MCC Technical Report Number ACA-AI-056-88-Q

Distributed Knowledge-Based Systems for Antares: Research Plan

Michael N. Huhns and Larry M. Stephens

February 1988

ACA Confidential and Proprietary

Microelectronics and Computer Technology Corporation
Artificial Intelligence Laboratory
3500 West Balcones Center Drive
Austin, TX 78759
(512) 343-0978

Abstract

The goal of the Antares Project is to develop methods for interconnecting separately developed knowledge-based systems to enable them to cooperate in solving problems beyond the capabilities of any one of the knowledge-based systems. This objective will be achieved in two major phases.

1. Develop a knowledge-based system in CYC whose domain of expertise is the use and control of other knowledge-based systems that have also been developed in CYC. The operation of this system will depend on the crown of consensus reality that is currently being developed for CYC. This underlying common-sense knowledge base will relate terms that are common among the systems and will serve as the "glue" that interconnects the knowledge-based systems. A centralized control strategy is essential to understand what is needed for cooperation among separate knowledge-based systems. Thus, this phase of the research will provide the results needed to develop the more powerful system of autonomous, cooperating experts planned for the second phase.

2. Develop and implement in CYC principles of self-interest for each knowledge-based system, whereby each pursuing its own local goals results in coherent global problem-solving performance. These principles involve the local estimation of global goals and problem-solving states, and the communication of local goals, tasks, and results. This approach would enhance CYC by identifying and adding to it the knowledge and procedures needed to effect cooperation, such as those for negotiating among agents. This would be useful for human-computer interactions, as well as computer-computer interactions. In addition, CYC would obtain an awareness of other knowledge-based systems and a notion of self-interest.

1 Introduction

Distributed Artificial Intelligence (DAI) is concerned with the cooperative solution of problems by a decentralized group of agents. The agents may range from simple processing elements to complex entities exhibiting rational behavior. The problem solving is cooperative in that mutual sharing of information is necessary to allow the group as a whole to produce a solution. The group of agents is decentralized in that both control and data are logically and often geographically distributed.

DAI addresses two of the major problems currently confronting knowledge-based system developers (the first pragmatic and the second theoretical):

- Many expert system shells have been sold and there are an estimated 10,000 expert systems built or under development. *NONE* of these can work together, *i.e.*, if there is a problem that overlaps the domains of several of these systems, the only current solution is to build a new expert system for this overlapping domain.

- The current rule-of-thumb for knowledge engineering is to use only *ONE* expert. If there are several experts, or several nonexperts that together have the ability of an expert, there is *NO* current way to use them to engineer a successful knowledge-based system.

Distributed artificial intelligence suggests the following approach to these problems: build a separate subsystem for each problem domain based on the knowledge and expertise of each domain expert; then make these subsystems cooperate. This approach has the following advantages:

Knowledge acquisition: It is easier to find experts in narrow domains than in broader domains. Also, many problem domains are already partitioned or hierarchical. This is an advantage that should be exploited since problem decomposition and task planning are difficult problems.

Reusability: A small, independent knowledge-based system could be a part of *many* distributed knowledge-based systems—its expertise would not have to be reimplemented for each.

Capability: DAI can potentially solve problems that are too large for a centralized system, because of resource limitations induced by a given level of technology. Limiting factors such as communication bandwidths, computing speed, and reliability result in classes of problems that can be solved only by a distributed system.

Applicability: DAI is the most appropriate solution when the problem itself is inherently distributed, such as in weather forecasting, air-traffic control, and distributed information retrieval.

Modularity: The complexity of a knowledge-based system increases rapidly as the size of its knowledge base increases. Partitioning the system into N subsystems reduces the complexity by significantly more than a factor of N. The resultant system is easier to develop, test, and maintain.

Speed: The subsystems can operate in parallel.

Capability: By interconnecting separate knowledge-based systems that have different, but possibly overlapping expertise, the resultant system can solve problems whose domains are outside that of any one knowledge-based system. The result is that

$$IQ(system) \geq \max_i IQ(A_i)$$

where $IQ(A_i)$ represents the intelligence of the ith agent.

