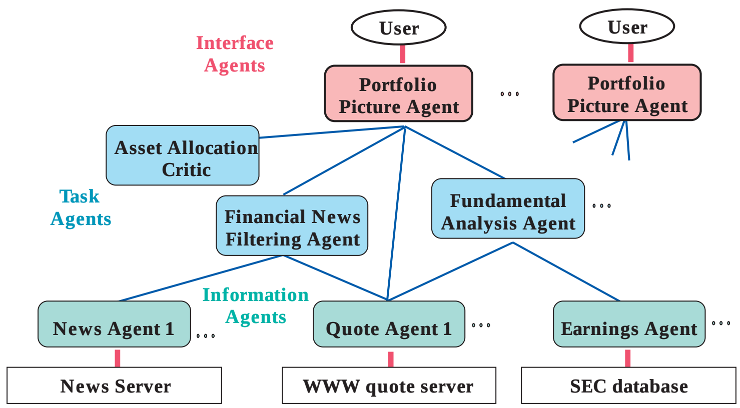

Figure 1: The WARREN System as Represented in RETSINA

As exemplified by the requirements laid out above for a WARREN-type application, it is the need to be able to adapt intra-agent and inter-agent problem solving to the dynamics of the environment in both short- and long-term ways that will differentiate these types of agent-based systems from more conventional distributed systems architectures where adaptability, especially at the domain problem-solving level, is not a primary motivation.

3. The Nature of Multiagent Interaction

One of the key problems in cooperative multiagent systems is how to get agents to cooperate effectively. The need to interact in such systems occurs because agents solve subproblems that are interdependent, either through contention for resources or through relationships among the subproblems. These relationships arise from two basic situations related to the natural decomposition of domain problem solving into subproblems. The first situation is where the subproblems are the same or overlapping, but different agents have either alternative methods or data that can be used to generate a solution. For example in a distributed situation assessment application, overlapping subproblems occur when different agents are interpreting data from different sensors (independent information sources) that have overlapping sensor regions (cover similar information) [3]. Another form of interdependence occurs when two subproblems are part of a larger problem in which a solution to the larger problem requires that certain constraints exist among the solutions to its subproblems. For example, in a distributed expert system application involving the design of an artifact where each agent is responsible for the design of a different component (subproblem), there are constraints among these subproblems that must be adhered to if the individual component designs will mesh together into an acceptable overall design [32]. We include in this latter case the simple situation where the results of one subproblem are needed to solve another. Additional interdependencies among subproblems, not inherent in the problem domain, arise when it is not possible to decompose the problem into a set of subproblems such that there is a perfect fit between the computational requirements for effectively solving each subproblem and the location of information, expertise, processing, and communication resources in the agent network [33, 34]. This lack of a perfect fit often leads to a situation where there may be insufficient local information or resources for an agent to completely or accurately solve its assigned subproblems through its own processing. Further, resource contention issues in multiagent systems do not entail simply reasoning about exclusive access to a shared resource, but may involve more subtle issues such as what percentage of a resource's capacity an agent will use (e.g., a communication channel), the creation of shared agent plans so that the use of a scarce resource can satisfy multiple objectives [52], or the reconfiguration of resources to better meet the competing needs of agents [53].

Depending upon the character of subproblem interdependencies, the interactions among agents in a multiagent system can be complex, often requiring a multi-step dialogue similar to an asynchronous co-routine type of interaction. For example, it may be impossible for one agent to completely solve subproblem p_j without another agent first partially solving subproblem p_i, or solving p_i may simply make it easier to solve p_j, or knowing the solution to p_i may obviate the need to solve p_j. These types of interactions are exemplified in a recently fielded commercial multiagent system for service restoral of an electricity transportation grid involving agents for fault detection, fault isolation and diagnosis, and network reconfiguration [6]. Consider the example of two expert agents in this system performing different forms of fault diagnosis, i.e., overlapping subproblems whose solutions need to be consistent. Each of these agents, operating concurrently, uses very different algorithms to do their diagnosis and the information they use is not identical. Both can make mistakes but generally will not make the same mistake. They interact by exchanging partial results to focus their local diagnostic search processes towards promising areas of the transportation grid where the fault likely originated, and away from unpromising ones. They also exchange final results to increase the confidence in the eventual diagnosis that they agree to; if they disagree then a more complicated interaction is warranted (i.e., negotiation) in order to understand the basis of the disagreement and to subsequently reach a different diagnosis based on this resolution of conflicting viewpoints. Thus, by working together, they not only produce a solution of higher quality, but will often accomplish the task quicker as well.

These types of agent interactions can lead to the need for coordination decisions by agents about which tentative diagnosis hypotheses to communicate and how reliable and precise these hypotheses need to be before they are sent to another agent. Choices also need to be made about the areas on which to focus diagnosis, based on the information needs of other agents. Though not emphasized in this example, there can be additional needs for coordination in order to: recognize when other agents are working on interacting subproblems; identify which other agents can and should solve a specific subproblem; decide if, how and when to solve a specific subproblem based on local and non-local criterion.

