AI Magazine Volume 8 Number 3 (1987) (© AAAI)

WORKSHOP REPORT

1986 Workshop on Distributed AI

N. S. Sridharan

This report contains a historical perspective on previous Distributed Artificial Intelligence Workshops, highlights of the roundtable discussions, and a collection of research abstracts submitted by the participants

Continuing the round of annual workshops dedicated to topics in distributed artificial intelligence (DAI), 25 researchers gathered at Twin Lights Manor, a pleasant country inn and seaside resort in Gloucester, Massachusetts, from 28-31 October 1986. Some of the participants characterized the workshop as "very intense" and "very productive" but pointed out the conviviality of the gathering and the pleasantness of the warm sun, miles of view, and the gentle sea breezes.

In response to requests from several participants, nearly half the time available was scheduled for open roundtable discussions. Presentations were limited to 13 thirty-minute talks. A short after-dinner session was held on Tuesday to establish topics for the roundtables. Two themes initially brought up were echoed throughout the rest of the workshop: (1) What are issues and concepts of DAI that distinguish it from AI and distributed computing? and (2) What are the differences and commonalities between viewing DAI as the synthesis of a single intelligent agent from distributed components and seeing it as the organization of multiple intelligent agents (committees? societies?)?

This report is organized into three parts. The first section presents a short historical perspective on the previous workshops. The second section contains highlights of the roundtable discussions. The report concludes with a collection of research abstracts submitted by participants.

Historical Perspective

A young discipline feels compelled to carry on a search for its identity. These researchers were all in the same quandary, asking questions such as "What exactly is DAI?" "What are the different camps in DAI?" "Why are we interested in DAI (isn't AI hard enough?)?" "What do we have as common concerns?" This quest for definitions has been characteristic of all the workshops since 1980.[1] I reproduce some of the earlier definitions from previous meeting reports to provide an idea of where we are now.

[1980] "Distributed AI is concerned with those problems for which a single problem solver, single machine, or single locus of computation seems inappropriate. Instead we turn to the use of multiple, distinct problem solvers each embodied in its own system."—*R. Davis.*

[1981] "DAI is concerned with problem-solving situations in which several agents cooperate to achieve a common set of objectives."—*P. Thorndyke, D McArthur, S Cammarata, and R Steeb.*

[1982] "A DAI system is a network of individual intelligent systems designed to cooperate in some way. Most work on DAI may be characterized as of this type. The alternative point of view is that a DAI system is composed of a large number of elements each of which is capable of a very limited amount of problem solving, and the intelligence of the overall system (its global coherence) is a result of the pattern of interaction among these `dumb' elements."—*M Fehling and L. Erman*

[1984] "DAI is concerned with cooperative solution of problems by a decentralized and loosely-coupled collection of knowledge sources, each embodied in a distinct processor node."—*R. G Smith.*

[1985] "DAI research has so far operated under a self-imposed set of blinders. Work has progressed on narrow issues of cooperation but the assumption of 'agent benevolence' has always been present.... We have to allow for true conflict, negotiation and compromise among intelligent agents." —*M. Genesereth, M. Ginsberg, and J. S. Rosenschein.*

[1986] "DAI is a continuum of study, both scientific research (reflective or examining) and engineering research (design-oriented), focused on how to understand and organize groups of intelligent problem-solvers. Groups of people, groups of automated intelligent processes, and human-computer interactive systems are all a part of DAI research. The best research in each of these areas tends to reinforce research in the others." —*L. Gasser*

In the 1986 workshop, we witnessed a stronger representation of parallel processing, emergent intelligence, and multiple expert system models contrasted with issues of interactions among intelligent agents. We were also a very critical and skeptical group insistent on defining more terms. Thus, the discussions were punctuated by questions such as "What do you mean by agent?" "What is coherence?" "What is cooperation?" "What makes an agent intelligent?" "How do you know when AI without distribution is inappropriate for the problem?" "How do you know you are not reinventing techniques in distributed computing or in AI?" "How do you measure the success of a DAI effort?" The report of the roundtable sessions goes into greater depth with some of these issues.

Another point of historical perspective is sociological. Only Victor Lesser (who described himself as the "granddaddy of DAI") has attended all the workshops in the series. There is no other overlap between those who attended the first workshop in 1980 and the seventh workshop in 1986 Similarly, none of the research projects other than that of Lesser and his colleagues has sustained over the years. It is unclear whether this lack of continuity is the result of the inherent difficulties and frustrations of DAI or external circumstances. What is evident, however, in thinking back over the previous workshops, is that there has been little passing on of tools, techniques, and methodologies. Because there is little published in the field of DAI that offers a good perspective, new research teams seem to ponder the same old issues over and over again.

Several Dimensions			
1. Individual	Committee	Society	MODEL
2. Fine	Medium	Coarse	GRAIN
3. Small	Medium	Large	SCALE
4. Learning	Adaptation	Programmability	
5. Controlled		Autonomous	AGENTS
6. Building by Synthesis			Building by Decomposition
7. Restricted		Ample	RESOURCES
8. Simple		Complex	INTERACTIONS

Table 1. A Dimensional Model for DAI.

Mike Huhns of Microelectronics and Computer Consortium (MCC) has edited a volume of selected articles for a book, *Distributed Artificial Intelligence,* to be published in 1987 by Pitman Publishing Limited (jointly with Morgan-Kaufmann) as part of the Research Notes in Artificial Intelligence series. However, this book contains no article explaining the perspective of DAI.[2] It is time that we collectively started some serious work in perspective setting. Two welcome indications of such a beginning were the intensive round-table discussions described in the next section and a few attempts to build computational test beds for some of the DAI research efforts.

