Use of Meta-Level Control for Coordination in a Distributed Problem-Solving Network

Daniel D. Corkill and Victor R. Lesser

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

August 1983

This paper was presented at IJCAI-83.

Abstract

Distributed problem-solving networks provide an interesting application area for meta-level control through the use of organizational structuring. We describe a decentralized approach to network coordination that relies on each node making sophisticated local decisions that balance its own perceptions of appropriate problem-solving activity with activities deemed important by other nodes. Each node is guided by a high-level strategic plan for cooperation among the nodes in the network. The high-level strategic plan, which is a form of meta-level control, is represented as a network organizational structure that specifies in a general way the information and control relationships among the nodes. An implementation of these ideas is briefly described along with the results of preliminary experiments with various network problem-solving strategies specified via organizational structuring. In addition to its application to Distributed Artificial Intelligence, this research has implications for organizing and controlling complex knowledge-based systems that involve semi-autonomous problem solving agents.

1 Introduction

Cooperative, distributed problem-solving systems are distributed networks of semi-autonomous processing *nodes* that work together to solve a *single* problem. Each node is a sophisticated problem-solving system that can modify its behavior as circumstances change and plan its own communication and cooperation strategies with other nodes. Our research has emphasized applications where there is a natural spatial distribution of information and processing requirements among the nodes but insufficient local information for any node to make completely accurate processing and control decisions without interacting with other nodes. An example of this type of application is a distributed sensor network [1, 2, 3, 4]. Our approach for implementing these applications is to have the nodes cooperate via an iterative, coroutine exchange of partial and tentative high-level results. In this way, the system as a whole can

This research was sponsored in part by the National Science Foundation under Grant MCS-8006327 and by the Defense Advanced Research Projects Agency (DOD), monitored by the Office of Naval Research under Contract NR049-041.

function effectively even though the nodes initially have inconsistent and incomplete views of the information used in their computations [5, 6, 7, 8, 9].

A key problem in cooperative, distributed problem-solving networks is obtaining sufficient global coherence for effective cooperation among the nodes [10]. If this coherence is not achieved, then the performance (speed and accuracy) of the network can be significantly diminished as a result of:

- lost processing as nodes wait for something to do;
- wasted processing as nodes work at cross-purposes with one another;
- redundantly applied processing as nodes duplicate efforts;
- mis-allocation of activities so that important portions of the problem are either inaccurately solved or not solved in timely fashion.

These problems have been observed in our experiments with three-to-five node networks [5, 7]. We expect these problems will become even more significant as we move to networks containing larger numbers of nodes operating in changing environments.

In this paper we describe a decentralized approach to network coordination that relies on each node making sophisticated local decisions that balance its own perceptions of appropriate problem-solving activity with activities deemed important by other nodes. Each node is guided by a high-level strategic plan for cooperation among the nodes in the network. This strategic plan, which is a form of meta-level control, is represented as a network organizational structure that specifies in a general way the information and control relationships among the nodes.

In the next section we expand on the use of organizational structuring as a meta-level network coordination technique. In Section 3, we briefly describe the local control component of a node and how organizational structuring decisions influence this component. Section 4 presents the results of preliminary experiments with various network problem-solving strategies specified via organizational structuring. Section 5 discusses the prospects of more complex forms of meta-level control using organizational structuring. We conclude by comparing this approach to recent applications of meta-level control in knowledge-based Artificial Intelligence systems.

2 Network Coordination via Organizational Structuring

Network coordination is difficult in a cooperative, distributed problem-solving network because limited inter-node communication restricts each node's view of network problem-solving activity. In addition, network reliability issues (which require that the network's performance degrades gracefully if a portion of the network fails) preclude the use of a global "controller" node. It is important that the network coordination policies do not consume more processing and communication resources than the benefits derived from the increased problem-solving coherence. We believe that in networks composed of even a small number of nodes, a complete analysis to determine the detailed activities at each node is impractical. The computation and communication costs of optimally determining the activities far outweigh the improvement in problem-solving performance. Instead, coordination in distributed problem-solving networks must sacrifice some potential improvement for a less complex coordination problem.

What is desired is a balance between problem solving and coordination so that the combined cost of both activities is acceptable. The emphasis is shifted from optimizing the activities in the network to achieving an acceptable performance level of the network as a whole. These policies must also have enough flexibility to provide sufficient system robustness and reliability to respond to a changing task and hardware environment. In order for network control to satisfy these requirements, it must be able to tolerate the lack of up-to-date, incomplete, or incorrect control information due to delays in the receipt of information, the high cost of acquisition and processing of the information, and errors in communication and processing hardware.

We feel that the balance between local node control and network-wide control is a crucial aspect of the design of such decentralized network control policies. It is unrealistic to expect that network control policies can be developed which are sufficiently flexible, efficient, and require limited communication, while simultaneously making all the control decisions for each node in the network. We believe a node needs a sophisticated form of local control that permits it to plan sequences of activities and to adapt its plan based on its problem solving role in the network, on the status and role of other nodes in the network, and on self-awareness of its activities.