A more detailed example of the issues involved in the sharing of partial results involves a distributed interpretation application [3]. In this application, each agent is sensing and interpreting data from overlapping acoustic sensor regions in order to track vehicle movements in the sensed environment (see Figure 2). This example shows that a complete and accurate track map could not be created from agent A and B's independently developed solutions. Major adjustments of individual interpretations are required. Agent A can use the information from agent B to recognize that its interpretation of the data associated with track G_2 was faulty. This data, instead of being a representation of an actual vehicle moving in the environment at those locations, was in reality environmental reflections of a vehicle moving in a different region, i.e., a ghost track. Agent A can also use agent B's portion of track T_4 as predictive information, allowing agent A to make assumptions about its sensor having missed signals at times 4 and 5 that could complete track T_4. Further, agent A can produce an acceptable interpretation for the remainder of its original ghost track (times 4 through 7 data), based on communication with agent B to confirm most of this data (times 5 through 7 in the overlapping region) as ghost data and can provide a source (T_4) for the G_2 ghost track. Agent B's uncertainty over its interpretations (the time 5 through 10 portion of track T_4) because of the limited number of points over which it is able to track the vehicle can be decreased due to agent A's ability to find a continuation of the track in its area. This cooperative adjustment process requires back-and-forth communication between the agents rather than simply having one agent's "better" solutions override the others. As in the first example, the coordination decisions about when and what hypotheses (and what level of detail) to transmit, what hypotheses to work on, and how much effort to put into the development of specific hypotheses can have dramatic impact on how fast this resolution process converges.

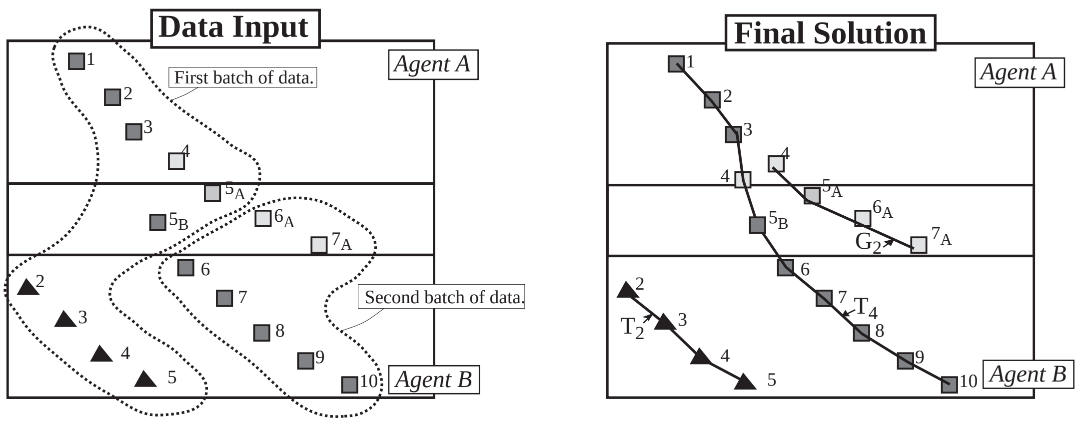

Figure 2: Example of a Two-agent Distributed Aircraft Monitoring Scenario[2]

A final example of the need for cooperative interaction in order to solve interdependent subproblems (in this case subproblems related through the use of similar resources) is distributed airport scheduling [66]. In this application, each arriving plane needs to be assigned not only a gate at which to land but also baggage handlers/trucks to unload and load baggage, and equipment and personnel for servicing the plane (fueling, cleaning, etc.). If we assume that different concourses of an airport are scheduled by different scheduling agents with their own complement of resources, the need for cooperation (the lending of resources) occurs when there are insufficient resources assigned to a particular concourse given the set of flights that need to be handled during a specified period. This lack of sufficient resources will then delay the scheduled landing and departure times for planes. In this overload situation, other concourse agents could potentially lend resources (assuming cooperative agents) if they have appropriate resources available during the period of overload. An example of one of the coordination problems in this domain is how to decide which

---

[2]The left-hand figure is the acoustic input to the two agents and the right-hand figure is the final interpretation of the agents. Agent A's sensor region is the upper and middle regions of the figure and agent B's sensor region is the middle and lower regions. The agents independently sense events occurring in the middle region. The data point symbols represent the positions of groups of acoustic signals detected by the sensors. The numbers associated with the data points give the times that these signals were generated and the subscripts indicate the agent receiving the data. Data points include the position of the signal source and the frequency class of the signal. The shading of each box indicates the loudness of the sounds being sensed (the darker the shading the louder) and is an indication of the likelihood of the sensory data being correct. Box 4 of track T_4, which only appears in the final solution, is not directly supported by acoustic data (only high-level predictions) and thus is not shaded. T_4 is a vehicle track and G_2 is a ghost track caused by the environmental reflections of sounds from T_4.

agent(s) should be asked to lend resources and how much this agent should disrupt the scheduling of its own concourse's activities in order to accommodate the needs of another. An interesting aspect of this problem is that when there is no coordination among agents it is often the case that when an overloaded agent asks for help it will be costly and disruptive to the lending agents' concourse schedule to revise it to make the needed resources available. However, if the scheduling agents can communicate about their anticipated resource needs for particular time periods before the detailed allocation of resources has been made, then this meta-level information can be extremely helpful in allowing underloaded agents to anticipate specific resource shortages. This anticipation allows them to not prematurely commit those resources that may be needed by other agents until they understand the exact nature of the requests by overloaded agents. In this way, they will maintain as much scheduling flexibility with respect to the overloaded resource as is consistent with their own needs; thus, they will be able to respond to a borrowing request with less disruption to their own concourse schedule. By implicitly coordinating through the exchange of meta-level information, the agents' solutions to their own local scheduling problems will be more optimal from a global perspective.

In summary, the need for interaction among agents to efficiently solve their interacting subproblems may require agents to closely coordinate their activities during problem solving. This coordination is based on reasoning about the nature of subproblem interdependencies, the agents' current state of problem solving, and the status of network resources. Inappropriate or lack of any coordination can contribute to groups of agents generating solutions that are sub-optimal, wasting both computational and communication resources due to generating and communicating unneeded, redundant or poorly timed results, and in the most serious case, failing to generate an overall solution because of the outright failure to generate key results.

