A Retrospective View of FA/C Distributed Problem Solving[1]

Victor R. Lesser[2]

Computer Science Department
University of Massachusetts, Amherst

Abstract

The Functionally-Accurate, Cooperative (FA/C) paradigm provides a model for task decomposition and agent interaction in a distributed problem-solving system. In this model, agents need not have all the necessary information locally to solve their subproblems, and agents interact through the asynchronous, co-routine exchange of partial results. This model leads to the possibility that agents may behave in an uncoordinated manner. This paper traces the development of a series of increasingly sophisticated cooperative control mechanisms for coordinating agents. They include integrating data- and goal-directed control, using static meta-level information specified by an organizational structure, and using dynamic meta-level information developed in partial global planning. The framework of distributed search motivates these developments. Major themes of this work are the importance of sophisticated local control, the interplay between local control and cooperative control, and the use of satisficing cooperative control. Ongoing and new directions for research in FA/C systems are presented.

Keywords: distributed problem solving, distributed interpretation, solution and control uncertainty, satisficing, distributed search, cooperative control

1 This is an extended version of the paper that appeared *in IEEE Transactions on Systems, Man, and Cybernetics*, 21(6):1347–1362, Nov/Dec 1991. This version contains material that had to be eliminated from the original journal submission due to space constraints.

2 This paper was written while the author was on a sabbatical leave at the Knowledge Systems Laboratory of Stanford University. Long-term funding for this research has come from DARPA (NR049-041,N00014-89-J-1877), NSF (MCS78-04212, MCS-8306327, DCR-8500332) and ONR's University Research Initiative program (N00014-86-K-0764).

1. Introduction

The Functionally-Accurate/Cooperative (FA/C) model for distributed problem solving was presented in a paper that appeared in the 1981 special issue of the *IEEE Transactions on Systems, Man and Cybernetics on Distributed AI* [1]. The purpose of this retrospective is two-fold: first, to elaborate more fully this model based on insights acquired over the last ten years, and second, to provide a coherent perspective on its development and extension since its original description [2, 3, 4, 5, 6, 7]. A major focus will be on explaining why the sophistication of an agent is crucial to the effective implementation of the FA/C approach. This sophistication involves the agent's ability to understand the intermediate state of its computation (what it has done, what it has left to do, what type of information is most crucial to its further progress, etc.) and how its problem-solving activities relate to those of other agents. The framework of distributed search will be used as the basis of the explanation.

The FA/C paradigm was developed in response to what were perceived as deficiencies in the conventional model (circa 1980) of how agents (processors) in a distributed system should interact. In the conventional model, tasks are decomposed so that each agent has sufficient data to solve its assigned subproblems completely and accurately with little or no interaction with other agents. Agents do not need to communicate the intermediate stages of their computations for other agents to generate correct results. We call this model of task decomposition, *completely-accurate/nearly-autonomous*. This model was motivated by a desire to limit the communication bandwidth required to support the distributed computation.[3]

For many applications (distributed interpretation, distributed planning and resource allocation, etc.) that seem to be suited naturally for a distributed implementation, this model of task decomposition is inappropriate. It is not possible to decompose the problem into a set of subproblems such that there is a perfect fit between the location of information, expertise, processing, and communication capabilities in the agent network and the computational needs for effectively solving each subproblem (computational task, problem-

[3] Even though both local and wide area networks are increasing in speed, there is an important and often overlooked computational cost to inter-agent communication. This cost involves, on the sending side, the selection and packaging of information to be transmitted, and, on the receiving side, the assimilation of received information into local knowledge bases. If there is a large amount of information to be sent or the agents are heterogeneous, the hidden computational costs associated with transmitting and receiving information can be significant. Thus, limiting communication bandwidth is still an important consideration in structuring a distributed problem-solving system.

solving goal). Thus it requires too much communication bandwidth and synchronization time delays to guarantee that each agent has accurate, complete and up-to-date information.

Overview of FA/C Paradigm

An alternative model for task decomposition and agent interaction is one in which agents need not have all the necessary information to solve all their subproblems completely and accurately. The key question is how to structure cooperative interactions among agents to limit communication costs and still generate an acceptable answer within a reasonable amount of time. The FA/C paradigm provides an architecture for answering this question when an application can be structured in the following ways:

- agents are solving interdependent,[4] large-grained subproblems;
- agents can generate partial and tentative high-level solutions in spite of incomplete and uncertain information;
- agents can partially resolve inconsistencies and uncertainties based on constraints derived from partial solutions to interdependent subproblems received from other agents.