An *organizational structure* is used to provide each node with a high-level view of problem solving in the network. It specifies a general set of node responsibilities and node interaction patterns that is available to all nodes. Included in the organizational structure are control decisions that are not quickly outdated and that pertain to a large number of nodes. The sophisticated local control component of each node is responsible for elaborating these relationships into precise activities to be performed by the node. In this way we have split the network coordination problem into two concurrent activities [11]:

1. construction and maintenance of a network-wide organizational structure;
2. continuous local elaboration of this structure into precise activities using the local control capabilities of each node.

The organizational structure provides a control framework which reduces the amount of control uncertainty present in a node (due to incomplete or errorful local control information) and increases the likelihood that the nodes will be coherent in their behavior by providing a general and global strategy for network problem solving. The organizational structuring approach to limiting control uncertainty still preserves a certain level of control flexibility for a node to adapt its local control to changing task and environmental conditions.

In order for any network coordination policy to be successful, it must achieve the following conditions:

coverage — any given portion of the overall problem must be included in the activities of at least one node

connectivity — nodes must interact in a manner which permits the covering activities to be developed and integrated into an overall solution

capability — coverage and connectivity must be achievable within the communication and computation resource limitations of the network.

The organizational structure specifies a range of possible coverages and connectivity patterns that can potentially satisfy the capability condition. Using the coverage and connectivity guidelines specified in the organizational structure, the local control component of each node selects a problem-solving strategy based on the dynamics of the specific local problem solving situation.

3 An Implementation of Organizational Structuring

To provide a framework for studying the use of organizational structuring in coordinating the local activity decisions of the nodes in a cooperative, distributed problem-solving network, we have constructed the Distributed Vehicle Monitoring Testbed [7]. The testbed simulates a network of problem-solving nodes attempting to identify, locate, and track patterns of vehicles moving through a two-dimensional space using signals detected by acoustic sensors. By varying parameters in the testbed that specify the accuracy and range of the acoustic sensors, the acoustic signals that are to be grouped together to form patterns of vehicles, the power and distribution of

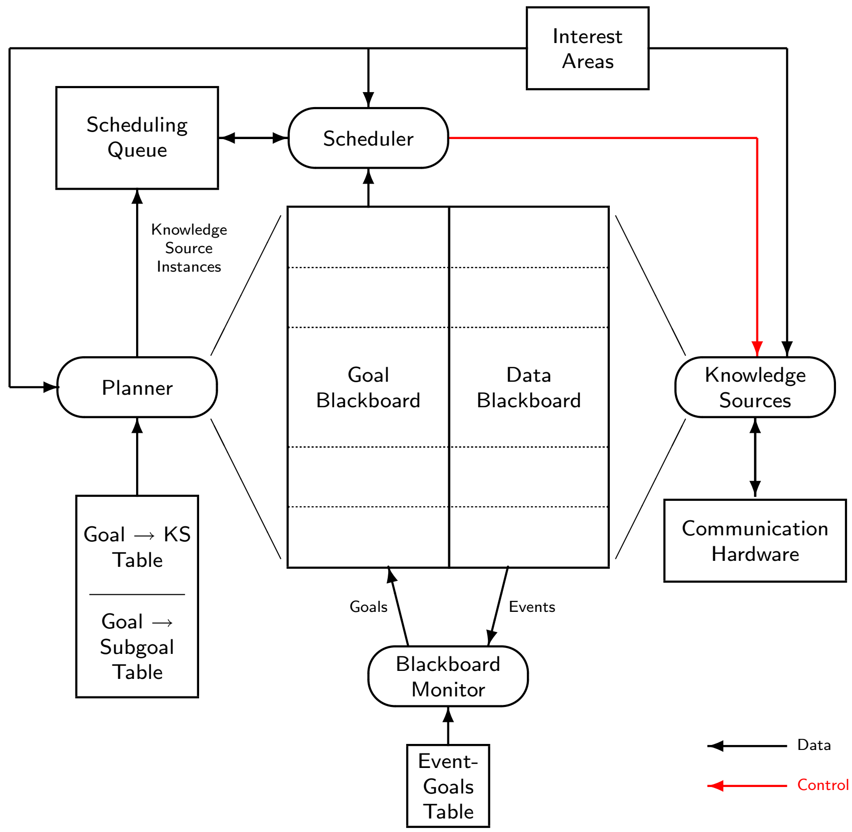

Figure 1: Testbed Node Architecture

knowledge among the nodes in the network, and the node and communication topology, a wide variety of cooperative, distributed problem-solving situations can be modeled.

Each problem-solving node is an architecturally complete Hearsay-II system [12] (with knowledge sources appropriate for the task of vehicle monitoring). The basic Hearsay-II architecture has been extended to include more sophisticated local control and the capability of communicating hypotheses and goals among nodes [13, 14]. In particular, a planning module, a goal blackboard, and communication knowledge sources have been added (Figure 1). Goals are created on the goal blackboard to indicate the node's intention to abstract and extend hypotheses on the data blackboard. The planner can adapt the node's local activities in response to the potential processing activities of the node (based on the goals created from the node's hypothesis structure) and to externally directed requests from other nodes (communicated goals).