An Eight-Dimensional Model for DAI

Table 1 lists eight dimensions we collectively identified that can help us see where a particular work, result, or claim fits. Seeing things in terms of this space can help identify what assumptions underlie the work. Each dimension forms a spectrum in itself and is not binary (yes-no) in character. The first dimension was mentioned earlier. Sometimes, we are aiming to get a large collection of intelligent agents to solve problems together, leading to a "society" model of computation. At the other extreme, we employ relatively simple computational elements to produce some intelligent behavior, as can be witnessed in connectionist models or neural networks. To drive home the point that this dimension is a spectrum, we can discuss a group of expert systems, each with some limited intelligence in its respective domain, attempting to solve a problem with cooperation and possibly some conflicts, thus forming a "committee."

The second dimension, "grain," refers to the level of decomposition for the problem (data, task, communication packets, and so on). Economic necessities probably do not permit coarse grain coupled with large scale. Fine grain coupled with small scale probably is too weak a combination to be of interest. Really, the two dimensions can vary independently.

Because distribution might or might not imply actual parallel processing using computing elements, a quite separate dimension is "scale," which accounts for the number of computing elements employed Use of a serial processor or a few (2-16) processors on a bus or ring network forms the small end of the spectrum. The use of a few hundred (100-10,000) processors lies in the middle. The use of the connection machine, with up to a million elements, forms the high end of this spectrum.

Some systems are built from "programmed" elements, whereas others lend themselves only to "learning" systems. In between lie systems where part of the organization, structure, or interaction patterns are fixed and programmed, but the system has aspects that are "adaptable." Once again, large scale and learning are heavily correlated, and small scale and programming seem correlated.

Another variable is the degree to

which each element seems "autonomous" (thus is capable of acting intelligently and rationally or is adaptive to global conditions) or each element seems fully "controlled" and devoid of volition. The reader might detect a possible correlation between autonomy, coarse grain, and a society model and between fully controlled, fine grain, and an individual model. However, these correlations seem accidental and extraneous. No compelling logical necessity exists for these correlations. It is conceivable to have central control in a large-scale, fine-grained system as well as distributed control in a small, coarse-grained system.

Some systems are built by carefully "decomposing" the problem into components and perhaps decomposing the components again. Some of the systems are built by the "synthesis" of existing elements. Methodologically, they impose different constraints on the system designer, and thus, the designer can face substantially different sets of design issues.

Resources available in the system and limits on their utilization form one of the most crucial concerns for the designer. Whether the resources (such as communication bandwidth, abstraction languages available, computation, memory, devices, knowledge or expertise, and so on) are ample or whether they are tightly limited affects the design and its effectiveness and can tilt the balance in favor of one design or another. For example, the work of Genesereth, Ginsberg, and Rosenchein (see earlier quotation from 1985) assumes highly impaired or actually nonexistent communication channels. Lesser assumes limited communication bandwidth but allows exchange not only of data from sensor net processing modes but also of tasks, goals, plans, and so on that are higher-level types of information. This dimension then actually represents several, based on each of the different resources one identifies.

Finally, we speak in terms of the simplicity or complexity of interactions among agents or elements. Neural nets and connectionist models tend to emphasize simple, as well as uniform, types of interactions. Autonomous agents collected in a contract net have relatively complex interactions.

One paradigmatic way to use this eight-dimensional space is to take sample claims (for example, greater distribution of control demands in order to achieve global coherence, exchange of tasks, and communication of goals and plans) and see in which regions of the space the claims seem to be valid. Performing this task would enable us to add qualifying clauses to the claims. Another way to use this model is to actively map current efforts into this space and then search for problems in a particular part of the space.

To indicate a research effort might be at opposing ends of one spectrum at the same time, one can look at the research being done on natural language processing at Bolt, Beraneck, and Newman (BBN). A system to support natural language interaction with a database is being designed. A top-level decomposition of a natural language question-answering system yields a syntax module that builds a parse tree, a semantic module which converts this tree into a logical query formalism, an interface that formulates this as a query to a specific database, and the database itself which calculates the answers. A window-based workstation takes care of the human-machine interaction. At this level of description, we get a coarse-grained, small-scale DAI system with relatively simple interactions. Yet, in an attempt to achieve speedup through parallelism, we have programmed the syntax module to yield medium-grained, medium-scale concurrency (capable of running on a Butterfly™* multiprocessor with up to 128 processors). If the same treatment is extended to the remaining modules, one could see the system as medium grained and medium scale but with global information sharing. Thus, it is important to realize that the dimensions are not attributes of systems per se but of systems considered at particular levels of description.

One example (see Parunak and and Kindrick's abstract) was described at this workshop where a problem was initially viewed as a committee of intelligent, autonomous agents. It was later discovered that a connectionist view was more favorable. This change underscores the fact that these attributions are merely useful points of view.

One other dimension comes up repeatedly—the issue of homogeneous, or a uniform set of, agents. Heterogeneous agents seem to complicate design, and homogeneity tends to simplify analysis or synthesis. Not enough experience exists, however, to know if this is a valid or useful dimension to add.

Highlights of the Roundtable Discussions

In the following subsections, the main focus of each of the six roundtable meetings is provided along with a summary of the discussion.

Roundtable 1

This roundtable was focused on some of the important dimensions mentioned earlier. It stated the grain size of distribution was an implementation issue, whereas the adoption of a society-committee-individual model was a conceptual matter; hence, these two dimensions are quite independent of each other. Similarly, the question of scale (counted in terms of the number of processors) was at the hardware level, regardless of what grain was used in designing the software.

With very small numbers of agents, one could have them be autonomous without encountering serious problems of global coherence. As we increase the scale, a form of organization is needed to coordinate the work of the agents. This organization typically takes a hierarchical form based on control, that is, task decomposition, allocation, and distribution.

Vic Lesser supports this view and claims even connectionists will have to resort to some organizational design for modeling higher intellectual activities. Peter de Jong claimed that when the scale gets really large, the organization will not be hierarchical but will be dynamic and approach open system structures. Dr. Charles Schmidt stated that coherence of overall activity is intimately related to the extent which limited local information can be used to test goal *satisfaction or program toward goals.* This discussion established the importance of global coherence when using multiple problem solvers.