This resolution can take the form of producing more complete partial results, resolving solution uncertainty due to competing, alternative partial solutions, detecting inconsistencies in previously generated results (either locally generated or received from other agents), and speeding up local problem solving because the space of possible solutions that needs to be examined is constrained. In this way, *errors introduced as a result of incomplete, inconsistent and out-of-date local information are resolved as an integral part of the asynchronous, co-routine exchange of tentative, high-level partial results among agents. "Asynchronous, co-routine*" means that agents do not delay their local computations waiting to receive information from other agents, and agents exchange information based on intermediate stages of their local problem solving. This form of agent interaction is called functionally-accurate/cooperative to distinguish it from the completely-accurate/nearly-autonomous form of interaction because agents converge on an acceptable answer through the exchange of intermediate results that may be incorrect.

[4]Interdependence among subproblems arises from two basic situations. The first situation is where the subproblems are the same but different agents have either alternative methods or data that can be used to generate a solution. Another form of interdependence occurs when two subproblems are part of a larger problem in which a solution to the larger problem requires that certain constraints exist among the solutions to its subproblems.

The intent of the FA/C model of distributed problem solving is to trade off more complex and extensive local and cooperative problem solving against:

- lower message traffic (since partial results rather than raw data are communicated among agents);
- more system reliability in face of processor, communication and sensor failure (due to error resolution being an integral part of cooperative problem solving);
- less agent idle time and more parallelism (because agents are not delayed waiting for results from other agents and for validating the consistency and up-to-dateness of information through complex synchronization protocols).

More generally, the goal of FA/C distributed problem solving is to permit agents to cooperate effectively even though they have limited and inconsistent information about the problem-solving activities of other agents, different criteria for what are the most appropriate activities to perform, contradictory raw information and conflicting long-term problem-solving knowledge, and even errors in their hardware and software. The emphasis in the FA/C paradigm on handling the resolution of these inconsistencies as an integral part of cooperative problem solving among agents ties this work closely with recent work on negotiation in distributed problem solving [8, 9, 10], distributed consistency maintenance [11, 12, 13, 14] and open information systems [15, 16].

The remainder of the paper is broken into six sections. The next section (Section 2) first presents an example of FA/C problem solving. This example provides the context for discussions on the implications of this style of agent interaction on local and cooperative problem solving. Section 3 presents early experimental results to motivate the major focus of our research over the last ten years: the development of cooperative control strategies. Section 4 discusses issues in cooperative control based on an informal model of distributed search. The emphasis in this discussion is on the character of solution and control uncertainties that arise in such a distributed search and how these uncertainties can lead to uncoordinated behavior if they are not ameliorated by effective cooperative control. This discussion on solution and control uncertainty is used to explain intuitively the uncoordinated behavior observed in early experimental results discussed in Section 3. The next section (Section 5) also uses this discussion on uncertainty to examine the development of a series of increasingly sophisticated mechanisms for coordinating agent behavior. Section 6 lays out future research directions and Section 7 ends the paper with a short summary.

2. Implications for Local and Cooperative Problem Solving

To get an idea of the kinds of interactions that must occur between FA/C agents in order to converge on correct solutions, we will consider the aircraft monitoring scenario in Figure 1. Each type of aircraft produces a characteristic spectrum of acoustic frequencies. The goal of the system is to identify any aircraft that are moving through the regions of interest, determine their types, and track them through the regions. There are two agents A and B whose regions of interest overlap. Each agent receives data only about its region, from its own acoustic sensor.

Solution uncertainty arises from several sources, including improperly sensed signals, ghosting, and environmental noise. As a result of acoustic signal propagation and limitations in the acoustic sensors, not all acoustic signals emanating from an aircraft are properly sensed; some or even all of the frequencies in the spectrum may be missing and others may be shifted into the wrong frequency class. Ghost signals may appear as a result of environmental reflections of signals. Non-aircraft sources of acoustic signals may also be detected—these are referred to as noise.