Meta-level control via organizational structuring is introduced into this node architecture through the use of a nonprocedural and dynamically variable specification of the behaviors of each node's planner, its scheduler, and its communication knowledge sources. These data structures, called *interest areas*, are used to implement particular network configurations and coordination policies. Each interest area is a list of regions and levels of the data or goal blackboard. Associated with each interest area are one or more parameters that modify the behavior of the node. There are five sets of interest areas for each node in the testbed:

Local processing interest areas influence the local problem-solving activities in the node by modifying the priority ratings of goals and knowledge source instantiations and the behavior of the node's planner and scheduler. Each local processing interest area has a single parameter associated with it: a weight specifying the importance of performing local processing within the interest area. By changing the blackboard regions and their weights, problem solving can be restricted to particular blackboard regions and levels, and problem solving on particular regions and levels can be given priority (changing the characteristics of the search performed at a node). Knowledge sources are scheduled based on the confidence of their input data, the priority of the type of problem solving performed by the knowledge source, and the rating of processing goals. The goal rating is determined directly from the interest area weight and indirectly from the goal's relation to higher-level processing goals. Each node's local processing interest area specification also includes a sub-goaling specification. This data structure lists the blackboard levels and regions where processing goals are to be sub-goaled and the levels, sizes, and ratings of the subgoals.[1] Threshold values indicating the minimum rating needed for a goal to be sub-goaled are also specified.

Hypothesis-transmission interest areas and **goal-transmission interest areas** influence the behavior of the hypothesis and goal transmission knowledge sources at the node. Transmission interest areas are specified for one or more lists of nodes that are to receive information from the node. Each transmission interest area has a weight specifying the importance of transmitting hypotheses or goals from that area (to nodes specified in the node-list) and a threshold value specifying the minimum hypothesis belief or goal rating needed to transmit from that area.

Hypothesis-reception interest areas and **goal-reception interest areas** influence the behavior of the hypothesis and goal reception knowledge sources at the node. Reception interest areas are specified for lists of nodes that are to transmit information to the node. Each reception interest area has a weight specifying the importance of receiving a hypothesis or goal in that area (from a node specified in the node-list), a minimum hypothesis belief or goal rating needed for the hypothesis or goal to be accepted, and a credibility weight. The credibility weight parameter is used to change the belief of received hypotheses or the rating of received goals. A node can reduce the effect of accepting messages from a node by lowering the belief or rating of messages received from that node. Each hypothesis reception interest area also has a focusing weight parameter that is used to determine how heavily received hypotheses are used in making local problem-solving focusing decisions. This is accomplished by modifying the rating of local processing goals indicating potential work on these received hypotheses.

There are also additional parameters associated with the interest areas of each node that specify the relative weighting a node gives to performing activities it perceives as important versus activities proposed by other nodes. The settings of these parameters control the various authority relationships among the nodes in the network.

These interest area and authority specifications provide the interface between the activity decisions made by a node and organizational structuring decisions. They can be used to control the amount of overlap and problem solving redundancy among nodes, the problem-solving roles of nodes (such as "integrator," "specialist," and "middle manager"), the authority relations between nodes, and the potential problem-solving paths in the network. These data structures can be viewed as rudiments of a third blackboard—an *organizational blackboard* containing the organizational roles and responsibilities for the node. A node's organizational responsibilities can be established and changed by simply modifying these data structures. The specification data structures themselves do not provide an explicit, high-level representation of these organizational roles and responsibilities (this will involve future work), but instead serve as a low-level "job description" of those activities a node should be performing and those activities a node should be avoiding.

[1]The size and rating of a subgoal are specified in terms of the size and rating of the original processing goal.

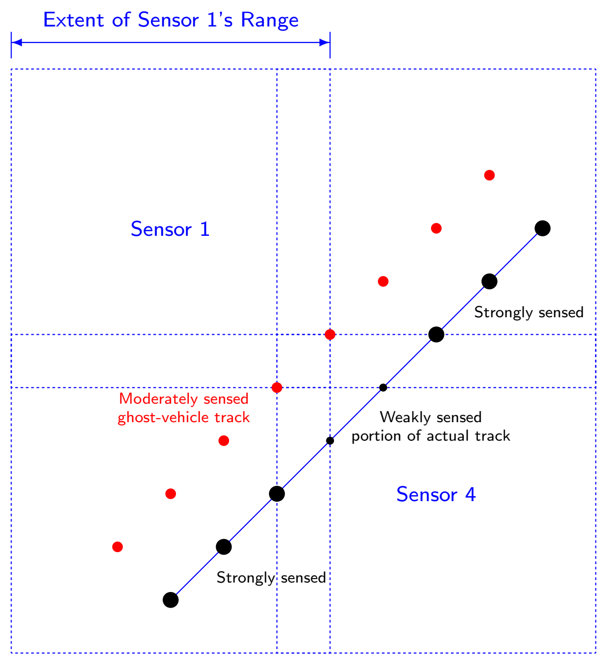

Figure 2: Sensor Configuration and Input Data

4 Testbed Experiments with Organizational Structuring

In this section we show how different organizational strategies for network problem solving can be achieved by appropriate settings of the interest area specifications at each node and how these different strategies perform in a specific distributed problem-solving situation. Characteristics that were varied included:

- whether communication is *voluntary* (a node transmits hypotheses at its pleasure), *requested* (a node transmits hypotheses only when that information is requested by another node), or a *mixed-initiative* combination of voluntary and requested hypotheses (a node volunteers only its highest rated hypotheses and awaits requests before transmitting any other hypotheses);
- whether a node is *self-directed* or *externally directed* in its activities (or a combination of both);
- whether hypotheses, goals, or both hypotheses and goals are used for inter-node coordination.

The organizational strategies were evaluated using two different network architectures: a laterally organized, four-node network with broadcast communication among nodes and a hierarchically organized, five-node network in which the fifth node acts as an integrating node. In both architectures, the network is structured so that the nodes cooperate by exchanging partial and tentative high-level hypotheses.

The sensor configuration and input sensory signal data used in these experiments is shown in Figure 2. Four sensors with identical characteristics and slightly overlapping ranges cover the monitoring area. This environmental scenario was designed to test the network's ability to use prediction to extend strongly sensed portions of an actual vehicle track through weakly sensed portions in the presence of a moderately sensed "ghost" track. Ghost tracks are a particularly problematic phenomenon in the vehicle monitoring domain, caused by multiple propagation paths of the actual signals and by geometrical ambiguity in combining signals from multiple vehicles. The ghost track in this environment mirrors the actual vehicle track for eight consecutive time frames. This is unusual. Typically ghosts behave as normal vehicles for a brief period only to abruptly disappear or to turn at sharp angles and accelerate to infinite velocity [15]. The ghost in this environment represents a "worst-case" situation, appearing as a normal vehicle with moderately strong sensory support.

Synthesis of the answer map in these experiments involves five blackboard levels: signal location, group location, vehicle location, vehicle track, and pattern track. The **Signal location** level contains hypotheses representing the output of low-level analysis of the sensory data. Each signal location hypothesis includes the frequency, approximate position, time frame,[2] and belief (based partly on signal strength and sensor quality) of the detected acoustic signal. The **group location** level contains hypotheses formed from harmonically related signal location hypotheses at the same time frame and approximately the same position. Each group location hypothesis includes the fundamental frequency of the related signals and its approximate position, time frame, and belief (a function of the beliefs and characteristics of the related signal locations). The **vehicle location** level contains hypotheses formed from group location hypotheses that can be combined to form a particular type of vehicle. Each vehicle location hypothesis includes the identity of the vehicle, approximate position, time frame, and belief. The **vehicle track** level contains hypothesized movements of vehicles over time. Each vehicle track hypothesis includes the identity of the vehicle, its approximate position at successive time frames, and belief. The **pattern track** level contains hypotheses formed from vehicle track hypotheses of specific vehicle types that maintain a particular spatial relationship among themselves. Pattern tracks were included in the testbed to investigate the effect of strong constraints between distant nodes.

In the four-node network each node is positioned near one of the sensors and receives signal location hypotheses from that sensor only. The interest areas on the organizational blackboard of each node specify that it is to synthesize its sensory data to the vehicle track level and transmit any of these vehicle track hypotheses that can be extended into the sensory area of another node to that node. Each node is also directed to attempt to generate hypotheses at both the vehicle track and pattern track levels which span the entire monitoring area. This means that each node is in a race with the other three to generate the complete answer map.

In the five-node network four of the nodes are positioned near one of the sensors and receive signal location hypotheses only from that sensor. (Their signal location input is identical to the four-node network.) The fifth node receives no sensory data. Instead, it is instructed through interest areas on the organizational blackboard to work only at the vehicle track and pattern track levels with vehicle track hypotheses received from the other four nodes. The four nodes with sensory data are assigned the role of synthesizing their signal location hypotheses to the vehicle track level and transmitting them to the fifth node. In the five-node network configuration, these four nodes do not work outside the area of their sensory data at any blackboard level and do no work at the pattern track level.

In the four-node configuration, voluntary communication is obtained by providing each node with hypothesis transmission interest areas specifying transmission of vehicle track hypothesis to nodes with sensors in the area of possible extension of these hypotheses. To keep the node entirely self-directed in its local activity decisions, each node is instructed not to generate processing goals from hypotheses received from other nodes. The beliefs of the received vehicle track hypotheses, however, are not reduced. This means that the node can use received information in extending its own hypotheses without having to find local information that can be combined with the received

[2]The environment is not sensed continuously. Instead, it is sampled at discrete time intervals called *time frames*.

hypotheses. This separation of belief in the data from focusing priority fits nicely into the integrated data- and goal-directed architecture. Externally directed control is obtained by instructing each node to create goals from hypotheses received from other nodes and to use only those goals in its local activity decisions. In this strategy, the receipt of a highly believed hypothesis from another node causes the receiving node to try its best to find something that can be combined with the received hypothesis. Combined self-directed and externally directed control is obtained by instructing each node to use goals generated from both internal and received hypotheses in its activity decisions.