4. Quantitative/Statistical Perspective on Coordination

Coordination strategies enable groups of agents to solve problems effectively through decisions about which agents should perform specific tasks and when, and to whom they should communicate the results. The potential complexity involved in making these decisions can be seen in the simple situation where one agent needs the results of a subproblem that another agent is solving. If it can be arranged that the producing agent will deliver the desired result in a timely fashion so that the consuming agent does not have to idle waiting for the results, then system performance is improved. On the surface this coordination decision is simple. However, suppose that the producing agent has other tasks to do with their own deadlines in addition to producing a result for the other agent. To further complicate this decision process, the agent may have alternative methods for doing those tasks that trade off the quality of the task solution against the time to complete the task. Similarly, the consuming agent may also have flexibility about when it does its tasks because there are other tasks it is also working on, and it may also be able to make trade-offs in how it accomplishes its tasks. Additional complexity is introduced when neither the time that a method takes nor the quality of its results are known precisely but rather can be described by a statistical distribution. Meta-level reasoning may also be involved in the coordination decision process when there are alternative coordination strategies that can be used in the current situation. In this case, alternative coordination strategies must be analyzed in terms of their computational and communication resource requirements, the end-to-end delay in reaching a decision, the optimality of the coordination achieved, the potential gains achieved as a result of more effective coordination, and the current need for resources by other activities and their relative priority.

This coordination decision process can be further complicated when the information an agent is using to make its decisions is incomplete, out-of-date or inconsistent with that of other agents. Obtaining all the appropriate non-local information is often not practical due to:

- Limited communication bandwidth and computational capabilities which make it infeasible to package, transfer, and assimilate all pertinent information in a timely manner;

- The heterogeneity of agents which makes it difficult to share information; and the potential for competitive agents who, out of self-interest, are not willing to share certain information;

- The dynamic character of the environment due to changing problems, agents, and resources and the inability to predict with certainty the outcome of agents' actions.

Thus, making effective coordination decisions that appropriately deal with the uncertainty of information combined together with the complex set of factors that need to be taken into account is potentially a complex process.

It is my strong feeling that in order to design efficient and effective coordination strategies that will work in a wide variety of environments they must explicitly account for the benefits and the costs of coordination in the current situation in a quantifiable way [69]. The current situation includes the goals (and their importance or value) that the agent is currently pursuing and likely to pursue in the near term, the performance characteristics of the methods available to the agent for achieving its goals, the requirements these goals/methods impose on other agents, the requirements that the goals/methods of other agents impose on this agent, the state of network resources, and domain constraints on agent activities. Another way of saying this is that making coordination decisions is a complex, multi-level optimization problem based on how coordination actions (usually involving a statistical perspective) contribute to high-level system tasks meeting their performance objectives, and the relative importance of each of these tasks.

This emphasis on a quantitative/statistical perspective on coordination should not distract from the importance of mechanisms, protocols and formal frameworks [2, 6, 7, 16, 20, 21, 29, 43, 71] that: establish which tasks are important to accomplish and which agents/resources are capable of accomplishing them; determine how to decompose tasks into subtasks and provide sequencing constraints among these tasks; decide what information to transmit upon completion of a task and to whom; and define how to react to unexpected events in terms of what needs to be communicated to whom and what further actions need to be taken. These intended activities are in response to the various objectives (desires) of the agent, such as the local processing goals it is pursuing and the various requests by other agents for its assistance.

In essence, what is being suggested is that there must be a quantitatively oriented mechanism [77] at the lowest control layer to arbitrate among activities generated by higher, non-quantitative layers. This is especially true in situations where there are complex subproblem interdependencies among agents, where there are time pressures and resource bounds which preclude all goals of the system being solved in an optimal manner, where there are many choices available about how to solve a goal, and where the goals being solved and the agents/resources available to solve them are changing over time.

5. Key Principles Used in Building Multiagent Systems

The ubiquitousness of uncertain and incomplete information and the computational complexity of making optimal coordination decisions leads to a number of principles that are useful for structuring multiagent systems.

The first principle relates to the need to view the performance of a multiagent system in terms of a complex set of criteria in which there is rarely a way to optimize all criteria simultaneously. This principle, usually called "satisficing" behavior [35, 47, 48], was developed as a way of explaining how large organizations function. It underlies most of the other principles to be discussed in this section. For most real-world multiagent applications, the design goal of producing an "optimal" answer with minimal use of communication and processing resources, while at the same time being able to respond gracefully to a dynamically changing environment, is unrealistic; this is due to the communication and computational costs and delays that would be necessitated in acquiring the information necessary to make these "optimal" decisions. Instead, "satisficing" criteria for successful performance are adopted based on using a "reasonable" amount of resources to reduce uncertainty "sufficiently" so that it is "likely" that an "acceptable" answer will be achieved [33]. This emphasis on satisficing behavior also subtly moves the focus from the performance of individual agents to the properties and character of the aggregate behavior of agents.