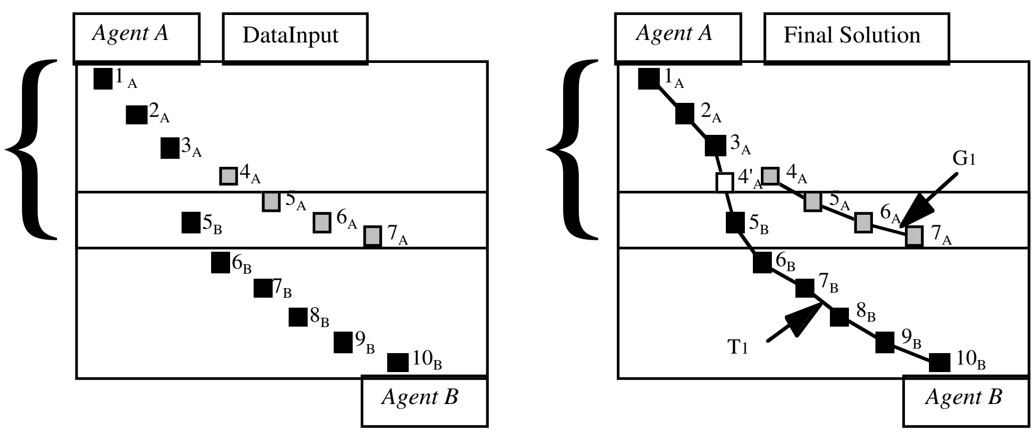

Fig. 1. This is an example of a two-agent distributed aircraft monitoring scenario. The left-hand figure is the acoustic input to the two agents and the right-hand figure is the final interpretation of the agents. The data point symbols represent the positions of groups of acoustic signals detected by the sensors. The number associated with the data points give the times that these signals were generated and the subscripts indicate the agent receiving the data. Data points include the position of the signal source and the frequency class of the signal. The shading of each box indicates the loudness of the sounds being sensed (the darker the shading the louder) and is an indication of the likelihood of the sensory data being correct. Box $4'_A$, which only appears in the final solution, is not directly supported by acoustic data (only high-level predictions) and thus is not shaded; T_1 is a vehicle track and G_1 is a ghost track caused by the environmental reflections of sounds from T_1.

Agents A and B must communicate in order to converge on the correct solution and in order to produce reasonable levels of certainty in their solutions. Without any communication, agent A would incorrectly interpret its input data (for times 1 through 7) as a ghost track. This would happen because agent A's sensor has failed to detect any signals from track T_1 at times 4 and 5 (i.e., at T_1 points $4'_A$ and 5_B in the final solution of Figure 1). Thus, the most credible interpretation of the data from agent A's local perspective involves combining the fragment of actual vehicle data at times 1 through 3 with the ghost data at times 4 through 7 into a long ghost track. Were the missing data available to agent A, it would suggest the alternative (correct) explanation of agent A's data for times 1 through 3 as being due to an actual aircraft (that produces T_1). Without any communication, agent A would also continue to be very uncertain about its ghost track explanation for the data; it would not be able to find a source for the ghost track and could not be sure that the ghost track did not continue beyond its border with agent B (since this might suggest that the data was really due to an actual aircraft). Likewise, agent B's would be uncertain over in its interpretations of its data (the time 5 through 10 portion of track T_1) because of the limited number of points over which it is able to track the vehicle.

This example also shows that a complete answer map could not be created from the agents' independent solutions. There would have to be major adjustments of some of the individual interpretations. This cooperative adjustment process requires back and forth communication between the agents rather than simply having one agent's "better" solutions override the others. This process is based on three types of interdependences among solutions to agent's subproblems: solutions involving overlapping regions of interest among agents must be consistent, "track" hypotheses that can extend into other agents' areas must be consistent, and agents must be able to find appropriate external evidence when the hypotheses require evidence which could be in other agents' areas—e.g., ghost track source (explanation) or tracking a wide-area formation of vehicles.

In this adjustment process agent A uses agent B's portion of track T_1 as predictive information, allowing agent A to make assumptions about its sensor having missed signals at times 4 and 5 that could complete track T_1. Agent A must also be able to produce an acceptable interpretation for the remainder of its original ghost track (times 4 through 7 data). Once again, communication with agent B helps to confirm most of this data (times 5 through 7 in the overlapping region) as ghost data and can provide a source (T_1) for the G_1 ghost track (see [17] for more details).