The requested communication strategy is obtained by instructing each node to process its local sensory data to the vehicle track level, but rather than voluntarily transmitting vehicle track hypotheses, any vehicle track extension goals that are within the sensory area of another node are sent to that node. When a node creates a vehicle track hypothesis that satisfies one of these received goals it transmits the hypothesis to the originator of the goal. Within the requested communication strategy, self-directed, externally directed, and combined control strategies are obtained by instructing each node to use goals generated from internal hypotheses, goals received from other nodes, or both in its local activity decisions, respectively.

In the five-node configuration, mixed-initiative communication is obtained by having the worker nodes transmit only highly rated hypotheses to the integrating node. The integrating node transmits goals to the worker nodes informing them of its need for additional data. If the received goals are not used for focusing, the worker nodes remain self-directed in their local activity decisions, only responding to those goals that are achieved as a result of self-directed processing activity. If the received goals are used for focusing, the worker nodes become externally- directed and attempt to achieve the received goals. Again, a combined self-directed and externally directed approach can also be specified.

4.1 Results of the four-node network experiments

Each of the organizational problem-solving strategies were run on the environment of Figure 2. The network was stopped when the complete actual pattern track hypotheses was formed at one of the four nodes. The results are shown in Table 1. Whether the network used voluntary or requested communication of hypotheses had little effect on the number of network cycles[3] required to generate an answer. Whether the strategy was self-directed or externally directed had a much greater effect on network performance. The completely externally directed strategies performed much worse than the completely data-directed strategies, with the combined strategies in between.

Why does externally directed control perform so poorly in these experiments? A closer inspection reveals why. Node 1 (the node associated with Sensor 1) senses signal location hypotheses in only two time frames. Its signal location hypotheses are associated with the false ghost track. It does not sense the actual vehicle track at all. Having no other work to perform Node 1 quickly forms a two time-frame segment of the ghost track and transmits it to the other three nodes. This hypothesis is rated higher than the strongly sensed signal location hypotheses because it is at a higher blackboard level and appears to be a reasonable vehicle track from Node 1's perspective. Due to their bias to external direction the other three nodes suspend work on the strongly sensed lower level hypotheses of the actual track and attempt to extend the ghost track, resulting in inappropriate knowledge source activities and lost time. This is a prime example of distraction [5].

[3] A "network cycle" is the execution of one local processing knowledge source at each node in the network. If a node has no work to perform during a cycle, its potential knowledge source execution is lost.

Four-Node Experiments

Problem-Solving Strategy	Network Cycles	Sent Hyps	Sent Goals
VH/SD	33	23	0
VH/ED	86	39	0
VH/S&ED	79	45	0
RH/SD	32	32	80
RH/ED	83	35	133
RH/S&ED	75	40	78

Five-Node Experiments

Problem-Solving Strategy	Network Cycles	Sent Hyps	Sent Goals
VH/SD	27	20	0
MH/SD	25	18	14
MH/ED	40	33	30
MH/S&ED	29	22	18

Strategies:

VH	Voluntary Hypothesis Communication
RH	Requested Hypothesis Communication
MH	Mixed-Initiative Hypothesis Communication
SD	Self-Directed Control
ED	Externally Directed Control
S&ED	Combined Self-Directed and Externally Directed Control

Table 1: Summary of Network Experiments

To verify that distracting information received from Node 1 is indeed the cause of the poor performance of the externally directed strategies, the requested communication with both self-directed and externally directed control experiment was rerun with Node 1 disabled. The number of network cycles was reduced from 75 with Node 1 to 38 without Node 1. The network actually performs much better without Node 1, even though the remaining nodes still receive all signal location hypotheses associated with the ghost track.

4.2 Results of the five-node network experiments

The results of the five-node experiments are also shown in Table 1. In this case the network was stopped when the complete actual pattern track hypothesis was formed at the integrating node. Whether the network used voluntary or mixed-initiative communication of hypotheses again had little effect on the number of network cycles required to generate an answer. As with the four-node network experiments, whether the strategy was self-directed or externally directed had a much greater effect on network performance. The completely externally directed strategies performed much worse than the completely data-directed strategies, with the combined strategies in between.

In this case the information received by the integrating node (Node 5) from Node 1 causes it to make inappropriate coordination decisions for the other three worker nodes. In place of distracting hypotheses received directly from Node 1, distraction of the worker nodes takes the indirect form of distracting goals received from Node 5.