Solution quality: Different domain experts may suggest different solutions, some of which are better than others. Also the agents may interact, each learning from the other to improve the quality of its solutions.

Reliability: The system can continue to operate even if part of it fails.

The research problem that must be addressed is to develop methods to make the knowledge-based systems cooperate to achieve a global goal, much as groups of people are often able to do. In a group of people, cooperation occurs when each person believes that he as an individual will benefit more by cooperating than by acting otherwise. Cooperation for mutual benefit also occurs among groups of people: nations cooperate to improve security and businesses cooperate to leverage research and increase profits[1]. Each agent—an individual or a group—cooperates to improve its own interests. Interacting knowledge-based systems whose interests are compatible might intentionally cooperate or might unknowingly cooperate as they pursue their separate interests. In either case, self-interest can lead to cooperation.

However, effective cooperation requires each agent to know and understand the interests of the other participants. Cooperation among humans is possible because people have a common understanding of themselves and the world. Human-computer interactions are often difficult because neither adequately understands the other. Furthermore, computer-computer interactions are difficult because computers have a limited ability to understand other computers. In natural systems, such understanding evolved; in artificial systems, this understanding must be provided. Sufficient knowledge to enable this understanding and engender cooperation will be encoded in the CYC knowledge base as a part of the Antares Project.

2 Proposed Research

2.1 Objectives

The primary objective of the proposed research is to develop a robust and domain-independent mechanism for interconnecting several knowledge-based systems, each developed within the framework of CYC [20]. Currently, there are several systems that allow the cooperation of multiple knowledge-based systems. However, they require either that all the knowledge-based systems have identical knowledge bases (*e.g.*, DVMT [23]), or that the form and methods of cooperation be fixed before the systems are developed (*e.g.*, Hearsay-II [8]).

The proposed system will ease these requirements by enabling the cooperation of knowledge-based systems having different knowledge bases and different intended applications. This proposed system can be viewed as a new type of "shell" for the modular development of a knowledge-based system. It will use the common-sense knowledge in CYC to provide a basis for globally consistent semantics among the knowledge-based systems, but will not require globally consistent beliefs. This is compatible with the *local consistency hypothesis*[2] advanced by Lenat and Feigenbaum [21].

[1] In fact, this is the model for MCC.

[2] "There is no need—and probably not even any possibility—of achieving a *global* consistent unification of several knowledge-based systems' knowledge bases (or, equivalently, for one very large knowledge base). Large systems need *local consistency*."

The research objective will be met in two major phases. In the first phase, a knowledge-based system will be developed whose domain of expertise is the use of other knowledge-based systems. This system will both enforce and enable cooperation among the other knowledge-based systems. The architecture for this phase of the research is shown in Figure 1. In this hierarchy, the control expert explicitly directs the domain expert in solving the problem.

The second phase involves the development of principles of self-interest for each knowledge-based system, whereby each pursuing its own local goals results in a globally coherent process of problem solving (Figure 2). In effect, the knowledge of the control expert developed in the first phase of the research will be distributed among the collection of domain knowledge-based systems; the principles of self-interest will be used as a focus-of-control mechanism for the entire system. These principles involve the local estimation of global goals and problem-solving states, and the communication of local goals, tasks, and results. This approach would enhance CYC by the identification and addition of the knowledge and procedures needed to effect cooperation, such as those for negotiating among agents. This would be useful for human-computer interactions, as well as computer-computer interactions. In addition, CYC would obtain an awareness of other knowledge-based systems and a notion of *self-interest*.[3]