An associated corollary is, given the rich set of criteria that can be used to define satisficing behavior and the tremendous diversity of environments and tasks, there is no single approach to organizing agent behavior that will be right for all situations. It has been shown that even in a relatively simple environment where there is a lot of variance in the performance characteristics of tasks, a single coordination strategy is not optimal over the range of task characteristics. In this situation, dynamically choosing a strategy at run time based on the known characteristics of the tasks leads to better performance than using any one fixed strategy [37]. It is my conjecture that in the future agents will be required to perform some form of meta-level reasoning so as to balance the level of optimality of their control decisions with the level of resources required to make the decisions, based on the characteristics of their tasks and the environment [9, 12, 15, 34, 76].[3]

The second principle relates to the need for flexibility in agent problem solving. Agent flexibility with respect to the availability, completeness and accuracy of its information and the availability and capabilities of external resources is often a key aspect of a multiagent system design. It enables agents to react dynamically to the emerging state of the group problem-solving effort. In other words, hard-coded assumptions about the character and availability of information and resources are typically avoided. In general, agent problem-solving architectures that deal explicitly with the uncertainty of information and the incompleteness of their local data bases are more adaptable for use in a multiagent context [25, 33, 34]. This flexibility can be equated with agent autonomy. One way this can be accomplished is through a sophisticated domain problem-solving architecture that can respond opportunistically to emerging conditions. Another way of achieving flexibility is for agents to have alternative methods available for solving subproblems that have varying information and resource requirements. The agent then pieces together at run time a set of methods that will be appropriate for the given situation. These are end points on a spectrum and, obviously, combinations of these approaches are possible. Additionally, flexibility can come from increasing the scope of activities that an agent is involved in so that the agent may pursue multiple goals. Thus, the agent can change focus when information or resources are not currently available to pursue a specific goal. This flexibility can come at the cost of increased reasoning about the nature of the problem-solving system itself, resulting in less computational time directed toward actual problem solving and more directed toward coordinating effectively with other agents. It can be expected that, for agents in some of the more advanced multiagent applications that are beginning to emerge, this coordination reasoning could be quite complex and time consuming.

Sophisticated control of local domain problem solving is also necessary in many cases for effective agent interaction. Agents need to explicitly reason about the intermediate states of their computation (in terms of what actions they expect to take in the near term, what information from other agents would be valuable for making further progress in their local problem solving, etc.). An aspect of

[3]A key issue is how to make this meta-level reasoning sufficiently powerful, yet not a significant computational cost in its own right.

this reasoning can involve an explicit representation of the uncertainty and incompleteness of its current problem-solving state. By having this representation, an agent can make more informed decisions about the value of obtaining specific information or doing further work locally to resolve its uncertainty [3][4]. Agents also need to be able to acquire, represent and reason about beliefs concerning the state of other agents, and to use assumptions about the rationality of other agents' problem solving in their reasoning. It has been shown that even the exchange of a coarse description of other agents' states (meta-level information) can be used to make effective coordination decisions [12]. However, if agents have very good models of the behavior of other agents, it may not be necessary to coordinate through the exchange of meta-level information but, rather, just the observation by one agent of the external actions of another agent may be sufficient [26].

The third principle relates to the need to exploit the efficiencies of organized behavior in coordinating large agent societies. Organizing the agents in terms of roles and responsibilities can significantly decrease the computational burden on coordinating their activities since there are fewer options and constraints that need to be evaluated in order to make appropriate coordination decisions. However, these assignments (long-term commitments) should not be so strict that an agent does not have sufficient latitude to respond to unexpected circumstances, nor should they be necessarily fixed for the duration of problem solving. Organizational control should be thought of as modulating (circumscribing) local control rather than dictating [8]. Implicit in this discussion are the concepts of commitment and intention. The ability to appropriately bound the intentions of agents, and to create and sufficiently guarantee the commitments of agents to accomplish certain tasks is at the heart of efficient organized behavior. These concepts, either implicitly or explicitly represented, are important keys to not only understanding but also implementing complex organized agent behavior in both small and large agent societies [5, 15, 19, 21, 29, 31].

6. Major Challenges and Research Directions

The field faces many challenges, some pragmatic and others deeply theoretical. A pragmatic one that seems very important and pressing is the development of an appropriate high-level software infrastructure/ framework to support the building of multiagent systems. At this point, the programming overhead to create a non-trivial multiagent system is still high and, thus, the number of fielded commercial applications is small. The development of such a framework is timely because of the emerging software infrastructure and standards being developed for mobile computing and interoperability among programs residing at distant sites (e.g., Java) which will simplify the construction of agents. However, this work will only partially solve the problems of building multiagent systems since it does not deal with high-level coordination issues. There are two possible approaches to building these higher level capabilities. One takes a language-oriented view, providing a set of operations and associated protocols for locating and communicating with agents such as KQML [16]. The decision about when to use the protocol, what information to transmit, etc., is left to the agent programmer. An alternative approach is a high-level framework where, once an agent has described its needs and capabilities for interacting with other agents in a domain-independent way, the framework will automatically make all the coordination decisions [10]. Again, these are extreme points on a continuum of possible approaches [6] to creating a software framework to ease the burden of implementing multiagent coordination strategies. Another continuum in terms of choices for coordination frameworks is centered on the following question: When is an end-to-end planning viewpoint (in terms of how a specific choice explicitly

[4]Resolution of all uncertainty may not be necessary for meeting the criteria of "satisficing" performance. In general, problem-solving architectures that deal explicitly with the uncertainty of information and the incompleteness of their local data bases are more adaptable for use in a multiagent context [25, 33, 34].

contributes to achieving the global objective) warranted for making an effective coordination decision, versus a reactive and local view of the effects?

Along these lines, one of the important developments in the field over the last few years has been the development of high-level coordination frameworks inspired by logic-based approaches to explicitly modeling cooperative interaction among small human teams. The computational viability of these agent coordination frameworks is often accomplished by limiting their model semantics so that they can be implemented via procedural reasoning. Examples include the joint intentions framework [7, 43, 70], the SharedPlan model [20, 21], joint responsibility [6, 67, 68], and hybrid models such as [71]. There is also recent work, in this case motivated by modeling of complex software processes, on an agent coordination language that also shows much promise for being able to specify and implement complex agent interaction patterns [72].