The mixed-initiative communication with externally directed control experiment was rerun with Node 1 disabled. Again the loss of Node 1 improved the performance of the network by eliminating its distracting influence. The number of network cycles was reduced from 40 with Node 1 to 29 without Node 1. The network again performed much better without the distractions

Problem-Solving Strategy	Normalized Four-Node Network Cycles	Five-Node Network Cycles
VH/SD	26.4	27
R-MH/SD	25.6	25
R-MH/ED	66.4	40
R-MH/S&ED	60.0	29

Strategies:

VH	Voluntary Hypothesis Communication
R-MH	Requested or Mixed-Initiative Hypothesis Communication
SD	Self-Directed Control
ED	Externally Directed Control
S&ED	Combined Self-Directed and Externally Directed Control

Table 2: Network Cycle Comparison of the Four- and Five-Node Experiments

from Node 1.

4.3 Comparing the four-node and five-node experiments

When the additional processing provided by the fifth node is taken into account, the performance of the lateral four-node network was basically identical with the performance of the hierarchical five-node network in comparable self-directed experiments (Table 2). The five-node network does appear to perform better than the four-node network in the externally directed strategies. When a node in the four-node network receives distracting information it generally processes it to the pattern track level before resuming work on its own lower level hypotheses (due to the generally higher belief associated with higher abstraction levels). A worker node in the five-node network only processes distracting information to the vehicle track level, and then sends the information on to the integrating node. Thus the worker node can resume its activities sooner than a node in the four-node architecture. The integrating node, while distracted, is not synthesizing low level data and is therefore less affected by the distracting information. By dividing the additional work caused by the distracting hypotheses between nodes with different problem-solving responsibilities, the overall effect of distraction is reduced.

While the experiments reported in this section indicate that different network problem-solving strategies specified via organizational structures have different problem-solving characteristics, they do not provide sufficient data for drawing any conclusions on the particular benefits of particular organizational strategies. These experiments were performed with a single environmental scenario with fairly unrestricted communication. Different problem-solving characteristics may favor different organizational strategies. Particularly important is exploration of larger networks. (These experiments are just beginning.) A four or five node network simply has too few nodes for organizational structuring decisions to have a significant impact. Experiments with tens or even hundreds of nodes are needed before the full effect of organizational structuring will be seen.

5 More Complex Meta-Level Control

While organizational structuring could be performed by directly changing the interest areas of each node (the approach used in the experiments reported here), an indirect approach allows the node to adopt or reject its organizational roles.

Instead of modifying the specifications directly, a second, separate set of node activity specification data structures is kept at each node. The original interest areas remain as the behavioral command center of the node. Their settings directly influence the node's activities. The second specifications set forms the lowest level of the full-fledged organizational blackboard. They are the result of elaborating higher-level organizational roles and responsibilities into an "organizational job description." The complete structure of this organizational blackboard, and the processing needed to perform the elaboration, remain an open research issue. What is important here is that the specifications directly controlling the behavior of a node and the behavior suggested by the organizational structure are *separated*. The node undertakes its organizational activities only by transferring organizational specifications into its interest areas.

The activities of a node should also be influenced by its potential for performing them. A node is continually receiving sensory data and hypotheses from other nodes. This information provides numerous opportunities for local node activities. However, the node's interest areas (possibly set from the organizational blackboard) may be strongly opposed to performing these activities. The node's potential for work is represented on a fourth blackboard, the *local node focusing blackboard*. This blackboard specifies where the node perceives there is substantial work it is able to perform. As with the organizational specifications, these focusing specifications can be transferred to the node's interest areas, at which point the node will actively pursue these activities.

When the roles and responsibilities represented in the organizational blackboard are in conflict with the criteria on the local node focusing blackboard, an arbiter for determining the actual interest areas is needed (Figure 3). Favoring the specifications on the organizational blackboard make the node's behavior more in line with the organizational structuring decisions (more of a "company node"), while favoring the local node focusing specifications make the node more responsive to its ability to immediately perform work on quality data. Such *node skepticism* is an important source of network robustness when organizational structuring decisions are made using incomplete and inaccurate information. A skeptical node's local activity decisions are constantly pulled in two directions: toward the responsibilities specified by the organizational structure and toward the activities suggested by its local data and interactions with other nodes. The tension between these two directions can lead to an increase in the network's ability to tolerate organizational control errors. If a node's organizational responsibilities are inappropriate to its potential activities, the node can proceed with locally generated activities. Similarly, organizational responsibilities can be ignored by nodes which possess strong information to the contrary; a node with a unique perspective is not necessarily stifled by an uninformed majority. The degree of node skepticism exhibited by a node should dynamically change according the node's perception of the appropriateness of the organizational structure. If a node has no reason to doubt the organization structure it should be receptive to organizationally specified activities. As a node becomes skeptical of the organizational structure, it should switch to its own local activities, and disregard organizational activities which are in conflict with its local activities [11].

The existence of the organizational and local node focusing blackboards also help indicate when the portion of the network organizational structure relating to the node needs changing. A strong mismatch between the two blackboards is a sign of trouble, and the information contained in the focusing blackboard can be a valuable aid in determining new roles and responsibilities.