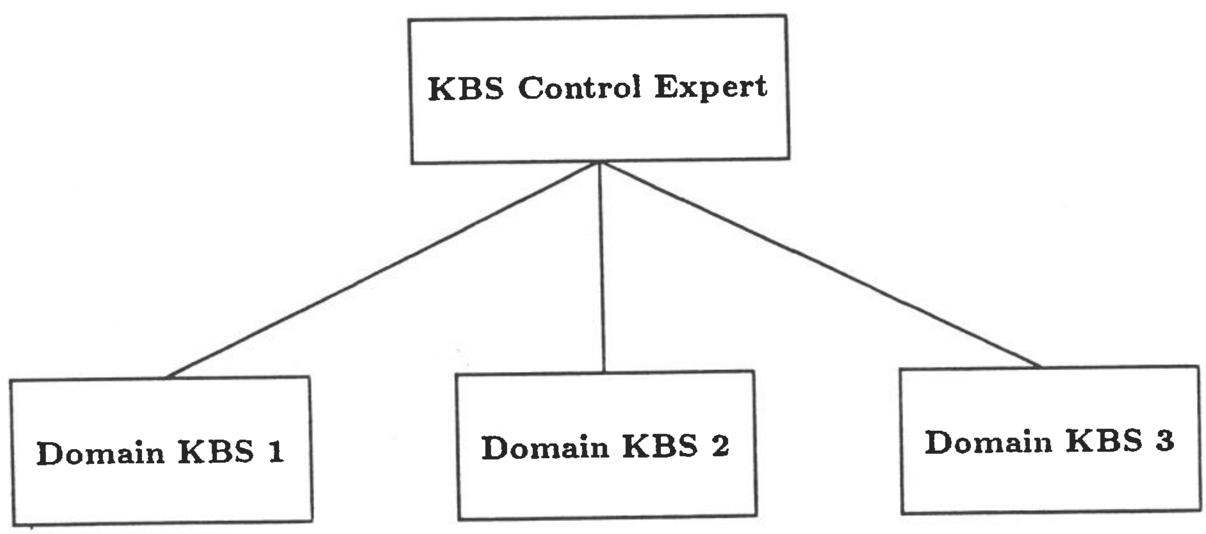

Figure 1: A distributed artificial intelligence system based on centralized control for cooperation: cooperation occurs if the domain KBSs are managed well

[3]A *self-interested agent* is one that is pursuing its own goals, knows about other agents, and tries to use their capabilities to work toward its goals.

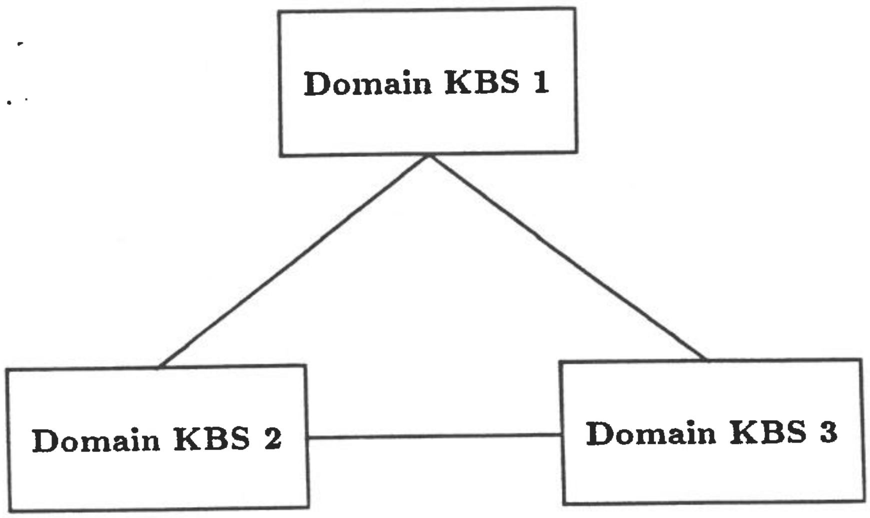

Figure 2: A distributed artificial intelligence system based on cooperating autonomous knowledge-based systems: cooperation occurs if the domain KBSs know about themselves and each other, and communicate intelligently

2.2 Research Plan

The research needed to meet the above objective will be partitioned into the following tasks:

1. Mechanisms for control that enable cooperative problem-solving behavior among a set of knowledge-based systems will be developed. These mechanisms include centralized control, negotiation, and self-interest. All require the communication of tasks and results. The research will determine when the knowledge-based systems must communicate, what they must communicate, and what language they should use for communication.