Roundtable 2

The discussion was focused initially on clarifying terms we use often, as well as on lending them an accepted meaning, a fundamental aspect of the intelligent and informative exchange of ideas between researchers *Cooperation, coherence, local control* and *agent* were taken up as test cases. It was hoped that such a discussion would help us identify key research issues in DAI

Les Gasser referred to his presentation of multiagent interactions in an attempt to clarify cooperation. One action, A, can interact with another action, B, by facilitating the execution of B Action A achieves a result, G, that is a prerequisite for B. Depending on whether this was intended by the agent of A and whether the agent of B knew about it, different forms of cooperation can result. He further elaborated the point that because A could interfere with B through another result it achieves, the same pair (A,B) could result in both conflict and cooperation. Including the goal G in the discussion clearly illustrates whether two actions cooperate. Thus, the contextual factors are important in making such attribution. It was clear that a careful analysis of interaction patterns is needed to precisely define this term

Locality of control is intimately related to the distribution of control. Distribution of data, decision making, assumptions, and goals are all aspects to be considered. The role of asynchrony in computation was also mentioned as a central issue in DAI. Albert Boulanger gave an example of how a system can be synchronous at a detailed level of description and can still be asynchronous at an abstract level. The Butterfly system is synchronous at the clock level but is asynchronous at the Lisp task level because the programmer has no model to predict or control the time or duration of task execution.

Miro Benda attempted to focus the discussion away from terms such as cooperation or coherence toward making clear what measurements on the system constitute success. He offered the suggestion that if there is a measure, let us say machine IQ applicable to each agent Ai and the system S as a whole, then the equation IQ(S) > max IQ(Ai) might be interesting.

Roundtable 3

This discussion was on issues of communication between agents in DAI. Jagannathan wanted to know if there is a set of abstractions to describe communicative actions and intentions. An arbitrary set of example terms was listed: Inform, Request to do, Request to send, Command, Reply, Acknowledge and No-acknowledge, Offer, Agree, Refuse, Accept, Bid, and Propose. In describing communication in natural systems in terms of these abstractions (that is, in defining such terms), one would be looking at the effect of the communication on the sender and receiver. In a constructed system, communication is often tagged as one type or another, reflecting the interest of the sender. Norms that constrain the overall behavior of the system can then be used to set up expectations and also to monitor communication.

Resource limits (time limits, memory limits, and so on) are often communicated in constructed systems. It was pointed out that in such systems, in contrast to human interaction, very precise information can be communicated concerning goals, constraints, and problem-solving status.

Roundtable 4

In this session, Arvind Sathi presented three models of negotiation among agents taken from economics, behavioral science, and AI (Sathi, Morton, and Roth 1986). The discussion focused on questions of why and how to characterize organizations and how to found a concept of organization in the behavior of the intelligent agents that make them up. Why characterize organizations of problem solvers? The answer is to provide a basis for taking measurements for explanation and performance evaluation and to give parameters for designing them. One central point of this roundtable was that organizations can be characterized along many dimensions. Most DAI research has focused on communication structure or knowledge (skill and data) distribution. Sathi and

Where to Contact Authors of Abstracts*

M. Benda, V. Jagannathan, R. T. Dodhiawala, Boeing Advanced Technology Center BCS, P. O. Box 24346, MS 7L-64 Seattle, WA 98124 Net mail: juggy@boeing.com

Roberto Bisiani, Computer Science Department, Carnegie-Mellon University, Schenley Park, Pittsburgh, PA 15213

Albert Boulanger, BBN Laboratories Inc., 10 Moulton Street, Cambridge, MA 02238.

Susan E. Conry, Electrical and Computer Engineering Department, Clarkson University, Potsdam, NY 13676.

Peter de Jong, The Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139

Les Gasser, Computer Science Department, University of Southern California, Los Angeles, CA 90089-0782

Victor Lesser, Edmund H. Durfee, and Daniel D. Corkill, Department of Computer and Information Science, University of Massachusetts, Amherst, MA 01003.

H. Van Dyke Parunak, and James Kindrick, Industrial Technology Institute, P. O. Box 1485, Ann Arbor, MI 48106

James P. Rice, Knowledge Systems Laboratory, Stanford University, 701 Welch Road, Bldg. C, Palo Alto, CA 94034

Arvind Sathi, Carnegie Group Inc., 650 Commerce Court, 6th Floor, Pittsburgh, PA 15219-1119.

John L. Goodson, RCA Advanced Technology Laboratory.

Charles F. Schmidt, Laboratory for Computer Science Research, Rutgers University, New Brunswick, NJ 08903.

* Listed in the order abstracts appear.

Gasser suggested that there are many more useful dimensions for characterizing organizations, including distribution of resources; perception; knowledge; and structures of communication, tasks, power, and skills. Another, problematic dimension was qualitative measures of actions and their outcomes. Ed Durfee defined the term organization as a system of constraints on individual behavior. Constraining an agent on any of these dimensions in a different way can lead to different performance. Because organizations of intelligent agents might need to be adaptive during problem solving, it is important to know how to vary particular dimensions in particular circumstances. This need is the problem of organizational self-design, a fundamental one in DAI. It also appears that DAI might provide a richer experimental domain for organizational self-design experiments because more variables are controllable than in human organizations.

Roundtable 5

The central question for this roundtable was how one evaluates success in a DAI effort. This matter is related to *metrics* on the system. By showing that the distributed approach gains significantly on some useful measure, one hopes to argue for or justify this approach. Presumably, this is one way of responding to the question, "why is the D in DAI?"

Several measures, or attributes, were considered during the discussion.

Speedup. Clearly, the promise of speedup is one of the main attractions in taking on a distributed approach, even if not all work in DAI is aimed this way. It is also known or believed that the burden of communication and the added effort this burden implies and the sequentiality dictated by it stand in the way of achieving the maximum theoretically expected speedup.