Implications for Local Problem Solving

The computation and reasoning needed in this example indicates some of the requirements that the FA/C paradigm imposes on local problem solving:

- operate in what could be called an *asynchronous mode* in which it cannot be assumed that all the information (results and raw data) necessary for fully completing a computation will be available when needed. This implies that algorithms should be structured so that they *do the best with the available information* and reevaluate decisions when more information is available or previously used information is found to be invalid.

- exploit constraints from partial solutions to interacting subproblems. This may involve a potentially complex inference process to extract the full set of constraints implied by the received partial result, to translate constraints into a form usable by the agent, and to break constraints into their constituent parts. This last capability allows important constraints to be acquired even though the partial result is not fully consistent with local information. This inference process may involve using knowledge about the sending agent and previous communications among agents. Local problem solving may also need to be reorganized so that high-level constraints can be used to speed up and to focus low-level processing of raw data.

- represent and reason about uncertainty in the current partial solutions. This may involve holding multiple, competing alternative solutions, retaining the information context used to generate partial results, and using heuristic knowledge for generating reasonable defaults in the event of missing information and for judging the plausibility of uncertain partial results.

Thus, local problem solving in an FA/C agent, in comparison to a stand-alone version of the agent, is in general more complex and computationally expensive.

Implications for Cooperative Problem Solving

How agents cooperate by sharing partial results is crucial to effective system performance because the partial solution to a subproblem may have important ramifications for problem solving throughout the system. The lack of effective cooperation (or what is commonly called *global coherence*) can lead to significant degradation in system performance due to:

- agents not generating and communicating, in a timely manner, solutions to specific subproblems that provide key constraints for further progress in overall problem solving;

- agents generating and communicating redundant results or results that are no longer appropriate given current progress in system problem solving;

- agents having no useful work to perform because of the inappropriate distribution of load among them.

Thus, the choices that an agent makes about which subproblems to work on, when to work on them, what methods to use to solve them, what type of partial results to generate, and where to send these results should not be mediated solely by the needs of local problem solving. Effective control of cooperative problem solving requires that local control decisions be influenced by the state of problem solving in other agents. This influence we will call *cooperative control*.

At one extreme, cooperative control mechanisms can be so tightly integrated into the local control that the boundary between local and cooperative control is difficult to assess. At the other extreme, cooperative control operates outside of and asynchronously with local control mechanisms. Its interface to local control is through the establishment of high level policies that the local control implements [3, 7]. In this case, the local control is clearly distinguishable from the cooperative control, has a certain amount of latitude in its decisions based on local information and may have no direct knowledge of the state of other agents' problem solving.

Cooperative control also involves the dynamic allocation of subproblems among agents based on the ongoing requirements of problem solving and the availability of hardware resources. In our discussions up to this point, it has been assumed that the partition of the subproblems (task decomposition, organizational structure, etc.) among agents is fixed depending on the *a priori* distribution of expertise, information, sensors, effectors, communication and processing resources. In fact, there are often many choices available in partitioning, especially choices involving subproblems that can be solved without access to large amounts of raw data or specialized expertise that resides in a limited number of agents. Different partitions can lead to systems that vary widely along dimensions such as reliability in face of processor or communication failure, effective processor utilization (amount of idle time), etc.

Algorithms for cooperative control are complex because the views of what other agents are doing may be incomplete, out-of-date and inaccurate due to limited communication bandwidth. This lack of an accurate view of agent activities also increases the complexity of cooperative protocols that decide when sufficient work has been done to terminate problem solving, that guarantee problem solving converges, and that recognize an overconstrained problem-solving situation.

Interplay Between Local and Cooperative Control

One lesson that we have learned is the importance of the interplay between control of cooperative and local problem solving for achieving coherent behavior among agents. The local control component needs:

- to provide to the cooperative control component a suitably abstract and relatively accurate description of its current and expected problem-solving activities, including expected results;
- to interact with the cooperative control component to modify local problem-solving activities to be more in line with the needs of other agents;
- to be sufficiently sophisticated in its decision making such that a certain level of inappropriate cooperative control decisions can be tolerated and recovered from gracefully.