2. A set of knowledge-based systems envisioned for the domain of digital logic design will be analyzed to determine the requirements for a controlling knowledge-based system that could interconnect them and cause them to cooperate. The centralized control mechanism developed in the first research task will be used to achieve this cooperation. The fundamental strategy for centralized control will be an iteration of the following steps:

   - Decompose the current design state into a set of design tasks
   - Assign the design tasks to the domain experts
   - Synthesize the returned results into a current design state

- Evaluate the current design state to determine if a satisfactory solution has been reached.

For decomposing the design, the control expert needs knowledge about planning, design, and the capabilities of the domain experts. For assigning tasks, the control expert must be knowledgeable about the domain experts' capabilities, workloads, and current status in performing other tasks. Result synthesis requires domain knowledge, common-sense knowledge to evaluate the suitability of partial results, and an estimate of the control expert's confidence in the partial results received from the domain experts.

3. A small set of knowledge-based systems, each with expertise in some aspect of digital logic design, will be developed in CYC. These will then be interconnected through the implementation of the controlling knowledge-based system whose principles were developed in the first task.

4. The resultant distributed knowledge-based system will be evaluated in terms of performance and capability by applying it to design problems. Two approaches will be considered.

   - Each domain expert can apply *different* design techniques to the *same* design problem to investigate the following hypothesis: through cooperation it is possible to obtain designs that are superior to those obtainable from any *one* of the knowledge-based systems.
   - Each domain expert, using the *same* design technique, can produce designs that explore *different* hypothetical worlds. In this case, each knowledge-based system is given one option of a trade-off to evaluate and uses negotiation to resolve competing solutions. Negotiation is potentially a more powerful technique than contradiction resolution because it allows compromises, rather than demanding total consistency.

   Metrics will be developed to formalize these evaluations. The following questions suggest metrics for comparing a DAI approach to a single knowledge-based system approach:

   - Is a solution possible using a DAI approach that is not possible using a single domain knowledge-based system?
   - Is the quality of the solution better using a DAI approach?
   - Is the elapsed solution time less using a DAI approach?

5. The principles of self-interest defined above will be developed and encoded into the CYC knowledge base. Cooperation can then take place without the agents incurring any overhead to establish and maintain it. Cooperation itself is important because

the agents are resource-bounded in time, knowledge, and data. Additional considerations for establishing cooperation among autonomous agents include determining how much domain knowledge they should have in common (in addition to the same consensus of reality); how much knowledge of themselves or other knowledge-based systems is needed; and how much communication and control knowledge is required.

6. We envision that the CYC system will be used in different design environments. As local users add knowledge to each copy of CYC, each will become more specialized in certain areas. Thus, each local copy of CYC will itself be a candidate for an intelligent agent with different areas of expertise and power. A fundamental problem-solving paradigm is to seek help when the problem cannot be solved by the agent working on it; this paradigm will be added to CYC's repertoire of problem-solving techniques. The agent providing the help will be another of these specialized versions of CYC. Cooperation will require a representation for the goals, tasks, and results of a knowledge-based system so that these can be communicated. Further, cooperation will require that the state of the problem-solving process for a knowledge-based system be identified, so that it too can be communicated to the other knowledge-based systems [11,12]. Information about this state will also be useful for explanations.

3 Background on DAI

Current research in DAI can be characterized and understood in the context of the following paradigm: there is a collection of agents (logically-distinct processing elements) that are attempting to solve a problem. Although there are identifiable global states and goals of the problem-solving, each agent has only a partial and inexact view of these.[4] Each agent attempts to recognize, predict, and influence the global state such that its local view of the goals is satisfied. The methods an agent has available to it are to *compute* solutions to subproblems, which are in general interdependent, and to *communicate* information such as results, tasks, goals, and estimates of the problem-solving state.

