

THE APPLICATION OF ARTIFICIAL INTELLIGENCE TECHNIQUES TO COOPERATIVE DISTRIBUTED PROCESSING

Victor R. Lesser and Daniel D. Corkill

Computer and Information Science
University of Massachusetts
Amherst, Massachusetts, 01003

Knowledge-based Artificial Intelligence (AI) systems have incorporated mechanisms which resolve uncertainty. This uncertainty stems from incompleteness and noise in input data and from errorful processing.

Resolution of uncertainty is also an important issue in the design of distributed processing systems. Uncertainty is introduced in these systems from the use of incomplete and inconsistent local data bases and from errorful communication channels.

The mechanisms used in knowledge-based AI systems provide a model for the design of distributed algorithms which can resolve uncertainty as an integral part of their problem solving activity. Use of such algorithms in a distributed processing system makes possible a reduction in the amount of inter-node communication required to resolve uncertainty. This reduction in communication requirements allows effective distribution of applications that are impractical using current approaches to the design of distributed algorithms.

1. INTRODUCTION

Conventional approaches to distributed-system design can be characterized by their emphasis on the maintenance of correctness in all aspects of the distributed computation. The distributed processing system is organized so that a processing node's local data bases contain exact copies of appropriate portions of the overall problem solving data base needed by the node's algorithms [1, 2].

In these systems, a node rarely needs the assistance of another node in carrying out its problem solving function. We call this type of distributed processing decomposition completely-accurate, nearly-autonomous (CA/NA), because each node's algorithms operate on complete and correct information ("completely-accurate") and because each node usually has in its local data base the information it requires to complete processing ("nearly-autonomous"). When this information is not locally available, a node must request another node to calculate the information, which is returned as a complete and correct result.

The CA/NA approach, however, is not suitable for applications in which algorithms and control structures cannot be replicated or partitioned based on the distribution of data in the network. In this situation, a CA/NA system is very expensive to implement due to the high communication and synchronization

costs required to guarantee completeness and consistency of the local data bases. We feel that the almost exclusive use of the CA/NA approach has severely limited the application areas to which distributed processing has been effectively applied. This is especially true for distributed interpretation and monitoring applications. Although apparently well-suited to distributed processing implementations, these applications are difficult to implement effectively using CA/NA techniques. This difficulty arises because a non-local view of the sensory data is required to perform the processing.

An alternative and new approach to structuring distributed problem solving systems is to allow incorrect and inconsistent intermediate results and a range of acceptable answers. In this approach, inter-node communication can be reduced, at the risk of inconsistency and incompleteness of local views and the possibility of unnecessary, redundant, or incorrect processing. We call a system with this problem solving structure functionally-accurate (FA) because it exhibits correct (within tolerance) input/output behavior but is distinct from completely-accurate problem solving structures, in which all intermediate aspects of the computation are required to be correct and consistent.

2. A MODEL FOR FUNCTIONALLY-ACCURATE, COOPERATIVE DISTRIBUTED SYSTEMS

FA distributed systems are more complex than CA/NA distributed systems because the algorithms and control structures operate on

This research was supported by National Science Foundation Grant MCS78-04212.

local data bases which are incomplete, inconsistent, and possibly errorful. In order to resolve the uncertainties in these local data bases and still keep communication bandwidth low, nodes must exchange partial results (at various levels of abstraction) and share common goals. Since new information may be based on processing which used incomplete or incorrect data, an iterative, coroutine type of node interaction is required to resolve uncertainty. This type of interaction leads us to view such a distributed system as a "cooperative network of interrelated tasks." Therefore, we call such FA systems functionally-accurate, cooperative (FA/C).

In FA/C systems, it is often appropriate to have the control of cooperation among the nodes decentralized and implicit. Each node uses its local estimate of the state of network problem solving to control its processing (i.e., what new information to generate) and its transmissions to other nodes [3]. This allows node activity to be self-directed. For instance, if a node does not receive an appropriate partial result in a given amount of time, it is able to continue processing, utilizing whatever data are available at that time.

When organizing a FA/C distributed system, it is appropriate to think of the system as being synthesized from local systems operating at each node. This is in contrast with the way a CA/NA distributed system is thought of: a centralized system distributed over a network, with each piece (node) in the decomposition viewed as a part of the whole system.

In a FA/C system, it is natural to think of dealing with uncertainty and errors in control, data, and algorithms caused by the distribution as an integral part of the network problem solving process. In fact, additional mechanisms required to handle hardware, communication, and processing errors may be unnecessary given that uncertainty resolving mechanisms are already an integral part of the distributed system architecture.