Three additional components are relevant to the organizational structuring approach to

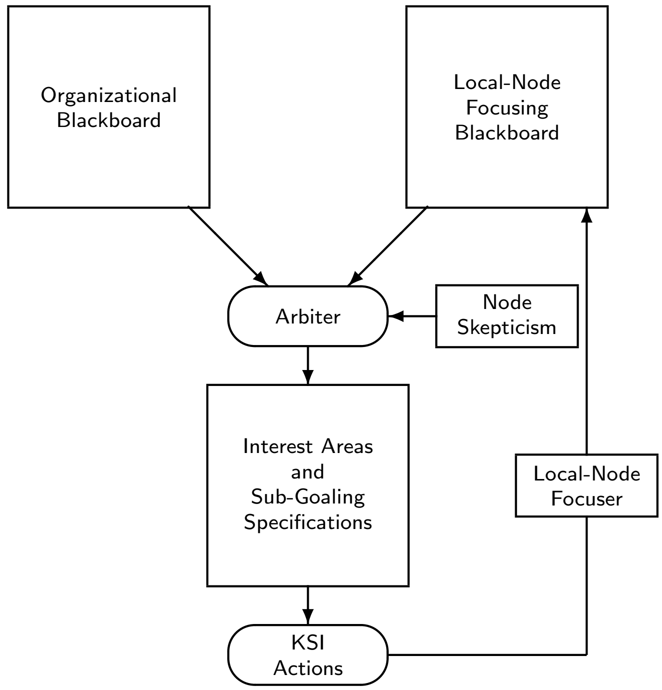

Figure 3: The Organizational and Local Node Focusing Blackboards and Node Skepticism

network coordination.

- A **distributed task allocation** component for deciding what dynamic information and processing goals should be transmitted among the nodes. Given the high-level strategic plan for the allocation of activities and control responsibilities among nodes (the organizational structure) there is still a need to make more localized, tactical decisions that balance the activities among the nodes based on the dynamics of the current problem-solving situation [16].

- A **knowledge-based fault-diagnosis** component for detecting and locating inappropriate system behavior. We are looking to not only isolate problems caused by hardware errors, but also inappropriate settings of the problem-solving parameters that specify strategic and tactical network coordination [17].

- An **organizational self-design** component for initially developing an organizational structure and for modifying that structure to reduce the effect of hardware errors or an inappropriate organizational structure (both recognized by the fault-diagnosis component). When a hardware error is detected, the the network coordination policy needs to be modified so that the offending hardware and resulting incorrect processing does not distract problem solving in other parts of the network and to establish alternative paths for generating a more accurate version of the needed information wherever possible. When the organizational structure becomes inappropriate (due to changes in the internal or external environment of the distributed problem-solving network) plausible alternative structures need to be determined and evaluated as potential candidates for network reorganization [11].

6 Conclusion

Distributed problem-solving networks provide an interesting application area for meta-level control through the use of organizational structuring. The organizational structure provides each node with a high-level view of problem solving in the network. The sophisticated local control component of each node is responsible for elaborating these relationships into precise activities to be performed by the node, based on the node's problem-solving role in the network, on the status and organizational roles of other nodes in the network, and on self-awareness of the node's activities. The balance between local node control and organizational control is a crucial aspect of this approach.

We have implemented this approach in the Distributed Vehicle Monitoring Testbed [7]. Our preliminary experiments using the testbed indicate that by adjusting the organizational structure different network problem-solving strategies can be obtained. The next step in this research is experimentation with larger distributed problem-solving networks where the effects of organizational structuring decisions will become increasingly significant.

It is interesting to note that the themes of this research, which advocate the interplay between organizational control and sophisticated local node control, are close in emphasis to recent trends emphasizing meta-level control and sophisticated planning in knowledge-based Artificial Intelligence systems [18, 19, 20, 21, 22]. The introduction of an organizational-level of control into distributed problem solving is an example of the use of meta-level control to coordinate activity in a complex system. As Nilsson has noted, the field of distributed Artificial Intelligence serves to illuminate basic Artificial Intelligence issues [23]. In this case, the need to control the uncertainty inherent with semi-autonomous problem solving agents possessing only a local and possibly errorful view of the global state of problem solving is very similar to the control problems that are being faced in the development of the new generation of knowledge-based problem solving systems which have significantly larger and more diverse knowledge bases.

Our use of meta-level control with its emphasis on providing general guidelines for acceptable problem-solving behavior differs from the work of Hayes-Roth, Davis, and Stefik which uses meta-level control to make specific strategic problem-solving decisions [18, 19, 21]. In our approach the specific strategy decisions are made by the local control component of a node using the guidelines provided by the meta-level organizational structure.

References

[1] R. Lacoss and R. Walton. Strawman design of a DSN to detect and track low flying aircraft. In *Proceedings of the Distributed Sensor Nets Workshop*, pages 41–52, 1978. Published by the Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Pennsylvania.

[2] Reid Garfield Smith. *A Framework for Problem Solving in a Distributed Processing Environment*. PhD thesis, Stanford University, December 1978. Also published as Technical Report STAN-CS-78-700. A revised version was published by UMI Research Press.

[3] Victor R. Lesser. Cooperative distributed problem solving and organizational self-design. *SIGART Newsletter*, page 46, October 1980. (Part of the "Report on the Workshop on Distributed AI").

