A DAI Perspective on Cooperating Knowledge-Based Systems

Michael N. Huhns
MCC Corporation
Austin, TX USA

1 Motivations

Business operations, including sales, marketing, manufacturing, and design, can no longer be done in isolation, but must be done in a global context, i.e., as part of an enterprise. A characteristic of such enterprises is that their information systems are large and complex, and the information is in a variety of forms, locations, and computers. The information systems can be large in several senses: (1) they can be geographically distributed, (2) they can have many components, (3) they can have a huge content, both in the number of concepts and in the amount of data about each concept, and (4) they can have a broad scope, i.e., coverage of a major portion of a significant domain. Also, the components of the systems are typically distributed and heterogeneous. The topology of these systems is dynamic and their content is changing so rapidly that it is difficult for a user or an application program to obtain correct information, or for the enterprise to maintain consistent information.

There are four major techniques for dealing with the size and complexity of these enterprise information systems: modularity, distribution, abstraction, and intelligence, i.e., being smarter about how you seek and modify information. The use of intelligent, distributed modules is one way of combining all four of these techniques, yielding a distributed artificial intelligence (DAI) approach. In accord with this approach, computational agents need to be distributed and embedded throughout an enterprise. The agents could function as intelligent application programs, active information resources, "wrappers" that surround and buffer conventional components, and on-line network services. The agents would be knowledgeable about information resources that are local to them, and cooperate to provide global access to, and better management of, the information. For the practical reason that the systems are too large and dynamic (i.e., open) for global solutions to be formulated and implemented, the agents need to execute autonomously and be developed independently.

To maintain consistent information across such large enterprises requires the resolution of conflicts in both knowledge and data. DAI has developed several techniques for maintaining knowledge-level consistency. Research in distributed databases (DDB) has contributed techniques for data-level and schema-level consistency, especially when data is updated via distributed transactions. This combination of techniques from DAI and DDB has been termed cooperating knowledge-based systems (CKBS) [2]. In this paper I will focus on the contributions to CKBS from DAI.

2 DEFINITIONS AND HISTORY

Distributed artificial intelligence is a melding of artificial intelligence with distributed computing. From artificial intelligence comes the theory and technology for constructing or analyzing an intelligent system. But where artificial intelligence uses psychology as a source of ideas, inspiration, and metaphor, DAI uses sociology, economics, and management science for inspiration. Where the focus of artificial intelligence is on the individual, the focus of DAI is on the group. Distributed computing provides the computational substrate on which this group focus can occur.

However, DAI is more than just the design of intelligent systems. It also provides insights and understanding about interactions among humans, as they organize themselves into various groups, committees, societies, and economies in order to improve their lives. For example, economists have been studying multiple agents for more than two hundred years, ever since Adam Smith in the eighteenth century, with the goal of being able to understand and predict economies. Economics provides ways to characterize masses of agents, and these are useful for DAI. But in return, DAI provides a means to construct artificial economies that can test economists' theories before, rather than after, they are applied.

Moreover, DAI is growing, even at a time when AI itself is not. I think there are three major reasons for this: (1) DAI deals with open systems, i.e., systems that are too large or unpredictable to be completely characterized—most real systems are of this type; (2) DAI is the best way to characterize or design distributed computing systems; and (3) DAI provides a natural way to view intelligent systems. I will elaborate on each of these reasons in turn.

First, real systems cannot be meaningfully closed and bounded for analysis purposes. No matter how they are defined, they will always be subject to new information from outside themselves, causing unanticipated outcomes. For example, to analyze fully the operation of a banking system and produce answers to such questions as "How many of the customers will try to access the banking system at the same time, and will the system be able to handle the resulting load?" one must attempt to include all of the people that use the system. This is infeasible. By taking an open systems approach and a social perspective, DAI provides notions of systems of commitment and joint courses of action that permit such questions to be considered naturally.

