Cooperation for DAI through Common-Sense Knowledge

Michael N. Huhns, Larry M. Stephens, and Douglas B. Lenat

Microelectronics and Computer Technology Corporation
Artificial Intelligence Laboratory
3500 West Balcones Center Drive
Austin, TX 78759
(512) 338-3651
huhns@MCC.COM

1 Introduction

The goal of the Antares Project at MCC is to develop methods that enable separately developed knowledge-based systems to cooperate in solving problems beyond the capabilities of any one of the knowledge-based systems. There have been many previous attempts to develop distributed systems of cooperating experts (see [8] for a comprehensive summary of these). For the most part, these attempts have been *ad hoc* and have met with few successes. The resultant systems have been domain dependent and did not generalize. Further, it has not been sufficient that each knowledge-based system be written in the same language [9] or have access to a global data structure, such as a blackboard [5, 6]. The fundamental hypothesis of our research is that these systems lack the common-sense understanding of the world that is necessary for cooperation and that is possessed by cooperating humans.

Our approach for engendering cooperation among distributed knowledge-based systems utilizes 1) a large global base of common-sense knowledge that permits enforcement of semantic consistency and 2) an explicit characterization of self-interest to control global problem solving while allowing autonomy. It has been shown [1, 4] that autonomous systems can cooperate without the explicit exertion of control if they are self-interested. A general notion of self-interest for an autonomous knowledge-based system must include representations for the goals and problem-solving state of this system, as well as models of the goals and problem-solving states of the other systems with which it interacts. However, interaction and communication among these systems cannot occur without a common understanding of the global and individual problems the systems are attempting to solve. The difficulty is that there has been no way to provide this understanding, which is essential for intelligent interaction among the systems and for exploiting the full power of the notion of self-interest. Our strategy is to use CYC [10, 11] to provide the necessary common-sense knowledge, as well as to provide the means and framework for characterizing self-interest.

2 CYC

At MCC, the CYC Project is building a large common-sense knowledge base of real-world facts, heuristics, and reasoning mechanisms. This knowledge base is being developed to assist with knowledge acquisition for constructing knowledge-based systems and to provide a means for overcoming brittleness in using knowledge-based systems. This development is also predicated on the notion that the more we know the more we can learn; development of the CYC system is thus self-synergistic.

The CYC system consists of the large knowledge base, a user interface, and a knowledge-representation language. The knowledge-representation language is frame-based; there is a frame in the system to represent each *instance* of an object, slot, function, problem-solving method, *etc.* In addition, there are frames to represent the *concepts* of an object, slot, function, problem-solving method, *etc.* The language is thus self-describing. Each frame in the knowledge base is linked to others via its slots, which can inherit values from other slots through multiple inheritance. The inheritance can occur along any slot in any direction and can be conditional.

The CYC knowledge base, besides encoding the CYC representation language, includes explicit representations for both encyclopedic knowledge and common-sense knowledge. Storage of this knowledge is expected to require approximately one-million frames. The frames do not represent simply a list of facts, but are highly interconnected, much as concepts are in the real world. This interconnection allows problems to be solved by using analogies to knowledge not closely related to the problems.

3 Using CYC for DAI

Antares is utilizing CYC as a framework within which the necessary components of self-interest—goals, tasks, problem-solving states, and results—can be implemented. As mentioned above, effective cooperation can be achieved through self-interest [1]. However, an implementation of self-interest requires more than just an explicit representation for the goals of a knowledge-based system and a stipulation that the system must pursue these goals [4]. It also requires each system to be aware of the goals and problem-solving capabilities of the other knowledge-based systems, as well as the current status of its own problem solving. Further, each system must be aware of and *understand* what goals and tasks are. This means we must represent such abstract concepts as "goal" and "task," in addition to specific instances of goals and tasks from a problem-solving domain. By providing the knowledge-based system with an understanding of such concepts, it would realize, for example, that satisfying a goal would be rewarded and not satisfying a goal would be penalized. This is precisely what is meant by "self-interest."

In the Hearsay-II speech understanding system [3], the knowledge sources were unaware of each other, of themselves, and even of the overall goals of the system. In later systems, such as DVMT [12], each knowledge source had an ability to transmit tasks and hypotheses to other knowledge sources and receive results and confirmations. This enabled them to cooperate more effectively, even though they had only a limited awareness of the other knowledge sources. This was later improved by providing each knowledge source with network information and a mechanism for refining this network [2]. In MINDS [7], the knowledge sources had a more complete awareness of the others, consisting of models of their capabilities and behaviors; these models were learned during the course of problem-solving. In Antares, we are expanding and exploiting this awareness more fully by explicitly representing the semantics of self-interest.