Reliability. Distributed systems make possible continued operation in the presence of failures, faults, or the dysfunction of components. Of course, the processing ought to have been designed with the issues of reliable operation in mind. Distributed systems only reduce the probability of a failure; they inherently do not guarantee fault-tolerance.

Quality of Solution. For those problems where it is sensible to talk in terms of the quality of the solution, one might expect to show that DAI provides a better-quality solution for the same effort (measured in time, cycles of processing, hypotheses considered, and so on).

Effort or Efficiency. For other problems, one might attempt to show that the total effort in a distributed approach required to arrive at the same answer is less than when using a conventional approach.

Utilization. The relative utilization of different resources (computing elements, communication channels, and so on) can also be used to evaluate a DAI system.

Clarity. For those researchers who approach DAI for its conceptual advantages (for example, modularity) and who are not necessarily seeking a distributed implementation, some nonmetrical notions of clear conceptual structure are probably satisfactory.

Fit to Problem. For those who view their problem to be inherently distributed, a DAI solution probably seems natural. Specifically, the ability to architect the system to reflect their own abstraction and conceptualization, might lead them to trust the correctness of the system.

Cost. For some researchers, the economics of the situation probably provides the drive. Using some cost measure based on current technology, they might wish to justify DAI in terms of providing an economical solution under suitable trade-offs.

Do We Need to Justify DAI? For some of us, like me, there is hardly any need to justify taking a distributed approach. There are many reasons why a distributed approach is absolutely essential, and is unavoidable. To quote Nils Nilsson, "Work in DAI will contribute to (and may even be a prerequisite for) progress in ordinary artificial intelligence" (see Davis 1980, p. 43) Let us see what some of these reasons are.

Reliability This was mentioned earlier. The prominent route to reliability is through redundancy, sometimes by duplication or replication, but for AI it is also by multiple expertise or multiple solution strategies. If reliability is key, then DAI is the way to follow.

Person-Machine Systems. In building an interactive AI system, the system ought to be designed using the principles of DAI. Schmidt and Goodson outlined in their presentation some axiomatic norms that govern the exchange of information (facts, hypotheses, and goals). Further studies are needed to develop norms for exchange of tasks and other kinds of information. The presence of the human agent certainly makes the system distributed, and one cannot forever ignore this aspect.

Autonomous Agents. Much of the work in AI assumes that the intelligent system being designed is the only intelligent agent in its universe. This assumption shows up glaringly if one examines the typical statement of a frame axiom for a robot: "The only changes in the world are the result of the action of the robot." In the presence of multiple autonomous agents, a certain amount of allowance must be given to the goals, actions, and plans of the other agents. This allowance necessitates the ability to model the belief and knowledge structure of other agents. Those persons who study multiple autonomous agents in a single environment do not sit and ponder the need for DAI research!

Human Organizations. The introduction of a centralized data processing or MIS function in organizations is giving way to a distributed function. The centralization was an economic accident resulting from the large data processing costs. The availability of lower-priced workstations allows the organization to respond to the desire for distribution and autonomy that is present in every organization. DAI is an inevitable consequence of this desire.

The Future is Here and Now. Some

even argue that the world is a distributed workplace. The future certainly lies in greater distribution of information and computing resources. The future of computing is parallel processing. The presence of serial computing was only a passing phase, a mere flicker in the history of computing. Thus, parallel processing and DAI have intrinsic interest for the future.

Incremental Aggregation. A DAI framework, especially with multiple interacting expert systems, permits the development of systems by incremental aggregation. Serial solutions, especially control-flow statements of these solutions, do not have any interesting property of additivity. Although no DAI framework exists with properties of additivity, it is easier to see that DAI offers this promise more than ordinary AI.

Roundtable 6 A Wrap-Up Session

A quick survey of funding sources revealed the following scattering of sources: SDF, Boeing internal, Kellogg Foundation (ITI), the U. S. Navy, RCA, NSF, GTE internal, DARPA, ONR/URI, the University of California, the U. S. Air Force, MCC, NCR, the U. S. Army, RADC, and NASA. On behalf of the DAI community, I appeal to these funding agencies for one or a few to steal the opportunity and establish a firm and a continuing program of research in this most crucial AI area. I am sure many of the workshop participants will be happy to serve if they are called upon to set up priorities and guidelines for research in DAI. In summary, we agreed that the following issues are among the central concerns for DAI research: the effect of asynchrony, the effective use of local control, the use of local evaluation of progress, and the building up of abstractions of control knowledge.

The 1987 workshop will be convened by Professor Les Gasser, Computer Science Department (SAL-200), University of Southern California, Los Angeles, CA 90089-0782. Those persons wishing to receive invitations can inquire by telephone, (213) 743-7794, or net mail to Arpanet, Gasser@USC-CSE.USC.EDU.

Collection of Research Abstracts[3]

On Optimal Cooperation of Knowledge Sources: An Empirical Investigation

M. Benda, V. Jagannathan, and R. T. Dodhiawala, Boeing Advanced Technology Center

We seek to investigate the issues involved in building large, possibly distributed, knowledge systems which have multiple knowledge sources cooperating to produce a solution. Some of the considerations that go into building such systems are the organizational structure of the knowledge sources, their mode of cooperation, and the communication cost associated with the organizational structures. In a multi-agent environment it is important to consider the amount of "intelligence" associated with each agent.

We define three primitive modes of cooperation which form the basis for organizing the knowledge sources (agents). Using these primitive modes, an algorithmic approach to constructing organizations is used to generate a reasonable set of candidate organizations. A set of nine organizations was generated in the case of four agents. A problem task was designed which allowed us to experiment with different organizational structures, and to study the efficiency and cost associated with these. The task chosen was such that it was necessary for the agents to cooperate in order to successfully achieve the solution. With the performance metrics chosen, the results of our experiments show why some organizations did better than others at the particular task, and the overhead of different communication and transaction costs that are inherent in a given organizational structure. A blackboard control model was chosen for our investigations.