The last point is crucial to the notion of satisficing control of cooperative problem solving introduced below.

Satisficing Problem Solving

An important aspect of the FA/C model that was alluded to but never developed in the original paper is the concept of *satisficing* problem solving. This concept was developed by March and Simon [18] to explain how complex organizations function when there is significant environmental uncertainty. For many applications, even though there exist strict criteria for acceptable answers, answers meeting less stringent criteria are often still useful. This approach is especially appropriate in face of unacceptable computational costs to produce answers that always meet the more exacting criteria. Similarly, the search for cooperative control algorithms optimal along all dimensions of resource utilization may be fruitless, especially in the face of the wide range of uncertainties that can arise in cooperative problem solving. The computation and communication costs to implement optimal cooperative control may lead to unacceptable overall costs. Thus, a more realistic approach is to construct cooperation strategies that produce acceptable solutions using a reasonable amount of processing resources.

3. Early Experimental Results

The major experimental validations of the FA/C model have all involved applications in the area of distributed interpretation [2, 19]. These applications have used a blackboard architecture for local problem solving because of the flexibility and modularity of its control regime and the incremental, opportunistic and asynchronous nature of its processing. This application area has provided a rich source of intuitions about the FA/C paradigm.

In the initial experiments with FA/C problem solving, the application was a distributed version of the Hearsay-II speech understanding system [2, 20]. There were three agents, each of which was a complete Hearsay-II system extended so that high-level partial results (e.g., phrase hypotheses) could be communicated. Each agent received a continuous fragment of acoustic data. These fragments overlapped to guarantee that the acoustic data associated with each word in the utterance would lay completely within at least one agent's raw data. Because all the information needed to solve the subproblem of recognizing a word was locally available in at least one agent[5], this permitted the agents to arrive at a solution without sharing low-level data. In general, interpretation systems exploit overlapping sensors to ensure more reliability in processing since sensors are not completely accurate, introducing errors such as missing data, distorted data and nonexistent data.

Experiments with this first application showed that the agents could cooperatively interact with a simple communication protocol to arrive at the same result that was produced in the centralized version of the system. The distributed system transmitted only a limited amount of information and realized a slight speedup of processing in comparison to the centralized system. Additional experiments involving an errorful communication channel indicated that problem solving was robust in spite of missing information.

Uncoordinated Agent Behavior

The experiments also highlighted important problems with the FA/C paradigm. The behavior of agents was not coherent: agents transmitted information that was often not

[5]Top-down grammatical constraints based on phrasal hypotheses were not sufficiently strong when combined with acoustic constraints based on incomplete low-level information to permit the top-down word verification knowledge source (VERIFY) to generate a *limited* number of plausible word hypotheses which included the correct choice. Thus, agents in the distributed Hearsay-II experiments were given overlapping acoustic data so as to guarantee that for every word of the utterance at least one agent had its complete acoustic constraints. Words were not hypothesized if there were only partial acoustic constraints for use in its verification. This avoided a high level of solution uncertainty in local word hypothesization. The use of incomplete low-level information would have also required significant changes to the system. The VERIFY KS would have had to be redesigned to perform only partial matching in the case of missing data. Additionally, its procedure for the estimation of word likelihood would have to have been modified so that a distinction was made between the uncertainty due to missing information and that of mismatching of data constraints.

relevant or not timely, derived information that had already been generated by another agent, and were not focused on generating results that would assist other agents. The phenomenon of *distraction* was also observed: an agent with weak constraints would quickly generate and transmit an incorrect partial result; this could cause a receiving agent to work on incorporating and extending this result instead of pursuing its own correct partial result.

These problems can be attributed directly to the highly self-directed behavior of individual agents in which there is no explicit form of cooperation. An implicit form of cooperation occurred when the agents used received information in the same way as locally generated information for making local control and communication decisions. This implicit cooperation strategy contributed to the system's robustness in spite of missing, incomplete and incorrect partial results because no expectations were made about the activity of other agents. However, this lack of expectations also led to inefficient utilization of processors and communication channels.