Multiagent research has long been divided into two camps, one concerned with cooperative (benevolent) agents and the other concerned with self-interested agents [14]. There has been very little cross-fertilization of ideas between these camps. Research on self-interested agents is often based on classical game theory with its assumptions of common knowledge among agents and complete rationality of agent reasoning. This is in contrast with the research on cooperative agents which makes no such assumptions; rather, it has generally been based on heuristic approaches having their roots in knowledge-based AI search, planning and scheduling mechanisms. However, researchers studying self-interested agents have recently begun to realize, as the class of problems being solved by their agents have become more complex, that these assumptions are not always reasonable [23, 45]. It is interesting to speculate whether there is more in common among cooperative and self-interested coordination mechanisms than currently believed — especially as the environments within which these mechanisms operate become more complex in terms of the computational difficulty of taking all appropriate factors into consideration and the increased level of information uncertainty and incompleteness [78]. An obvious challenge is how to construct agent societies consisting of a mixture of self-interested and cooperative agents that need to coordinate their activities. For example, consider a variant of the example in the introduction. Suppose in this case factory-1 is outsourcing both to factory-2 (owned by a different organization) and factory-3 (which is also owned by the parent company of factory-1). In this scenario, the scheduling agent of factory-1 will interact in a self-interested way with the scheduling agent of factory-2 while concurrently interacting in a cooperative way with the scheduling agent of factory-3.

There has also been a long tradition of work dating back to the inception of the field on coordination based on logical reasoning about the beliefs, desires, intentions (BDI) and commitments of agents [7, 20, 43, 49], and more recent work on the use of market mechanisms for solving multiagent resource allocation problems [54]. Similarly, there has been little cross-fertilization among these areas and the self-interested and cooperative camps. The synthesis of ideas from each of these different approaches to coordination holds great potential for future developments in the field.

Another issue is how to scale up to agent societies of hundreds and thousands of agents. There has been interesting work on cooperative behavior of a large number of agents [22, 27], and on organization self-design [13, 18]. However, this work has been done on simple reactive agents operating in artificial environments. Whether or not the results of this work can be applied to more complex agent societies operating in real-world environments is an open question. The challenge of how to design large-scale agent societies and how to evolve them as the environment changes is rapidly becoming a major issue in the field. Working on this problem will also shed light on many of the basic issues in multiagent systems research. For example, how complex must an agent be in order to interact effectively with other agents in a societal context? Is it best to think about an agent organization as an emergent property based on simple agent interactions, or do agents need to explicitly reason about and analyze their roles in the current organization to effectively adapt them to the changing state of the environment and the demands placed on the agent organization?

An important trend in the field is the development of analysis techniques to predict the performance of multiagent systems [4, 9, 46]. These performance characterizations also relate to the applicability of techniques. For example, it has been shown that certain self-interested agent interaction protocols can be guaranteed to produce truthful communications in problem domains with particular properties, while other domains can be guaranteed to produce lying [44]. The ability to bound the performance characteristics of systems is crucial to the acceptance of this field by the larger computer science community. Both the work on large-scale agent societies and this work on performance analysis are beginning to shift the field from focusing mainly on the syntax and semantics of agent interaction to the more encompassing study of the properties and characteristics of the aggregate behavior of agents.

Multiagent learning has also emerged as a major focus of study in the field over the last few years [55, 59]. This is not surprising because of the increasingly complex nature of multiagent systems, the fact that system performance can be very sensitive to the characteristics of the environment, and the dynamic and "open" operating environments of these systems. Thus, the ability to automatically tailor a system to its possibly evolving environment is crucial for its effectiveness. In the future, a learning component will be an integral part of the design of a multiagent architecture. To my knowledge, most of the current work in multiagent learning exploits existing learning algorithms that were designed to operate in a single-agent context; it will be interesting to see whether new learning techniques will evolve out of the multiagent character of the learning.

The issues and challenges discussed here so far are on the active agenda of the field. More long-term issues involve, for example, semantic interoperability among agents. How can agents with different internal representations, created at different times and operating in different environments, communicate and coordinate effectively? For example, can agents with different coordination protocols construct a new protocol that is appropriate for their intended interactions? Some of the surface issues associated with this problem are beginning to be studied in both the multiagent systems community [28, 42] and in the federated and multi-database community. However, the deeper issues still await serious work [17]. Another long-term issue is the integration of the work in the computer-supported cooperative work community, the intelligent user interfaces community, and the multiagent community. If a model of computing in which networks of computational agents and people seamlessly interact is to become a reality in the twenty-first century, then it seems obvious that these disparate fields will have to be more tightly integrated.

In summary, even though the use of multiagent systems technology is still in its infancy and the number of fielded commercial applications to date are small, there is tremendous potential and an exciting research agenda for the field. The field has already developed a rich set of concepts and mechanisms, both theoretical and practical, which will provide a solid base for future work. I expect the impact of multiagent systems on computer science to increase significantly during the next decade.

Acknowledgments

This material is based upon work that has been sponsored by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-97-1-0249; and by the National Science Foundation (under Grant Nos. IRI-9523419 and IIS-9812755). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency (DARPA), Air Force Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

References

[1] M.R. Adler, A. B. Davis, R. Weihmayer, and R. W. Worrest, "Conflict Resolution Strategies for Nonhierarchical Distributed Agents," L. Gasser and M. N. Huhns, eds., *Distributed Artificial Intelligence, Vol. II,* Pitman Publishing Ltd., 1989, pp. 139–162.