Agora

Roberto Bisiani, Carnegie-Mellon University

Agora is an environment that addresses the problem of supporting the design and implementation of heterogeneous systems on multiprocessors. Agora supports heterogeneous systems by providing a parallel virtual machine that is independent of any language, allows a number of different programming models and can be efficiently mapped into a number of different computer architectures. Rapid evolution is supported by providing incremental programming capabilities similar to those found in Lisp environments. Programs that run on the parallel virtual machine can be added to the environment and share the same data with programs that were designed independently This makes it possible to provide an unlimited set of custom environments that are tailored to the needs of a user, including environments in which parallel processing has been hidden from the end user. Finally, parallelism is strongly encouraged since systems are always specified as parallel computations even if they will be run on a single processor.

Agora is not an "environment in search of an application" but is "driven" by the requirements posed by the design and implementation of the Carnegie Mellon University distributed speech recognition system. During the past year, we designed and implemented an initial version of Agora and successfully used it to build two prototype speech-recognition systems. Our experience with this initial version of Agora convinced us that, when building parallel systems, the effort invested in obtaining a quality software environment pays off manifold in productivity.

Agora has reduced the time to assemble a complex parallel system and run it on a multiprocessor from more than six man-months to about one man-month. The main reason for this lies in the fact that the details of communication and control have been taken care of by Agora. Application research, however, calls for still greater improvement. Significant progress in evaluating parallel task decompositions, in CMU's continuous speech project, for example, will ultimately require a single person to assemble and run a complete system within one day.

Parallelism in the Execution of a Routine Knowledge Rule System on the Butterfly Computer

Albert Boulanger, BBN Laboratories Inc.

This project ported a routine knowledge rule system to the Butterfly multiprocessor. The goal was to explore parallelization techniques with an existing rule system originally written for serial execution. The rule system was rewritten to introduce parallelism, and run on a single processor to establish a benchmark for serial operation. The same version, with parallelism enabled, was then run on a 16-node Butterfly multiprocessor. The metering tools on the Butterfly were used to display task behavior and processor utilization. The information gained from these displays was used to guide further experimentation with the granularity of the rules and with the system code to investigate bottlenecks that were lengthening execution time.

The project demonstrated that parallelization of routine knowledge rule systems can yield substantial speedup. It also demonstrated that the metering tools on the Butterfly can be used to achieve additional speedup of parallel implementations. The implications of this research were discussed and compared to the findings of research at Carnegie-Mellon University on parallelizing production systems.

DAI at Clarkson University

Susan E. Conry, Clarkson University

The domain of interest in our research involves the monitoring and control of large communications systems. In such a system there is no single place at which knowledge about the system's operating state can be accurately maintained and therefore there is no single locus of control that can be relied upon. For these reasons, we have assumed that both the monitoring function and the control functions are decentralized.

The overall goal of problem solving in this domain is one of maintaining user to user service under changing traffic conditions, user requirements, and system disturbances. Distributed problem solving in this environment involves at least three major tasks: a distributed assessment task (to assess the impact of various disturbances), a distributed diagnosis task (to ascertain the source of an outage), and a distributed planning task (to determine plans for reconfiguring the network in the event of outage). These tasks are relatively independent of one another, yet each must be accomplished in a cooperative fashion by agents that are geographically distributed.

We are currently developing a testbed for investigating cooperative problem solving in this domain A shell system which can be used as a vehicle for the development of distributed problem solving systems has been implemented, and work is in progress on the planning problem and the diagnosis problem. A model for multi-stage negotiation useful in distributed planning has been formulated and a planner based on this model is being implemented. Domain knowledge relevant to the diagnosis task is being assimilated and mechanisms for inferencing to accomplish a reasonable diagnosis when the requisite knowledge is distributed are being investigated.

We believe that research in this domain is of particular relevance to DAI for several reasons. First, the domain provides a natural vehicle for examining issues that arise in the context of different problems all within the same framework. Thus it permits investigation of the degree to which problem characteristics affect the efficacy of a cooperation paradigm In addition, the ways in which nonhomogeneous distributed problem solvers interact in a DAI system can be readily investigated in this domain. Finally, we believe that this domain has characteristics that make it a natural model for a number of other types of problems, so that results obtained with respect to this domain will find application in a wide variety of problem solving situations.

Distribution and Convergence in UBIK

Peter de Jong, Massachusetts Institute of Technology

Ubik is a system in which organizational knowledge and action are represented. The knowledge is used to facilitate the development of new organizational applications and execution of existing applications. As Ubik is executing its applications, it will develop new concepts and reorganize existing concepts to provide a better match between its description of the organizational action and the actual interactive execution of the applications.

The distribution within an organization results from the parallel activities of distributed agents These agents develop their own models of the organization's structure and action. Each model is only a partial description of the organization. The concept of a central organization is emergent out of the multiple, distributed, partial models. Within Ubik, models can be distributed in physically separate databases. A model consists of concepts connected together in conceptual nets A concept consists of a name and attributes A conceptual net is a collection of concepts connected together using variables. Actions within a model are specified by messages sent to handlers. The handlers are attributes within a concept. When a handler receives a message, a conceptual net within the handler specifies the action which is to be performed.

Convergence of description within Ubik is accomplished by the calculation of prototypes A prototype is a summary representation of a collection of individual concepts. Many different prototypes can be constructed for a concept. For example, an employee prototype can be for a particular company, all manufacturing companies, all service companies, or all Massachusetts companies. Prototypes provide defaults for creating new individual concepts. The use of prototypes imposes a uniformity on the creation of new concepts. Before adequate prototypes are calculated, Ubik must reason over multiple, similar, individual concepts. This reasoning is done using goals which match generalized conceptual patterns. These goals are distributed between models using special messages called tapeworms.