3. KNOWLEDGE-BASED AI AND COOPERATIVE DISTRIBUTED SYSTEMS

We feel the key to the design of distributed systems is to incorporate mechanisms which can deal with uncertainty and error as an integral part of their problem solving approach. Knowledge-based interpretation algorithms such as Hearsay-II [4] and MSYS [5] are examples of algorithmic structures that can resolve errors in this way. Problem solving in these systems involves the examination of many alternative partial solutions in order to construct a consistent overall solution. This style of problem solving is required because of uncertainty (error) in data and the incomplete, approximate, and inconsistent nature of knowledge used in these systems.

The exploration of alternative partial solutions takes the form of a search process in

which a solution is constructed through the incremental piecing together of mutually constraining or reinforcing partial solutions. These partial solutions arise both from application of diverse knowledge to the same aspects of the problem and from application of the same knowledge to diverse aspects of the problem. If sufficient constraints are available during this search process, incorrect data and decisions based on those data will naturally die out. In this way, uncertainty is resolved as an integral part of the problem solving process.

In many knowledge-based systems, the number of possible partial solutions is very large. In general, the more uncertainty that exists, the larger the number of alternatives that must be explored. If there exists a large amount of uncertainty in the data and (in the approximate nature of) knowledge, a significant amount of search is required. In order to limit the search, it is important to focus quickly on information which constrains the search space. Therefore, problem solving in these systems is often asynchronous and opportunistic: there is no a priori order for decision making, and decisions, if they look promising, are tentatively made with incomplete information and later re-evaluated in the light of new information.

Due to the asynchronous nature of processing and the existence of diverse and overlapping knowledge in these systems, a solution may be derivable in many different ways (i.e., different ordering sequences of incrementally constructed solution components and possibly different solution components).

Another focusing technique used in some knowledge-based systems is to structure the search space into a loose hierarchy of increasingly more abstract representations of the problem. Using this structure, a high-level partial solution developed in an opportunistic way can be used to constrain the search.

We feel this approach to problem solving provides a basis for the development of design methodologies for FA/C distributed systems. The mechanisms used in these problem solving systems to resolve error from incorrect and incomplete data and knowledge can also be used to structure distributed algorithms so that they will work effectively with incomplete and inconsistent local data bases. Let us examine these mechanisms from this viewpoint.

Asynchronous Nature of Information Gathering / Reduced Need for Synchronization -- Problem solving is viewed as an incremental, opportunistic, and asynchronous process. Because of this style of problem solving, a node does not have an a priori order for processing information and can exploit incomplete local information. Thus, the processing order within nodes, and the transmission of information among nodes, does not need to be synchronized.

Use of Abstract Information / Reduced Inter-Node Communication Bandwidth Requirements

-- The ability to use abstract information permits nodes to cooperate using messages having high information content. This reduces the inter-node communication bandwidth needed for effective cooperation.

Resolution of Uncertainty through Incremental Aggregation / Automatic Error Resolution

-- Errors and uncertainty are implicitly resolved when partial results are aggregated and compared with alternative partial solutions. This incremental method of problem solving allows a distributed system to detect and reduce the impact of incorrect decisions caused by incomplete and inconsistent local data bases and by hardware malfunction.

Problem Solving as a Search Process / Inter-Node Parallelism

-- Because many alternative partial solutions need to be examined, parallel search by different nodes is possible. Furthermore, the additional uncertainty caused by incomplete and inconsistent local data bases can be traded off against more search. To the degree that this extra search can be performed in parallel without proportionally more inter-node interaction, communication bandwidth can be lowered without significant degradation in network processing time.

Multiple Paths to Solution / Self-Correcting Behavior

-- Because there are many paths to a solution, it is possible to correct for what would be considered fatal errors in a normal distributed system. In addition, system reliability can be varied (at the cost of additional processing and inter-node communication) without modifying the basic problem solving structure. This variability is achieved through appropriate selection and focusing of local node activity.

Knowledge-based systems use a number of additional mechanisms to implement uncertainty resolution. These mechanisms are also important in the effective decomposition of data and function in a distributed system, and include: (1) integrated representation of alternative partial solutions, (2) data-directed control structures, (3) focus of attention strategies for dynamic allocation of resources, and (4) generator control structures which incrementally generate credibility-ordered alternative hypotheses. (A more detailed discussion of these points and of the appropriateness of knowledge-based AI as a basis for distributed problem solving systems is contained in a longer version of this paper [6].)