[2] M. Barbuceanu and M. S. Fox, "COOL: A Language for Describing Coordination in Multi-agent Systems," *Proc. First Int'l Conf. Multi-agent Systems*, AAAI Press, Menlo Park, CA, 1995, pp. 17–24.

[3] N. Carver and V. Lesser, "A New Framework for Sensor Interpretation: Planning to Resolve Sources of Uncertainty," *Proc. Nat'l Conf. Artificial Intelligence*, 1991, pp. 724–731.

[4] N. Carver and V. Lesser, "A Formal Analysis of Solution Quality in FA/C Distributed Sensor Interpretation Systems," *Proc. Second Int'l Conf. Multi-agent Systems*, AAAI Press, Menlo Park, CA, 1996, pp. 11–17.

[5] C. Castelfranchi, "Commitments: From individual intentions to groups and organizations," *AI and Theories of Groups & Organizations: Conceptual and Empirical Research.* Working Notes of AAAI Workshop, M. Prietula, ed., 1993.

[6] D. Cockburn and N. R. Jennings, "ARCHON: A Distributed Artificial Intelligence System for Industrial Applications," Chapter 12, G.M.P. O'Hare and N.R. Jennings, eds., *Foundations of Distributed Artificial Intelligence*, John Wiley & Sons, 1996, pp. 319–344.

[7] P. Cohen and H. J. Levesque, "Intention is Choice with Commitment," *Artificial Intelligence*, Vol. 42, No. 3, 1990, pp. 213–261.

[8] D.D. Corkill and V. Lesser, "The Use of Meta-Level Control for Coordination in a Distributed Problem Solving Network" (long paper), *Proc. Eighth Int'l Joint Conf. Artificial Intelligence*, 1983, pp. 748–756.

[9] K.S. Decker and V. Lesser, "An Approach to Analyzing the Need for Meta-Level Communication," *Proc. 13th Int'l Joint Conf. Artificial Intelligence*, 1993.

[10] K.S. Decker and V. Lesser, "Designing a Family of Coordination Algorithms," *Proc. First Int'l Conf. Multi-agent Systems*, AAAI Press, Menlo Park, CA, 1995.

[11] K.S. Decker, A. Pannu, K. Sycara, and M. Williamson, "Designing Behaviors for Information Agents," *Proc. First Int'l Conf. Autonomous Agents*, 1997.

[12] E.H. Durfee, V. Lesser, and D. D. Corkill, "Coherent Cooperation Among Communicating Problem Solvers," *IEEE Trans. Computers*, Vol. 36, No. 11, Nov. 1987, pp. 1275–1291.

[13] E.H. Durfee and T. A. Montgomery, "Coordination as Distributed Search in a Hierarchical Behavior Space," *IEEE Trans. Systems, Man, and Cybernetics*, Vol. SMC-21, No. 6, 1991, pp. 1363–1378.

[14] E.H. Durfee and J. S. Rosenschein, "Distributed Problem Solving and Multi-agent Systems: Comparisons and Examples," *Proc. Thirteenth Int'l Distributed Artificial Intelligence Workshop*, 1994, pp. 94–104.

[15] E.H. Durfee, “Blissful Ignorance: Knowing Just Enough to Coordinate Well,” *Proc. First Int’l Conf. Multi-agent Systems*, AAAI Press, Menlo Park, CA, 1995, pp. 406–413.

[16] T. Finin, R. Fritzson, D. McKay and R. McEntire, “KQML: An Information and Knowledge Exchange Protocol,” K. Fuchi and T. Yokoi, eds., *Knowledge Building and Knowledge Sharing*, Ohmsha and IOS Press, 1994.

[17] L. Gasser, “Social Conceptions of Knowledge and Action,” *Artificial Intelligence*, Vol. 47, No. 1, 1991, pp. 107–138.

[18] L. Gasser and T. Ishida, “A Dynamic Organizational Architecture for Adaptive Problem Solving,” *Proc. Ninth Nat’l Conf. Artificial Intelligence*, 1991, pp. 185–190.

[19] L. Gasser, “DAI Approaches to Coordination,” N.M. Avouris and L. Gasser, eds., *Distributed Artificial Intelligence: Theory and Praxis*, Kluwer Academic Publishers, Boston, 1992, pp. 31–51.

[20] B.J. Grosz and C. L. Sidner, “Plans for Discourse,” P.R. Cohen, J. Morgan, and M.E. Pollack, eds., *Intentions in Communication*, MIT Press, Cambridge, MA, 1990, pp. 417–444.

[21] B.J. Grosz and S. Kraus, “Collaborative Plans for Complex Group Action,” *Artificial Intelligence*, Vol. 86, No. 2, 1996, pp. 269–357.

[22] N. Glance and T. Hogg, “Dilemmas in Computational Societies,” *Proc. First Int’l Conf. Multi-agent Systems*, AAAI Press, Menlo Park, CA, 1995, pp. 117–124.

[23] P.J. Gmytrasiewicz and E. H. Durfee, “A Rigourous, Operational Formalization of Recursive Modeling,” *Proc. First Int’l Conf. Multi-agent Systems*, AAAI Press, Menlo Park, CA, 1995, pp. 125–132.

[24] J.Y. Halpern and Y. Moses, “Knowledge and Common Knowledge in a Distributed Environment,” *Proc. Third ACM Conf. Principles of Distributed Computing*, 1984.

[25] C. Hewitt, “Open Information Systems Semantics for Distributed Artificial Intelligence,” *Artificial Intelligence*, Vol. 47, No. 1, 1991, pp. 79–106.