Convergence of action within Ubik

is accomplished by the sending of messages between models. In order for a message to be understood by a receiving model, the model would have to have appropriate concepts and conceptual nets. The sender of a message must induce the receiver of the message to construct these concepts and conceptual nets. It is not enough for the sender to send the concepts needed because the receiver will not know how to relate the new concept to its existing concepts. The sender must have a model of the concepts the receiving model already knows, and teach the new concepts in terms of the concepts they have in common. The sender only has to teach the receiver enough to carry out the requested action. Each participant has different models of the organization appropriate to its organizational role. The object is not to evolve all the models into a common model, but only to coordinate the models where necessary. The separate models provide multiple viewpoints of the organization and its environment. These multiple viewpoints provide the organization with the flexibility necessary to react to new situations, and evolve more effective descriptions and actions which match the organization to its environment.

On the Nature of Multi-Agent Systems

Les Gasser, University of Southern California

Up to now in DAI there has been some successful experimentation in systems which address single-domain problems or single-paradigm architectures, exploring questions such as global coherence, distributed control, and resource allocation. There is the start of a formal basis for representing action, time, belief, and rationality in single agents, so that they can act with purpose in a multi-agent world. But it seems to me that there is still missing a set of concepts which will philosophically and practically unify the concepts of action, representation, and reasoning in individual agents with a conception of multi-agent aggregates so that the organized actions of the multitude are grounded in the individual actions of participants, and so that the actions of individuals are appropriately constrained and aligned with the general frameworks or policies for action held in the aggregate. Historically, this is the problem of "social order." (Its manifestations appear in prior DAI work in the questions of global coherence and distributed control, etc.)

Programming languages and descriptive systems, while they have provided us with primitive mechanisms for asserting synchronization, concurrency, belief, action, etc. have left the burden of establishing "global order" among concurrent processes to programmers, or else have been overly restrictive (e.g. asserting rigid control relationships). We need to move "up" a level and address not just how to represent arbitrary actions or beliefs, but rather to ask what activities occur, what is believed, and so on in any interactional setting. What we need is a basic theory of interaction and social organization.

We propose here to address these deficiencies in a conceptual way, laying the groundwork for experimental and theoretical investigations into multi-agent systems. As in other AI research, we take as our goal understanding and emulation of human production and problem-solving activity, but our focus is on the nature of activity as it is carried out in aggregates, rather than by single individuals alone. This means of course focusing on the dual nature of multi-agent systems—the roots of aggregate behavior found in the individual, and the roots of individual action found in the aggregate.

Current DAI Research at the University of Massachusetts

Victor R. Lesser, Edmund H. Durfee, and Daniel D. Corkill, University of Massachusetts

Our research stresses the importance of sophisticated local control in cooperating agents that individually solve complex subproblems of a distributed problem. Although each agent is an independent problem solver, the distribution of problem information and problem solving knowledge among agents requires them to exchange partial problem solutions so that the network can converge on an overall solution. Because their subproblem solutions must be compatible, the agents are solving interacting subproblems and must coordinate their activities to be an effective team. However, the loosely-coupled agents may have different local views of the problem and of network activity. We have been developing mechanisms that allow an agent to use its local view of itself and other agents to make problem solving and communication decisions that contribute to network problem solving.

In our previous work, we developed an organizational approach to distributed problem solving where each agent has knowledge about its general problem solving interests and those of the agents it can communicate with (Lesser and Corkill 1983). An appropriate organizational structure can improve overall network problem solving by biasing each agent's decision about which activities to pursue next. The organization is a general, long-term framework for acceptable network coordination in a variety of problem solving situations, but, to be a more effective team in a given short-term situation, modes need to dynamically refine their views of network activity.

We have given agents the ability to plan their activities and to exchange these plans to recognize how best to cooperate in a particular situation. Initially, we implemented and evaluated (as part of our experimental testbed) planning and communication mechanisms that could substantially improve network performance in a limited range of distributed problem solving situations (Durfee, Lesser, and Corkill 1985). In our current work, we have developed a more sophisticated planner that improves an agent's view of its current and possible future activities (Durfee and Lesser 1986). We are working on ways in which agents can incrementally exchange and individually integrate their local plans into partial global plans, which are then used to modify local decisions based on predicted network activity. Besides studying how this form of distributed planning improves coordination, we are also investigating issues in organizational self-design, distributed diagnosis of incorrect network problem solving behavior, and how parallelism can be introduced into the problem solving and control activities of an individual agent. Our goal is to develop intelligent agents that have sufficient awareness of their local and network activities to individually recognize and choose actions that lead to effective cooperation.

A Connectionist Model for Material Handling

H. Van Dyke Parunak and James Kindrick, Industrial Technology Institute

The research we reported at the workshop addresses the problem of material handling (specifically transportation and storage) in a facility manufacturing discrete parts. A distributed architecture is promising in such an application because it offers robustness to local failure and a good match to the modular material handling hardware that is becoming increasingly popular. The problems to be addressed in this domain include routing of materials from one workstation to the next, maintaining levels of work-in-process inventory (WIP) high enough to keep workstations busy but low enough to avoid excess carrying costs, and avoiding congestion on the material handling hardware. Parunak, White, et al. (1986) outline the overall control architecture within which this application arises.

Our original architecture (Parunak, Lozo, et al. 1986) models each physical transportation unit as a "Mover," an intelligent agent that monitors the contents of a single random-access storage or transport unit (such as an automatic storage and retrieval unit or a loop in a conveyor system) and negotiates with its neighbors to acquire needed containers of parts and dispose of excess containers. The central data structure of each Mover is a "table of Mover parameters" or TOMP that records maximum and minimum capacity levels for each type of container a Mover can handle. Local propagation of constraints among Movers guides the system toward the desired global distribution of material. We call the system "CASCADE" because it operates by cascading parts locally from one entity to another.