The following items can help characterize a distributed knowledge-based system:

- The individual systems comprising the distributed knowledge-based system can have either the same or different knowledge and capabilities.
- This knowledge can be either correct or incorrect.
- The individual knowledge-based systems can have either the same or different data.
- This data can be either correct or noisy.
- The individual knowledge-based systems can have either the same or different goals.

[4] The global state may not be determinate if the distributed system is asynchronous.

- The individual knowledge-based systems can have either the same or different semantics for their terms.
- The individual knowledge-based systems can be written in either the same language or different languages.

For example, in an application to air-traffic control, the individual knowledge-based systems would have identical goals, semantics, knowledge, and capabilities, would be written in the same language, but would have different data to process. In an application to troubleshooting oscilloscopes, there may be individual knowledge-based systems with expertise in the diagnosis of power supplies, function generators, and cathode-ray tubes, but they would all have available the same diagnostic data.[5]

Research is underway on all aspects of the DAI paradigm and a cross-section of it is presented below. Two major areas of DAI research address ways to develop control and communication abstractions that efficiently guide problem solving, and description abstractions that result in effective organizational structure.

By reorganizing and managing others, an agent may be able to achieve its goals: the other agents then work to achieve the global state desired by the first. This is sometimes done by transmitting tasks, and possibly the knowledge needed to solve them, to the other agents. These organizational issues are addressed by Durfee *et al.* [4]. Further, Pattison *et al.* [26] present a language for describing, forming, and controlling organizations of agents.

Control and communication abstractions can be explicit or implicit. Examples of explicit abstractions are negotiation, blackboard architectures, multiagent planning, and the functionally accurate approach. Negotiation has been formalized by Davis and Smith [3,24] in a contract net that specifies communication and control mechanisms for distributed problem-solving environments. Task distribution in their system is an interactive process modeled after contract negotiation. A node having a task to perform decomposes it into subtasks and assigns these subtasks through a bidding procedure. The purpose of the bidding is to find the node that is best suited to perform a particular subtask. A disadvantage of this approach is that a node best suited for a subsequently defined subtask may have already accepted some other subtask and thus be unavailable to assist. This is because there is no way for a node to predict whether a more appropriate subtask will be offered for selection in the future.

Barbara Hayes-Roth [15] has developed an architecture for intelligent control of problem solving based on a blackboard data structure that distinguishes domain problems, control problems, knowledge, and solutions. A key feature of this architecture is its ability to respond in an opportunistic manner to unanticipated problem-solving situations. The BB1 system has recently been used to solve problems in which a set of objects must be arranged in some context to satisfy a set of constraints [16]. This architecture has also been applied in creating a distributed version of the OPS5 rule language that solves engineering problems using networks of heterogeneous computers [19]. Ensor and Gabbe [6]

[5]Note an advantage of DAI: the power supply knowledge-based system could also be used in knowledge-based systems for troubleshooting other electronic equipment.

have addressed the problems of synchronization and scheduling for concurrent execution of knowledge sources within a blackboard framework.

Multiagent planning systems require that the agents synchronize their activities to achieve their goals. In addition, the agents need to reason about their actions in relation to the other agents [13]. Researchers at the Rand Corporation have developed a distributed planner for air-traffic control in which each simulated aircraft has a knowledge-based system that senses the surrounding airspace for other aircraft, plans and evaluates flight paths, communicates with the other aircraft, and controls the execution of the flight plan [1,25]. The results indicate that the best overall performance is obtained using a policy that selects, as the primary planner, the aircraft with the least number of flight-path constraints, rather than the aircraft with the most knowledge about the environment.

The functionally accurate, cooperative approach to distributed problem solving is based on nodes that cooperate by generating and exchanging tentative, partial solutions derived from their limited local views of a network problem [22]. Through the iterative exchange of potentially incomplete, inaccurate, and inconsistent partial solutions, the nodes eventually converge on a globally consistent solution. This procedure has been successfully applied to speech understanding [7,8] and the tracking of vehicles in a distributed sensor network [23]. The use of more knowledgeable local control has been shown to improve the coherence of the problem-solving [5].