[4] Victor R. Lesser. Models of problem-solving. *SIGART Newsletter*, (73):51, October 1980. (Part of the "Report on the Workshop on Distributed AI").

[5] Victor R. Lesser and Lee D. Erman. Distributed interpretation: A model and experiment. *IEEE Transactions on Computers*, C-29(12):1144–1163, December 1980. (Also published in *Readings in Distributed Artificial Intelligence*, Alan H. Bond and Les Gasser, editors, pages 120–139, Morgan Kaufmann, 1988.).

[6] Victor R. Lesser and Daniel D. Corkill. Functionally accurate, cooperative distributed systems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-11(1):81–96, January 1981.

[7] Victor Lesser, Daniel Corkill, Jasmina Pavlin, Larry Lefkowitz, Eva Hudlická, Richard Brooks, and Scott Reed. A high-level simulation testbed for cooperative distributed problem solving. Technical Report 81-16, Department of Computer and Information Science, University of Massachusetts, Amherst, Massachusetts 01003, June 1981. (Revised and shortened versions of this report appeared in *Proceedings of the Distributed Sensor Networks Workshop*, MIT Lincoln Laboratory, Lexington, Massachusetts, pages 247–270, January 1982, and in *Proceedings of the Third International Conference on Distributed Computer Systems*, pages 341–349, October 1982.).

[8] Richard S. Brooks. A balance principle for optimal access control. Technical Report 80-20, Department of Computer and Information Science, University of Massachusetts, Amherst, Massachusetts 01003, November 1980.

[9] Richard S. Brooks. *Experiments in Distributed Problem Solving with Iterative Refinement*. PhD thesis, University of Massachusetts, Amherst, Massachusetts 01003, February 1983. (Also published as Technical Report 82-25, Department of Computer and Information Science, University of Massachusetts, Amherst, Massachusetts 01003, October 1982.).

[10] Reid G. Smith and Randall Davis. Frameworks for cooperation in distributed problem solving. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-11(1):61–70, January 1981. (Also published in *Readings in Distributed Artificial Intelligence*, Alan H. Bond and Les Gasser, editors, pages 61–70, Morgan Kaufmann, 1988.).

[11] Daniel David Corkill. *A Framework for Organizational Self-Design in Distributed Problem-Solving Networks*. PhD thesis, University of Massachusetts, Amherst, Massachusetts 01003, February 1983. (Also published as Technical Report 82-33, Department of Computer and Information Science, University of Massachusetts, Amherst, Massachusetts 01003, December 1982.).

[12] Lee D. Erman, Frederick Hayes-Roth, Victor R. Lesser, and D. Raj Reddy. The Hearsay-II speech-understanding system: Integrating knowledge to resolve uncertainty. *Computing Surveys*, 12(2):213–253, June 1980.

[13] Daniel D. Corkill and Victor R. Lesser. A goal-directed Hearsay-II architecture: Unifying data-directed and goal-directed control. Technical Report 81-15, Department of Computer and Information Science, University of Massachusetts, Amherst, Massachusetts 01003, June 1981.

[14] Daniel D. Corkill, Victor R. Lesser, and Eva Hudlická. Unifying data-directed and goal-directed control: An example and experiments. In *Proceedings of the National Conference on Artificial Intelligence*, pages 143–147, Pittsburgh, Pennsylvania, August 1982.

[15] Peter E. Green. Distributed acoustic surveillance and tracking. In *Proceedings of the Distributed Sensor Networks Workshop*, pages 117–141, January 1982. Published by the Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Pennsylvania.

[16] Jasmina Pavlin and Victor R. Lesser. Task allocation in distributed problem-solving systems. Technical Report unpublished technical report, Department of Computer and Information Science, University of Massachusetts, Amherst, Massachusetts 01003, May 1983.

[17] Eva Hudlická and Victor R. Lesser. Diagnosing the behavior of a distributed problem-solving system. Technical Report 84-03, Department of Computer and Information Science, University of Massachusetts, Amherst, Massachusetts 01003, May 1984.

[18] Barbara Hayes-Roth and Frederick Hayes-Roth. A cognitive model of planning. *Cognitive Science*, 3(4):275–310, October–December 1979.

[19] Randall Davis. Meta-rules: Reasoning about control. *Artificial Intelligence*, 15:179–222, 1980.

[20] Michael R. Genesereth and David E. Smith. Meta-level architecture. Technical Report Stanford Heuristic Programming Project Memo HPP-81-6, Computer Science Department, Stanford University, Stanford, California 94305, December 1982.

[21] Mark Jeffrey Stefik. *Planning with Constraints*. PhD thesis, Stanford University, 1980. (Available as Technical Report STAN-CS-80-784, Computer Science Department, Stanford University, Stanford, California.).

[22] Lee D. Erman, Philip E. London, and Stephen F. Fickas. The design and an example use of Hearsay-III. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pages 409–415, Vancouver, British Columbia, August 1981.

[23] Nils J. Nilsson. Two heads are better than one. *SIGART Newsletter*, (73):43, October 1980.