As we worked with a prototype implementation, we realized that the individual records in the TOMP, one for each pallet type, were more natural basic units of computation than the Movers themselves. In our revised architecture, each such record (now itself called a "TOMP") is a full-fledged object that manages the population of a single pallet type on a single transport loop. Movers still exist, but only as an interface between the TOMPs on a single transport loop and that loop itself. TOMPs communicate among themselves with two elementary messages: requests and deliveries of pallets. Each TOMP deals only in pallets of a single type. Related objects, called "Processes," transform pallets from one type to another, and interact with TOMPs just as other TOMPs do. Just as TOMPs are aggregated into Movers, corresponding to physical transport equipment, Processes are aggregated into Workstations, corresponding to physical production machinery.

The new architecture has far more objects than the old, but the messages among objects are simpler. In fact, we show in the full paper that the network of TOMPs and Processes is isomorphic to a neural network as formalized in the PDP model of Rumelhart, Hinton, and McClelland (1986). That model has eight elements, all of which may be identified in CASCADE:

1. The TOMPs and Processes correspond to the processing units, the neurons of a neural net.
2. The population of containers in a TOMP corresponds to the activation level of a single processing unit.
3. The filling and spilling rules for TOMPs correspond to the output functions for neurons, with the TOMP capacities corresponding to neural thresholds.
4. Like neurons, TOMPs have a specific pattern of connectivity.
5. In PDP, a propagation rule computes the input to the processing units from their outputs and the connectivity. In CASCADE, this rule models the delay resulting from physical transfer of pallets from one place to another.
6. The activation rule for TOMPs, as for many neural models, is a simple summing of inputs with existing populations.
7. The learning mechanism for CASCADE includes changes not only to the connectivity matrix, as in PDP models, but also to the node thresholds, as in some other neural models (Bienenstock, Cooper, and Munro 1982).
8. The environment of the network consists of the Processes, whose activity is scheduled by mechanisms external to CASCADE and therefore stochastic from its perspective.

The project thus far suggests two insights about DAI in general and connectionist architectures in particular. (1) The Pocket Knife Principle asserts that a good tool is useful in situations for which it was not originally designed. Neural nets were invented to model cognitive processes with an architecture similar to that used in nature, not to solve complex engineering problems like material management that tend to be too complicated for raw human intelligence. The usefulness of these models in such noncognitive domains is thus worth noting. Our application has much in common with the use of these networks to solve the traveling salesman problem in Hopfield and Tank (1985).

(2) In the Marching Band Syndrome, when large numbers of intelligent agents are closely coupled, they behave as though they were simpler than they actually are, and may be described using models that would be inadequate to represent their activity in less-structured environments. The shift in CASCADE's architecture from a smaller number of relatively complex agents with individual intelligence to a large number of simple agents with emergent collective intelligence is an example of this syndrome.

This research was sponsored by a grant from the Kellogg Foundation to the Industrial Technology Institute.

Advanced Architectures Project

James P. Rice, Stanford University

This is a DARPA funded project whose prime objective is to achieve 2-3 orders of magnitude in overall speedup for expert systems applications through the exploitation of parallelism. This project involves the development of a thin slice of system from the possible domains of system components to address the following issues. Hardware Design, Operating Systems, Languages, Resource Allocation, Knowledge Retrieval/Management, Problem Solving Frameworks and Applications. At present, work is concentrating on (1) Applications: Two applications are being developed in the field of real-time signal understanding. (2) Problem-Solving: Blackboard systems are being investigated for this purpose. Two such systems are under development and the applications are being mounted using them. (See Cage and Poligon below) (3) Languages/Operating Systems: An object oriented metaphor is being used Lamina (see below) is being used in this context and applications are being mounted directly in Lamina. (4) Hardware Design. A distributed memory, dynamic routing, message passing architecture has been chosen and is being simulated.

The CARE Simulator. The CARE simulator seems to be working quite well. We can now simulate machines with up to 256 processors. At present we have no problems with more parallelism than this, at least that we can extract.

We have found it to be very difficult to debug programs, even on our simulation of this distributed machine. The simulation is also very slow (2-3 orders of magnitude slower than an equivalent program running serially).

The Lamina Project. Lamina is an object oriented programming extension to Lisp which is designed to operate in a distributed memory multiprocessor environment. Lamina allows the easy management of processes and of remote objects and their intercommunication. Easy access is given to the streams through which processes can communicate. This makes it particularly easy to exploit pipeline parallelism. Special provision is made in Lamina to obviate delays due to process switching. Also, futures and similar data structures can be implemented by the use of the stream mechanism mentioned above Finally, provision is made for assisting in resource allocation and load balancing.

Two applications are currently being mounted on it. Work on one of the applications, which is largely complete, has shown up some problems, which are being addressed now by some major revisions. No speedup figures are available yet but good performance has been achieved with respect to load tolerance. The revisions are expected to deliver better load tolerance and performance.

The CAGE Project. The CAGE (Concurrent AGE) project is not the main stream of the AAP, since it is a version of AGE extended with constructs for the exploitation of concurrency which is targeted at shared memory and hardware. Parallelism can be exploited at the knowledge source, rule implementation. A high level language has been developed for the expression of Rules in CAGE and the parallelism in them. This language allows considerable sharing of application source code with the Poligon system (see below).

CAGE is operational, but undergoing development in a mode that uses a QLambda emulator. This is unable to deliver performance figures so the CARE simulator is being modified to allow the simulation of shared memory machines Two applications are now being implemented in CAGE.

The Poligon Project . Poligon is a high level language and system for the implementation of Blackboard like AI applications on distributed memory machines. Some of its unusual features are as follows: Poligon has no central control, scheduling or global data. The user program expresses which parts to execute serially, not those which are to be executed in parallel. Extensive use of futures is made automatically (not user defined) Unordered data structures, such as bags, are used to prevent blocking Compile-time strictness analysis allows blocking to happen as late as possible. Mechanisms are provided to help maintain consistency Real-time performance is assisted by automatic timestamping and ordering for user data. Sophisticated debugging tools are provided.