[26] M.J. Huber and E. H. Durfee, “An Initial Assessment of Plan-Recognition-Based Coordination for Multi-agent Teams,” *Proc. Second Int’l Conf. Multi-agent Systems*, AAAI Press, Menlo Park, CA, 1996, pp. 126–133.

[27] B.A. Huberman and T. Hogg, “The Behaviour of Computational Ecologies,” B.A. Huberman, ed., *The Ecology of Computation*, North-Holland Publ., Amsterdam, 1988, pp. 77–115.

[28] M.N. Huhns, N. Jacobs, T. Ksiezyk, W. Shen, M. Singh, and P. Cannata, “Enterprise Information Modeling and Model Integration in Carnot,” C. J. Petrie Jr., ed., *Enterprise Integration Modeling*, MIT Press, Cambridge, MA, 1992, pp. 290–299.

[29] N.R. Jennings, “Commitments and Conventions: The Foundation of Coordination in Multi-agent Systems,” *The Knowledge Eng. Rev.*, Vol. 8, No. 3, 1993, pp. 223–250.

[30] N.R. Jennings, *Cooperation in Industrial Multi-agent Systems*, World Scientific Series in Computer Science, Vol. 43, World Scientific Publishing Co. Pte. Ltd., Singapore, 1994.

[31] N.R. Jennings, "Coordination Techniques for Distributed Artificial Intelligence," Chapter 6, G.M.P. O'Hare and N.R. Jennings, eds., *Foundations of Distributed Artificial Intelligence*, John Wiley & Sons, 1996, pp. 187–210.

[32] S. Lander and V. Lesser, "Sharing Meta-Information to Guide Cooperative Search Among Heterogeneous Reusable Agents," *IEEE Trans. Knowledge and Data Eng.*, Vol. 9, No. 2, Mar.–Apr. 1997, pp. 193–208.

[33] V. Lesser and D. D. Corkill, "Functionally-Accurate Cooperative Distributed Systems," *IEEE Trans. Systems, Man, and Cybernetics*, Special Issue on Distributed Problem-Solving, Vol. SMC–11, No.1, Jan. 1981, pp. 81–96.

[34] V. Lesser, "A Retrospective View of FA/C Distributed Problem Solving," *IEEE Trans. Systems, Man, and Cybernetics*, Vol. 21, No. 6, Nov.–Dec. 1991, pp. 1347–1362.

[35] J.G. March and H. A. Simon. *Organizations.* Wiley, New York, 1958.

[36] C.L. Mason and R. R. Johnson, "DATMS: A Framework for Distributed Assumption Based Reasoning," L. Gasser and M.N. Huhns, eds., *Distributed Artificial Intelligence*, Vol. 2, Morgan Kaufmann, Los Altos, CA, and Pitman, London, 1989, pp. 293–317.

[37] M.V. Nagendra Prasad and V. Lesser, "The Use of Meta-level Information in Learning Situation-Specific Coordination," *Proc. Fifteenth Int'l Joint Conf. Artificial Intelligence*, 1997.

[38] D. Neiman, D. Hildum, V. Lesser, and T. Sandholm, "Exploiting Meta-level Information in a Distributed Scheduling System," *Proc. of Twelfth Nat'l Conf. Artificial Intelligence*, 1994.

[39] T. Oates, M. V. Nagendra Prasad and V. Lesser, "Cooperative Information Gathering: A Distributed Problem Solving Approach," *IEE Proc. Software Eng.*, Special Issue on Agent-based Systems, Vol. 144, No. 1, 1997.

[40] Office of Technology Assessment, "Electronic Enterprises: Looking to the Future," 1994.

[41] H.V.D. Parunak, "Manufacturing Experience with the Contract Net," M.N. Huhns, ed., *Distributed Artificial Intelligence*, Morgan Kaufmann, San Mateo, CA, and Pitman, London, 1987, pp. 285–310.

[42] R. Patil, R. Fikes, P. Patel-Schneider, D. McKay, T. Finin, T. Gruber and R. Neches, "The DARPA Knowledge Sharing Effort: Progress report," *Proc. Third Int'l Conf. Principles of Knowledge Representation and Reasoning*, Morgan Kaufmann, 1992.

[43] A.S. Rao and M. P. Georgeff, "Modeling Rational Agents within a BDI-Architecture," *Proc. Second Int'l Conf. Principles of Knowledge Representation and Reasoning*, Morgan Kaufmann, San Mateo, CA, 1991.

[44] S. Rosenschein and G. Zlotkin, *Rules of Encounter: Designing conventions for automated negotiation among computers.* M. Brady, D. Bobrow, and R. Davis, eds., MIT Press, Cambridge, MA/London, 1994.

[45] T. Sandholm and V. Lesser, "Coalitions Among Computationally Bounded Agents," *Artificial Intelligence*, Special issue on Principles of Multi-agent Systems, Vol. 94, No.1, 1997.

[46] S. Sen, *Predicting Trade-offs in Contract-Based Distributed Scheduling*. Ph.D. Dissertation, Univ. of Michigan, 1993.

[47] H.A. Simon, *Models of Man*, Wiley, New York, 1957.

[48] H.A. Simon, *Models of Bounded Rationality,* Vol. 2, The MIT Press, Cambridge, MA, 1982.

[49] M.P. Singh, "Towards a Formal Theory of Communication for Multi-agent Systems," *Proc. 12th Int'l Jt. Conf. Artificial Intelligence,* Sydney, Australia, 1991, pp. 69–74.

[50] T. Sugawara and K. Murakami, "A Multi-agent Diagnostic System for Internetwork Problems," *Proc. of INET'92*, 1992.