The experiences with this initial application of FA/C problem solving motivated much of our research agenda, based on the following question: *Can computationally tractable cooperative control strategies be developed that maintain both globally coherent activity and system robustness?* The remainder of this paper describes our experiences in answering this question, concentrating on the evolution of a series of increasingly more sophisticated mechanisms for achieving globally coherent cooperative activity. In attempting to place this evolution in perspective, we will first present an idealized model of distributed search.

4. Issues in Distributed Search

Describing the solution and control uncertainties that can arise in a simple distributed search will help to explain why maintaining globally coherent activity is difficult in an FA/C distributed problem-solving system. Consider a classical *and/or* goal structure search (depicted in Figure 2) that has been augmented to include the representation of constraints among solutions to goals and the data or resources needed to solve a primitive goal (see [21, 22, 23] for a more detailed view of the type of relationships that can occur among goals). Constraints can exist between the solutions of high-level sibling goals such as G_{k-1} and G_k, or constraints can be more distant in the goal structure such as between $G_{1,1}$ and $G_{k,2}$. In this case, G_1 and G_k become interacting goals if, for instance, $G_{1,1}$ is used to solve G_1 and $G_{k,2}$ is used to solve G_k. Further, G_1 and G_{k-1} are interdependent through a chain of interdependencies (through G_k). Constraints at one level in the structure, such as those from the solution to G_{k-1}, can have implications for solving low-level goals such as $G_{k,1,4}$. As part of this model of search, primitive goals may have multiple solutions where these solutions could be rank-ordered based on such measures as resource consumption, credibility, etc. A solution (result) for a goal is partial when its *and* subgoals have not all been solved completely.