4. EXPERIMENTS IN FUNCTIONALLY-ACCURATE, COOPERATIVE DISTRIBUTED PROCESSING

In order to test the relevance of knowledge-based AI techniques to distributed problem solving, a number of distributed applications based on these techniques have been developed.

One such application is distributed signal interpretation. In this application the Hearsay-II architecture was used as the basic model for interpretation. The distributed interpretation architecture based on this model was structured as a network of Hearsay-II systems. Each Hearsay-II system (node) in the network had a limited view of the complete problem solving data base, a limited set of knowledge sources, and transmitted only a limited subset of its (partial) results to a limited set of the nodes.

An experiment using this distributed architecture indicates that the basic uncertainty resolving mechanism of the Hearsay-II architecture can also correct for error introduced by the use of incomplete and inconsistent local data bases. In addition, this mechanism can also handle errors resulting from communication loss [3].

A second application studied is distributed network traffic-light control. In this application, the distributed algorithm was based on a centralized, sequential relaxation (iterative refinement) algorithm called SIGOP-II [7]. SIGOP-II computes optimal traffic-light settings for a network of lighted intersections.

Experiments with a distributed version of this algorithm show that good, but not optimal, solutions can be generated. We have found it difficult to reproduce the performance of the sequential version without significant additional inter-node communication and synchronization. We attribute this difficulty to the strong non-local interaction among traffic-light settings [8]. In these experiments, the uncertainties introduced in the distributed version of SIGOP-II cannot be resolved completely by the distributed relaxation process.

The experiments with these two applications indicate that knowledge-based AI techniques are potentially useful in FA/C distributed systems. However, much work needs to be done to understand the appropriateness of a given knowledge-based AI technique to a particular distributed application.

5. CONCLUSION

We feel methodologies can be developed for functionally-accurate, cooperative (FA/C) distributed systems in which the distributed algorithms and control structures function with both inconsistent and incomplete data. These methodologies are necessary in order to extend the range of applications that can be effectively implemented in distributed environments.

FA/C problem solving structures are also important to the implementation of complex applications in centralized environments. These applications are often organized in the form of a collection of independent modules. In such a structure it can be conceptually difficult to develop and expensive to maintain

a complete and consistent centralized problem solving data base with which the modules interact. Techniques which permit relaxation of completeness and consistency requirements would be a significant aid in the development and maintenance of these (logically distributed) systems.

We feel there are two concepts that form the basis of FA/C distributed methodologies:

1. To view a FA/C distributed system as a network of cooperating systems which share common goals. Each system is able to perform significant local processing on incomplete and inconsistent non-local information.
2. To handle the uncertainty in control, data, and algorithms introduced by distribution as an integral part of the problem solving process.

Techniques developed in the context of knowledge-based AI systems provide a basis for implementing both concepts.

ACKNOWLEDGMENTS

The authors would like to acknowledge the research efforts of Lee Erman on distributed Hearsay-II and Richard Brooks on network traffic control which are discussed in this paper. We would also like to acknowledge the very useful criticisms received from Elliot Soloway.

REFERENCES

- [1] Peebles, R., and E. Manning. "System Architecture for Distributed Data Management." Computer. 11:1 (1978) 40-47.
- [2] Bernstein, P.A., et al. "The Concurrency Control Mechanism of SDD-1: A System for Distributed Databases (The Fully Redundant Case)." IEEE Transactions on Software Engineering. SE-4:3 (1978) 154-168.
- [3] Lesser, V.R., and L.D. Erman. "An Experiment in Distributed Interpretation." Technical Report CMU-CS-79-120, Computer Science Department, Carnegie-Mellon University, May 1979.
- [4] Erman, L.D., and V.R. Lesser. "The Hearsay-II System: A Tutorial." Chapter 16 in W.A. Lee (Ed.), Trends in Speech Recognition. Englewood Cliffs, N.J.: Prentice-Hall, 1979.
- [5] Barrow, H.G., and J.M. Tenenbaum. "MSYS: A System for Reasoning About Scenes." AI Center Technical Note 121, Stanford Research Institute, 1976.
- [6] Lesser, V.R., and D.D. Corkill. "The Application of Artificial Intelligence Techniques to Cooperative Distributed Processing." Technical report, COINS, University of Massachusetts, Amherst, February 1979.
- [7] Lieberman, E.B., and Woo. "SIGOP-II: A New Computer Program for Calculating Optimal Signal Timing Patterns." Transportation Research Record. Report 596, 1976.
- [8] Brooks, R.S., and V.R. Lesser. "Distributed Problem Solving Using Iterative Refinement." Technical report, COINS, University of Massachusetts, Amherst, May 1979.