Self-interest, however, is not sufficient: cooperation will not occur without a common understanding of the domains involved. A limitation of current knowledge-based systems is that they have no understanding of the predicates they use to represent knowledge. For example, in a knowledge-based system for the domain of digital circuit design, an assertion of the form

```
(fan-out ?output-signal ?value)
```

might be used to represent facts about digital signals. Another knowledge-based system for mechanical design might use the same predicate name to represent the output of air from a fan. The problem is that these systems have no internal representation for the meaning of `fan-out`; the semantics must be supplied by a user external to the systems and, from the standpoint of these systems, the predicate might just as well have been named `foo`. Without any link to the real world or to a common knowledge base representing the real world, there is no reason to expect these systems to have a consistent semantics for their terms; the result is that these systems would conflict rather than cooperate.

However, if these two systems represented their knowledge within the CYC framework, then, after one system had created the predicate `fan-out`, the other system would be prohibited from changing that definition. The other system would be required to create a different frame for its notion of `fan-out`. Thus, the usage of the term `fan-out` would necessarily be consistent. If the systems have the same underlying representation and semantics and communicate at this level, then differences in the surface form of their knowledge become unimportant. The enforcement of semantics is discussed more fully in [10].

For example, imagine two knowledge-based systems developed separately for the domain of digital circuit design, one for designing circuits using two-level NAND gates and the other for designing circuits using two-level NOR gates. Each of these is implemented in CYC, thus insuring that each has access to the same base of fundamental common-sense knowledge. This knowledge includes Boolean algebra (specifically, DeMorgan's theorem). Suppose a decomposable problem is given to the NAND-gate system in sum-of-products form to design a circuit out of NAND gates only. The NAND-gate system can learn of the existence of the NOR-gate system from the CYC knowledge base (CYC can provide a list of available problem solvers for the domain of digital circuit design), decompose the problem, and send a subproblem to the NOR-gate system. The NOR-gate system can convert the subproblem into product-of-sums form and produce a solution using two-level NOR gates. This solution can be sent to the NAND-gate system, which converts it to two-level NAND gates using DeMorgan's theorem. The converted solution can finally be combined with solutions obtained by the NAND-gate system to the other subproblems. The two systems are able to cooperate because of the common underlying knowledge provided by CYC, the specific knowledge of problem solvers, and a common-semantics for all of this knowledge.

4 Conclusions

The Antares Project has just recently been initiated. Work is underway on developing autonomous knowledge-based systems for the domain of digital circuit design and on encoding the concepts of self-interest and problem-solving state within the CYC knowledge base. Our research is helping to solve epistemological problems in DAI involving inconsistent, incomplete, and mutual knowledge among distributed knowledge-based systems. By providing a common-sense knowledge base that these systems can share, we are helping to insure coherence. However, we have not addressed issues of allocating problems among the systems and modeling other systems, and many questions remain to be answered.

References

[1] 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, pp. 29–58.

[2] 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.

[3] 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.

[4] M. L. Ginsberg, "Decision Procedures," in M. N. Huhns, ed., *Distributed Artificial Intelligence*, Pitman Publishing Ltd., London, England, 1987, pp. 3–28.

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

[6] 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.

[7] M. N. Huhns, U. Mukhopadhyay, L. M. Stephens, and R. D. Bonnell, “DAI for Document Retrieval: The MINDS Project,” in M. N. Huhns, ed., *Distributed Artificial Intelligence*, Pitman Publishing Ltd., London, England, 1987, pp. 249–283.

[8] V. Jagannathan and R. Dodhiawala, “Distributed Artificial Intelligence: An Annotated Bibliography,” in M. N. Huhns, ed., *Distributed Artificial Intelligence*, Pitman Publishing Ltd., London, England, 1987, pp. 341–390.

[9] 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.

[10] D. B. Lenat, M. Prakash, and M. Shepherd, “CYC: Using Common Sense Knowledge to Overcome Brittleness and Knowledge Acquisition Bottlenecks,” *AI Magazine*, vol. 6, no. 4, Winter 1986, pp. 65–85.

[11] 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.

[12] 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.