Second, DAI is the best way to characterize or design distributed computing systems. Information processing is ubiquitous. There are computer processors seemingly everywhere, embedded in all aspects of our environment. My office has five, in such places as my telephone and my clock, and this number does not consider the electrical power system, which probably uses hundreds in getting electricity to my office. The large number of processors and the myriad ways in which they interact makes distributed computing systems the dominant computational paradigm today.

But there is a concomitant complexity in all this processing and interaction that is difficult to manage. One effective way is by considering such distributed computing systems in anthropomorphic terms. For example, it is convenient to think that "my toaster knows when the toast is done," and "my coffee pot knows when the coffee is ready." When these systems are interconnected so they can interact, then they should also know that the coffee and toast should be ready at approximately the same time. In these terms, my kitchen becomes more than just a collection of processors—a distributed computing system—it becomes a multiagent system.

Third, DAI also provides a natural way to view intelligent systems. Much of traditional AI has been concerned with how an agent can be constructed to function intelligently, with a single locus of internal reasoning and control implemented in a von Neumann architecture. But intelligent systems do not function in isolation--they are at the very least a part of the environment in which they operate, and the environment typically contains other such intelligent systems. Thus, it makes sense to view such systems in societal terms.

3 Societies of Long-Lived, Adaptive, Computational Agents

A program of research is needed to develop the basis and techniques for societies of autonomous computational agents that execute in open environments for indefinite periods.

There is a convergence of opportunities engendered by the development of large information environments, such as the national information infrastructure and the intelligent vehicle highway system, and preliminary explorations of multiagent systems. Planned information environments are too large, complex, dynamic, and open to be managed centrally or via predefined techniques--the only feasible alternative is for computational intelligence to be embedded at many and sundry places in such environments to provide distributed control. Each locus of embedded intelligence is best thought of as an autonomous agent that finds, conveys, or manages information. Because of the nature of the environments, the agents must be long-lived (they should be able to execute unattended for long periods of time), adaptive (they should be able to explore and learn about their environment, including each other), and social (they should interact and coordinate to achieve their own goals, and the goals of their society; they should rely on other agents to know things so they do not have to know everything).

Techniques for managing societies of autonomous computational agents will be useful not only for large open information environments, but also for large open physical environments. For example, such techniques would yield new efficiencies in defense logistics: by considering each item of materiel to be an intelligent entity whose goal is to reach a destination, a distribution system could manage more complicated schedules and surmount unforeseen difficulties.

Most importantly, agents must have the ability to acquire and use representations of each other. This is what is needed for negotiation, cooperation, coordination, and multiagent learning. What should be the contents of these representations? Books such as this one are providing the answer.

REFERENCES

[1] Rodney A. Brooks, "Intelligence without Representation," *Artificial Intelligence*, Vol. 47, 1991, pp. 139-159.

[2] S. M. Deen, "Cooperating Agents--A Database Perspective," in S. M. Deen, ed., *Cooperating Knowledge Based Systems 1990*, Springer-Verlag, London, 1991, pp. 3-29.

[3] Les Gasser and Michael N. Huhns, eds., *Distributed Artificial Intelligence, Volume II*, Pitman Publishing, London, 1989.

[4] Claudia V. Goldman and Jeffrey S. Rosenschein, "Emergent Coordination through the Use of Cooperative State-Changing Rules," *Proceedings of the Twelfth National Conference on Artificial Intelligence*, Seattle, WA, July 1994, pp. 408-413.

[5] Michael N. Huhns, ed., *Distributed Artificial Intelligence*, Pitman Publishing, London, 1987.

[6] Michael N. Huhns and David M. Bridgeland, "Multiagent Truth Maintenance," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 21, No. 6, December 1991, pp. 1437-1445.

[7] Amit P. Sheth and James A. Larson, "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases," *ACM Computing Surveys*, Vol. 22, No. 3, Sept. 1990, pp. 183-236.

[8] Jay K. Strosnider and C. J. Paul, "A Structured View of Real-Time Problem Solving," *AI Magazine*, Summer 1994, pp. 45-66.

[9] Gio Wiederhold, "Mediators in the Architecture of Future Information Systems," *IEEE Computer*, Vol. 25, No. 3, March 1992, pp. 38-49.