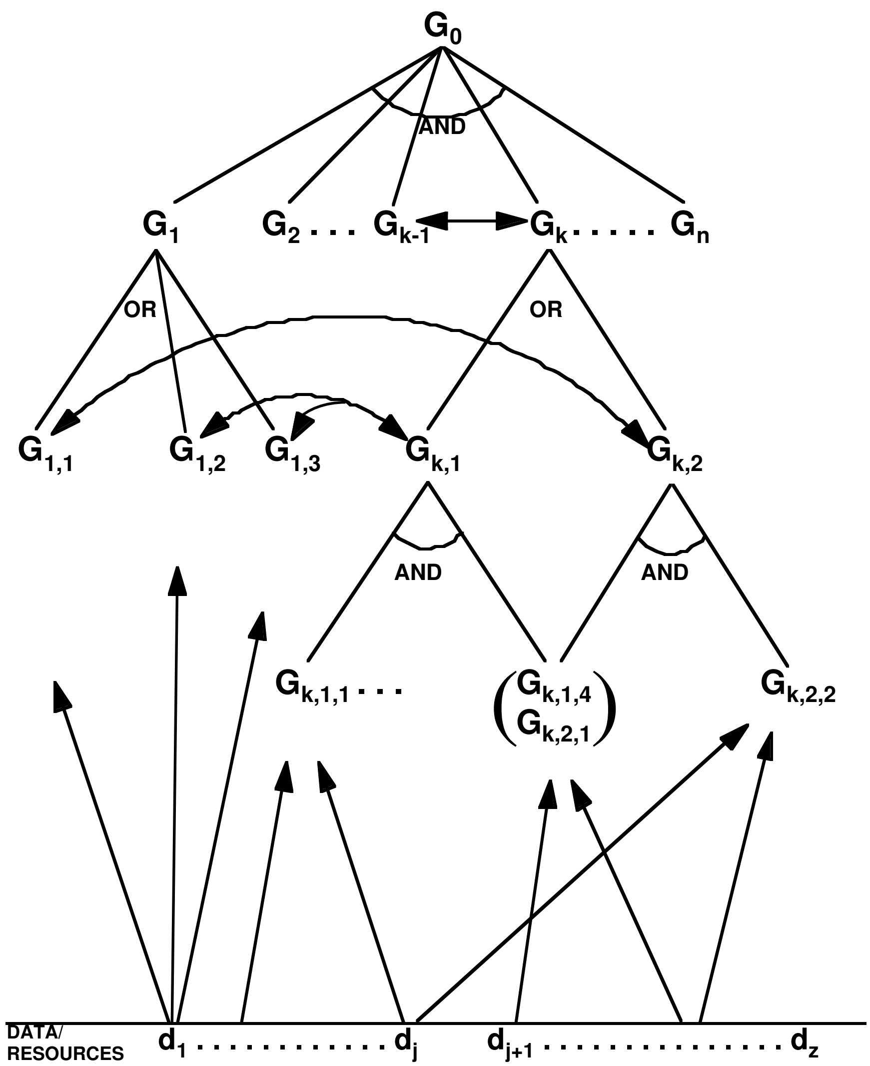

Fig. 2. A goal tree: The *G*'s represent goals and the *d*'s represent data or resources. The double-headed arrows between goals indicate that the goals are interdependent. The arrows between data and goals indicate that the data is required for that goal's solution.

In an application such as vehicle monitoring, the overall high-level goal G_0 might be to find the most consistent explanation of the sensory data, the intermediate-level goals ($G_1 ... G_n$) might be to find vehicle tracks in a narrowly defined region of time and space, and the lower-level goals ($G_{k,1,1}$ ...) might be to identify the types of vehicles present based on sensor reports (d_1....). We assume that vehicles do not instantaneously stop or reverse course. Thus, if two intermediate-level vehicle tracking goals (G_{k-1}, G_k) specify regions that are contiguous in time and space, then these goals are potentially mutually constraining. An intermediate-level vehicle tracking goal G_{k-1} can also constrain a lower-level vehicle identification goal $G_{k,1,4}$ if the sensory report being analyzed is in an area contiguous to that of the intermediate goal. Thus solutions to one of these goals implies constraints on the possible solutions to the other goal.

Multiple vehicles can share the same general area, so multiple correct solutions for a goal are possible. However, no vehicles can be in the same place at the same time so different solutions that use common data are competing alternatives. A partial solution to a vehicle identification goal can occur when the vehicle can be classified as an airplane rather than a helicopter but cannot further be distinguish as to whether it is a jet or a propeller-driven craft. More complex interactions among goals, as discussed in section 2, can occur if ghosting phenomena are modelled, movements of groups of vehicles in prespecified patterns can occur, or there can be multiple sensors, possibly of different types (e.g., radar and acoustic), that can generate reports about overlapping regions of the monitored space.

The entire goal structure need not be fully elaborated in order for problem solving to begin; rather, the structure may be constructed as problem solving progresses. This construction can involve a top-down elaboration process based on the higher level goals, a bottom-up process driven by the data, or a process combining both the top-down and bottom-up creation of goals (see [24] for a model of how goal structures are dynamically created in a distributed problem-solving system). Additionally, the criteria for problem-solving success may not be just the generation of a solution but one that is optimal along some dimension (such as cost or credibility).

In using a goal structure to represent the overall problem-solving task, we make no statement about whether this overall goal structure is statically defined or dynamically evolves from a composite view of the current, local goal structures of individual agents. Further, we make no assumption that this overall goal structure is consistent—typically it is not consistent [16]. Nor does this goal structure representation restrict the notion of a goal to a simple description of a first-order predicate. More encompassing views of goals [22, 25, 26] that define abstract and complex criteria for guiding problem solving are compatible with the discussions in this section. Goal structures provide an intuitive framework for understanding the strengths and weaknesses of different cooperative control strategies because:

> *Effective cooperative control is based on exploiting relationships among agents' goal structures.*

Solution and Control Uncertainty

Solution uncertainty in a goal-based search refers to uncertainty over whether the solution to a goal will be incorporated into the final solution of the system's top-level goal. This uncertainty is due to the possibility that:

- the final solution may not incorporate the solution to a specific goal because there are multiple derivation paths in the goal structure;
- the full set of constraints that is used to judge acceptability of a goal's solution may not be available at the time the solution is created;
- the local criteria for rating solutions may not reflect global considerations;
- the validity of the data used to construct the solution is in question.

Control uncertainty refers to the uncertainty over which goal(s) to pursue next in problem solving, what context, in terms of solutions to other goals, should be used in solving a goal, and how much effort to expend in generating a solution. This uncertainty is due not only to the fact that there are multiple, competing ways of deriving the top-level solution (and solutions to subgoals are themselves uncertain) but also that it is difficult to estimate accurately the amount of effort that will be required to produce a solution for a goal. This estimation has to take into consideration the character of the information used in problem solving (e.g., its certainty, its volume, its signal-to-noise ratio, etc.) and the impact of available constraints.