Like CAGE, Poligon is having two applications mounted on it. Because Poligon has a mode in which it simulates the semantics of the full parallel program, but in a serial mode, the applications are in a greater state of development than CAGE, though they have still not been developed far enough to get any performance figures.

Constraint Directed Negotiation Among Organizational Entities: An Alternative Model of Project Management

Arvind Sathi, Carnegie Group Incorporated

Large engineering projects involve a number of activities and cooperation across a large number of departments. The complexity of interdependence and uncertainty in market and technology make management of change a primary project management task, unfortunately ignored by the existing models of project management. We present an alternate model of change management using constraints and negotiation on constraints. The model provides insights on organizational distribution of goals, as when asked about the tools for project management

A Distributed Problem Solving Approach to Person-Machine Interaction

John L. Goodson, RCA Advanced Technology Laboratory, and Charles F. Schmidt, Rutgers University

The problem of how to achieve cooperation among multiple agents working on a common problem asynchronously and without benefit of centralized executive control is considered Cooperative problem solving among multiple agents requires that the communication from one agent to another be relevant to the recipient's problem solving activity. A communication is said to be relevant if the message is both interpretable by the recipient and if it is effective. A message is interpretable if the recipient can establish referents for the terms and relations mentioned in the message. A message is effective if the recipient's problem solving activity is altered as a result of the interpretation given to the message.

In this type of distributed problem

solving environment, there is no way in which to guarantee that messages will be relevant. Cooperative problem solving among human agents presupposes the ability to recognize whether or not communication is relevant Further, if the communication is recognized to be irrelevant, then there must exist a mechanism to alter the communication among the cooperating agents.

It is proposed that norms, conventions, and default theories concerning other agents beliefs provide the mechanism for establishing and modifying human cooperative interactions. Conventions are used to insure that the interpretability requirement is met. Norms are developed as rules which specify when communication to another agent is appropriate and what should be communicated Thus, norms provide a way to specify what may constitute an effective communication. Normative rules typically specify when communication should occur with reference to some aspects of the problem solving states of communicator and recipient rather than with reference to an objective frame such as time. This subjective frame of reference necessitates either the creation of additional norms to insure that the information concerning the beliefs of the other are available, or the use of default theories which allow for the prediction of the beliefs of the other. An interlocking set of such conventions, norms, and default theories gives rise to a particular pattern of roles that define the specific type of cooperative problem solving realized by a particular set of agents.

These ideas are illustrated within a particular complex man-machine problem solving system. In the design and development of this system, machine agents were created to improve the reliability of human problem solving on several interpretation tasks. Various patterns of cooperation were specified by assigning a set of norms, conventions, and default theories to the agents. These various patterns of cooperation were then evaluated with respect to how well they enhanced problem solving performance and with respect to the resources required of the agents to maintain the pattern of obligations specified by the norms placed on each agent.

Notes

1 Previous meetings that have been reported:

No. Report	Place/Year
1 MIT/1980	SIGART Newsletter No 73 (Oct 1980)
2 MIT/1981	SIGART Newsletter No. 80 (April 1982)
3 USC-ISI/1982	SIGART Newsletter No. 84 (April 1983)
4 U. Màss/1983	Not reported
5 Schlumberger/1984	*AI Magazine* (Fall 1985)
6 Sea Ranch, CA/1985	*AI Magazine* (Summer 1987)

2. A second volume is planned Prospective authors should contact Mike Huhns, MCC, 3500 West Balcones Dr., Austin, TX 78759 Telephone: (512) 338-3651. Arpanet: Huhns@MCC

3. See side box for mailing addresses where authors can be contacted.

*Butterfly is a trademark of BBN Laboratories Inc

References

Bienenstock, E L; Cooper, L. N; and Munro, P. W 1982 Theory for the Development of Neuron Activity: Orientation Specificity and Binocular Interaction in Visual Cortex. *Journal of Neuroscience* 2:32-48

Davis, R. 1980 Report on the Workshop on Distributed AI *Sigart Newsletter* 73:42-52.

Durfee, E H., and Lesser, V R 1986 Incremental Planning to Control a Blackboard-Based Problem Solver. In Proceedings of the Fifth National Conference on Artificial Intelligence, 58-64 Menlo Park, Calif.: American Association for Artificial Intelligence.

Durfee, E H; Lesser, V R.; and Corkill, D D 1985. Coherent Cooperation Among Communicating Problem Solvers, Technical Report, 85-15, Dept. of Computer and Information Science, Univ of Massachusetts

Hopfield, J J, and Tank, D. W 1985 "Neural" Computation of Decisions in Optimization Problems *Biological Cybernetics* 52: 141-152

Lesser, V R., and Corkill, D D 1983 The Distributed Vehicle Monitoring Testbed: A Tool for Investigating Distributed Problem Solving Networks. *AI Magazine* 4(3):15-33

Parunak, H V. D; Lozo, P W; Judd, R.; Irish, B W.; and Kindrick, J. 1986 A Distributed Heuristic Strategy for Material Transportation In Proceedings of the 1986 Conference on Intelligent Systems and Machines

Parunak, H V D; White, J F; Lozo, P. W; Judd, R; Irish, B. W; and Kindrick, J. 1986 An Architecture for Heuristic Factory Control In Proceedings of the 1986 American Control Conference

Rumelhart, D E; Hinton, G E; and McClelland, J L 1986. A General Framework for Parallel Distributed Processing. In *Parallel Distributed Processing*, eds. D E. Rumelhart and J L McClelland, 1 45-1 76 Cambridge, Mass.: MIT Press

Sathi, A.; Morton, T E.; and Roth, S. F. 1986 Callisto: An Intelligent Project Management System. *AI Magazine* (7)5:34-52.