Durfee it et al. [4] propose that self-interest can lead to cooperation in a collection of problem-solving nodes if the individual nodes have compatible goals. The primary strategy is to promote intelligent communication based on self-interest so that the nodes work together coherently. This is accomplished by enabling each node to understand its current and probable future problem-solving activities.

Similarly, Ginsberg [14] describes principles of rationality by which agents solving interdependent subproblems can achieve their goals without communicating explicit control information among themselves. Each agent, however, must be aware of the goals (in the form of payoff matrices) of the other agents. Cooperation then ensues in spite of the fact that each agent is entirely self-interested.

Several researchers have introduced models for knowledge and belief that we will consider for representing the concept of self-interest in CYC. Fagin *et al.* [9] define a semantic model that has the following properties:

- All knowledge in the system is assumed to be correct
- Agents know all consequences of their knowledge
- Agents are aware of their knowledge but not of their ignorance.

Konolige [18] presents a *deduction structure* for the belief system of an agent. This structure is deductively closed, but it does not require logical omniscience. It is thus a better model than the one in [9] for an agent's common-sense reasoning.

4 Conclusion

It is expected that the development of distributed knowledge-based systems in Antares will yield the following results:

- A knowledge-based system, relying on CYC, whose domain of expertise is the use of other knowledge-based systems
- A new type of knowledge-based system shell
- A new approach to hypothetical worlds
- Mechanisms, augmenting CYC, that allow knowledge-based systems to cooperate
- A capability to expand the usefulness of existing knowledge-based systems
- A distributed knowledge-based system for the domain of VLSI design
- Increased usefulness of CYC.

In particular, research on distributed knowledge-based systems for Antares will add the following features to CYC:

- Knowledge and awareness of other knowledge-based systems (agents)
- Knowledge of and procedures for negotiating among agents
- Strategies for resolving conflicts among agents
- A notion of **internal process**, *i.e.*, a mechanism for storing a trace of its knowledge-retrieving and problem-solving activities. This can be learned, abstracted, communicated and used to guide future knowledge-retrieving and problem-solving activities.

DAI provides a large step beyond current knowledge-based systems. DARPA has recognized its importance by including it in the Strategic Computing Initiative and making it a 1992 milestone for research in knowledge-based systems [2]. The ABE project at Teknowledge is attempting to industrialize the development of intelligent systems by producing a multilevel architecture that supports aggregations of cooperating, autonomous, problem-solving agents [7]. Intellicorp has recently begun a project to build a distributed knowledge-based system testbed and then use it to evaluate different architectures involving both cooperation and competition among its subsystems [10]. Cooperating knowledge-based systems is a major research theme for the European ESPRIT II program and the United Kingdom Alvey Program [27]. It is appropriate that MCC also conduct research in DAI.

References

[1] S. Cammarata, D. McArthur, and R. Steeb, "Strategies of Cooperation in Distributed Problem Solving," *Proceedings of Eighth International Joint Conference on Artificial Intelligence*, 1983, pp. 767–770.

[2] "DARPA: Strategic Computing Initiative," *Communications of the ACM*, vol. 28, no. 7, July 1985, pp. 692–695.

[3] R. Davis and R. G. Smith, "Negotiation as a Metaphor for Distributed Problem Solving," *Artificial Intelligence*, vol. 20, 1983, pp. 63–109.

[4] E. H. Durfee, V. R. Lesser, and D. D. Corkill, "Cooperation Through Communication in a Distributed Problem Solving Network," in M. N. Huhns, ed., *Distributed Artificial Intelligence*, Pitman Publishing Ltd., London, England, 1987.

[5] E. H. Durfee, V. R. Lesser, and D. D. Corkill, "Coherent Cooperation Among Communicating Problem Solvers," *IEEE Transactions on Computers*, vol. C-36, no. 11, November 1987, pp. 1275–1291.

[6] J. R. Ensor and J. D. Gabbe, "Transactional Blackboards," *Artificial Intelligence in Engineering*, vol. 1, no. 2, 1986, pp. 80–84.