[51] K.P. Sycara, S. Roth, N. Sadeh and M. Fox, "Distributed Constrained Heuristic Search," *IEEE Trans. Systems, Man, and Cybernetics*, Vol. SMC-21, No. 6, 1991, pp. 1446–1461.

[52] F. von Martial, "Coordinating Plans of Autonomous Agents," *Lecture Notes in Artificial Intelligence,* No. 610. Springer-Verlag, Berlin, 1992.

[53] R. Weihmayer and R. Brandau, "A Distributed AI Architecture for Customer Network Control," *GLOBECOM*, San Diego, CA, Dec. 1990.

[54] M. Wellman, "A Market-Oriented Programming Environment and Its Application to Distributed Multicommodity Flow Problems," *J. Artificial Intelligence Research,* Vol. 1, 1993, pp. 1–22.

[55] *Adaptation and Learning in Multi-agent Systems: Proc. of IJCAI'95 Workshop,* G. Weiβ and S. Sen, eds., Springer, Aug. 1995.

[56] *Artificial Intelligence*, Special issue on Principles of Multi-agent Systems, 1997.

[57] *Distributed Artificial Intelligence*, M. N. Huhns, ed., Pitman Publishing Ltd., London, 1987.

[58] *Distributed Artificial Intelligence*, Vol. 2, M.N. Huhns and L. Gasser, eds., Pitman Publishing Ltd., London, 1989.

[59] *Distributed Artificial Intelligence Meets Machine Learning: Learning in Multi-agent Environments*, Selected papers of ECAI'96 Workshop LDAIS and ICMAS'96 Workshop LIOME, G. Weiβ, ed., Springer, 1996.

[60] *Foundations of Distributed Artificial Intelligence*, G. O'Hare and N. Jennings, eds., Wiley Inter-Science, 1995.

[61] *IEEE Trans. Systems, Man, and Cybernetics,* Special Issue on Distributed Problem Solving, Vol. SMC–11, No.1, Jan. 1981.

[62] *IEEE Trans. Systems, Man, and Cybernetics.* Special Issue on Distributed AI, Vol. C–21, No. 6, Nov.–Dec. 1991.

[63] *Proceedings First Int'l Conf. Multi-Agent Systems,* AAAI Press, Menlo Park, CA, 1995.

[64] *Proceedings Second Int'l Conf. Multi-Agent Systems,* AAAI Press, Menlo Park, CA, 1996.

[65] *Readings in Distributed Artificial Intelligence*, A. Bond and L. Gasser, eds., Morgan Kaufmann Publishers, CA, 1988.

[66] D. Neiman, D.W. Hildum, V.R. Lesser, and T.W. Sandholm, "Exploiting Meta-Level Information in a Distributed Scheduling System," *Proc. Twelfth Nat'l Conf. Artificial Intelligence,* Seattle, 1994.

[67] N.R. Jennings, "Commitments and Conventions: The Foundation of Coordination in Multi-agent Systems," *The Knowledge Eng. Rev.*, Vol. 8, No. 3, 1993, pp. 223–250.

[68] N. Jennings, "Controlling Cooperative Problem Solving in Industrial Multi-agent Systems Using Joint Intentions," *Artificial Intelligence,* Vol. 75, No. 2, 1995.

[69] V.R. Lesser, "Reflections on the Nature of Multi-Agent Coordination and Its Implications for an Agent Architecture," *Autonomous Agents and Multi-Agent Systems*, Vol. 1, Kluwer Academic Publishers, 1998, pp. 89–111.

[70] A.S. Rao and M.P. Georgeff, "BDI agents: From Theory to Practice," *Proc. First Int'l Conf. Multi-Agent Systems,* AAAI Press, Menlo Park, CA, 1995, pp. 312–319.

[71] M. Tambe, "Agent Architectures for Flexible, Practical Teamwork," *Proc. Fourteenth Nat'l Conf. Artificial Intelligence*, Providence, 1997.

[72] S. M. Sutton, Jr. and L. J. Osterweil, "The Design of a Next-Generation Process Language," *Proc. Joint 6th European Software Eng. Conf. and 5th ACM SIGSOFT Symp. Foundations of Software Eng.*, Springer-Verlag: Zurich, Switzerland, 1997.

[73] *Proceedings Third Int'l Conf. Multi-Agent Systems,* IEEE Press, Paris, 1998.

[74] B. Huberman and S. H. Clearwater, "A Multi-agent System for Controlling Building Environments," *Proc. First Int'l Conf. Multi-Agent Systems,* AAAI Press: Menlo Park, CA, 1995, pp. 171–176.

[75] M. Boman, P. Davidsson, N. Skarmeas, K. Clark and R. Gustavsson, "Energy Saving and Added Customer Value in Intelligent Buildings," *Proc. Third Int'l Conf. Practical Application of Intelligent Agents and Multi-agent Technology,* 1998, pp. 505–517.

[76] M. Tambe and W. Zhang, "Towards Flexible Teamwork in Persistent Teams," *Proc. Third Int'l Conf. Multi-Agent Systems,* IEEE Press, Paris, 1998.

[77] T. Wagner, A. Garvey, and V. Lesser, "Criteria-Directed Task Scheduling," *Int'l J. Approximate Reasoning*, Vol. 19, 1998, pp. 91–118.

[78] S. Sen, "Reciprocity: A Foundational Principle for Promoting Cooperative Behavior among Self-Interested Agents," *Proc. Second Int'l Conf. Multi-Agent Systems*, AAAI Press, Menlo Park, CA, 1996, pp. 322–329.