[7] L. D. Erman, M. Fehling, S. Forrest, and J. Lark, "ABE: Architectural Overview," *Proceedings of the Fifth Workshop on Distributed Artificial Intelligence*, December 1985, pp. 321–336.

[8] L. D. Erman, F. Hayes-Roth, V. R. Lesser, and D. R. Reddy, "The Hearsay-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty," *Computing Surveys*, vol. 12, no. 2, June 1980, pp. 213–253.

[9] R. Fagin, J. Y. Halpern, and M. Y. Vardi, "What Can Machines Know? On the Epistemic Properties of Machines," *Proceedings of Fifth National Conference on Artificial Intelligence*, vol. 1, August 1986, pp. 418–434.

[10] R. E. Filman, "Architectures for Distributed Problem Solving Systems," *Proceedings of the Fifth Workshop on Distributed Artificial Intelligence*, December 1985, pp. 191–192.

[11] M. P. Georgeff and A. L. Lansky, "Reactive Reasoning and Planning," *Proceedings of Sixth National Conference on Artificial Intelligence*, vol. 2, July 1987, pp. 677–682.

[12] M. P. Georgeff and A. L. Lansky, "Procedural Knowledge," *Proceedings of the IEEE Special Issue on Knowledge Representation*, vol. 74, 1986, pp. 1383–1398.

[13] M. P. Georgeff, "A Theory of Action for MultiAgent Planning," *Proceedings of the Fourth National Conference on Artificial Intelligence*, August 1984, pp. 121–125.

[14] M. L. Ginsberg, "Decision Procedures," in M. N. Huhns, ed., *Distributed Artificial Intelligence*, Pitman Publishing Ltd., London, England, 1987.

[15] B. Hayes-Roth, "A Blackboard Architecture for Control," *Artificial Intelligence*, vol. 26, 1986, pp. 251–321.

[16] B. Hayes-Roth, M. V. Johnson, A. Garvey, and M. Hewitt, "Application of the BB1 Blackboard Control Architecture to Arrangement Assembly Tasks," *Artificial Intelligence in Engineering*, vol. 1, no. 2, 1986, pp. 85–94.

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

[18] K. Konolige, *A Deduction Model of Belief*, Pitman Publishing Ltd., London, England, 1986.

[19] L. V. Leao and S. N. Talukdar, "An Environment for Rule-Based Blackboards and Distributed Problem Solving," *Artificial Intelligence in Engineering*, vol. 1, no. 2, 1986, pp. 70–79.

[20] D. B. Lenat, J. Huffman, and R. V. Guha, "CYCL: The CYC Representation Language," MCC Proprietary Technical Manual (Draft), October 1987.

[21] D. B. Lenat and E. Feigenbaum, "On the Thresholds of Knowledge," *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milan, Italy, August 1987, pp. 1173–1182.

[22] V. R. Lesser and D. D. Corkill, "Functionally Accurate, Cooperative Distributed Systems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-11, no. 1, January 1981, pp. 81–96.

[23] V. R. Lesser and D. D. Corkill, "The Distributed Vehicle Monitoring Testbed: A tool for investigating distributed problem solving networks," *AI Magazine*, vol. 4, no. 3, Fall 1983, pp. 15–33.

[24] R. G. Smith and R. Davis, "Frameworks for Cooperation in Distributed Problem Solving," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-11, no. 1, January 1981, pp. 61–70.

[25] P. W. Thorndyke, D. McArthur, and S. Cammarata, "AUTOPILOT: A Distributed Planner for Air Traffic Control," *Proceedings of Seventh International Joint Conference on Artificial Intelligence*, pp. 171–177, 1981.

[26] H. E. Pattison, D. D. Corkill, and V. R. Lesser, "Instantiating Descriptions of Organizational Structures," in M. N. Huhns, ed., *Distributed Artificial Intelligence*, Pitman Publishing Ltd., London, England, 1987.

[27] D. Sleeman, "IKBS Strategy: recommendations for IT92," Alvey Directorate, 1987.