The amount of processing used can be greatly affected by the order in which goals are solved. For example, the solution to $G_{k,1,4}$ (because it is shared by the *or* goals $G_{k,1}$ and $G_{k,2}$) may provide information that indicates which of its parent *or* goals is more likely. In a similar way, the solution to G_k may provide strong constraints on the possible solutions for G_{k-1} and thus significantly speed up the search for its solution.

Solution and control uncertainty are increased when only part of the goal structure is available for decision making. This is exactly what happens in a distributed search. One of the consequences of distributing the search is that more of the search space from a composite perspective needs to be examined than in a centralized version of the search because there is more uncertainty in agents' control decisions which makes these decisions less accurate. Uncertainty is also increased when agent behavior is uncoordinated.

Consider the following example involving the control of a centralized search based on the goal structure in Figure 2. Let us assume that the search has reached a point where there are four possible goals that can be achieved; these goals and their associated ratings (indicating the perceived importance of their solution) are $G_{k-1}(90)$, $G_{1,2}(60)$, $G_{1,3}(50)$ and $G_{k,2,2}(40)$. The highest rated goal, G_{k-1}, is first achieved and, based on its solution, the ratings associated with the remaining goals are reevaluated. Additionally, new goals, that are now possible given the solution to G_{k-1}, are added to the list of goals to be considered for the next cycle; the list is now $G_{1,3}(40)$, $G_{1,2}(50)$, $G_{k,2,2}(60)$ and $G_{1,1}(60)$. If we suppose that G_{k-1},$G_{k,2,2}$ and $G_{1,1}$ were part of the final solution, notice how the solution to G_{k-1} provides information that results in a more informed control decision.

Let us now examine this same example in the context of a two-agent distributed search (see Figure 3). We start two steps earlier in the search, with $G_2(95)$ and $G_n(95)$ not yet processed. In this case, $agent_1$ has a choice among $G^1_2(95)$, $G^1_{k-1}(90)$ and $G^1_{1,2}(60)$. $Agent_2$ has a choice among $G^2_n(95)$, $G^2_{1,3}(50)$ and $G^2_{k,2,2}(40)$. The agents first choose G^1_2 and G^2_n, respectively, because they are the highest rated actions for each of them. After processing G^2_n, $agent_2$ has two choices, $G^2_{1,3}(50)$ and $G^2_{k,2,2}(40)$, neither of which is highly rated nor well differentiated. Unfortunately, the agent chose to achieve the wrong one first, $G^2_{1,3}$, because it is rated slightly higher. From a centralized perspective, the agent chose prematurely since it did not choose based on the solution to G^1_{k-1}, as in the centralized version of this search.

This example indicates the importance of coordinating the searches among agents. One strategy for achieving more coordination is for agents which have both an abundance of highly rated goals that can be achieved immediately and low control uncertainty with respect to their rating (e.g., $agent_1$) to reorder the sequence of achieving their goals (e.g., $G^1_{k-1}(90)$ before $G^1_2(95)$). This reordering should balance the efficiency of the agents' local search with the needs of other agents. This can be accomplished by first achieving those goals (e.g., $G^1_{k-1}(90)$) whose solutions can provide information to reduce the higher control and solution uncertainty present in other agents. The lack of this type of coordination in the distributed Hearsay-II experiments was partially responsible for the *early* generation of high-level, incorrect partial results that, when transmitted, distracted the problem solving of other agents. Also, the timely transmission of partial results not only can decrease control uncertainty but can also significantly speed up the solution of goals, by constraining the set of solutions that need to be examined.

Distributing the search also increases solution uncertainty because all the information necessary for validating a solution may not be available locally. For example, the solution to $G^2_{k,2,2}$ is uncertain because datum d^1_j, normally used in its construction, is not available locally (see Figure 3). Another example of increased solution uncertainty occurs with respect to $G^1_{1,2}$ which cannot completely validate its solution since only a partial solution to the constraining goal $G^1_{k,1}$ is available. Only partial solutions or solutions with low credibility for $G^1_{k,1}$ can be generated because $agent_1$ lacks information or expertise to solve all the *and* subgoals of $G^1_{k,1}$.

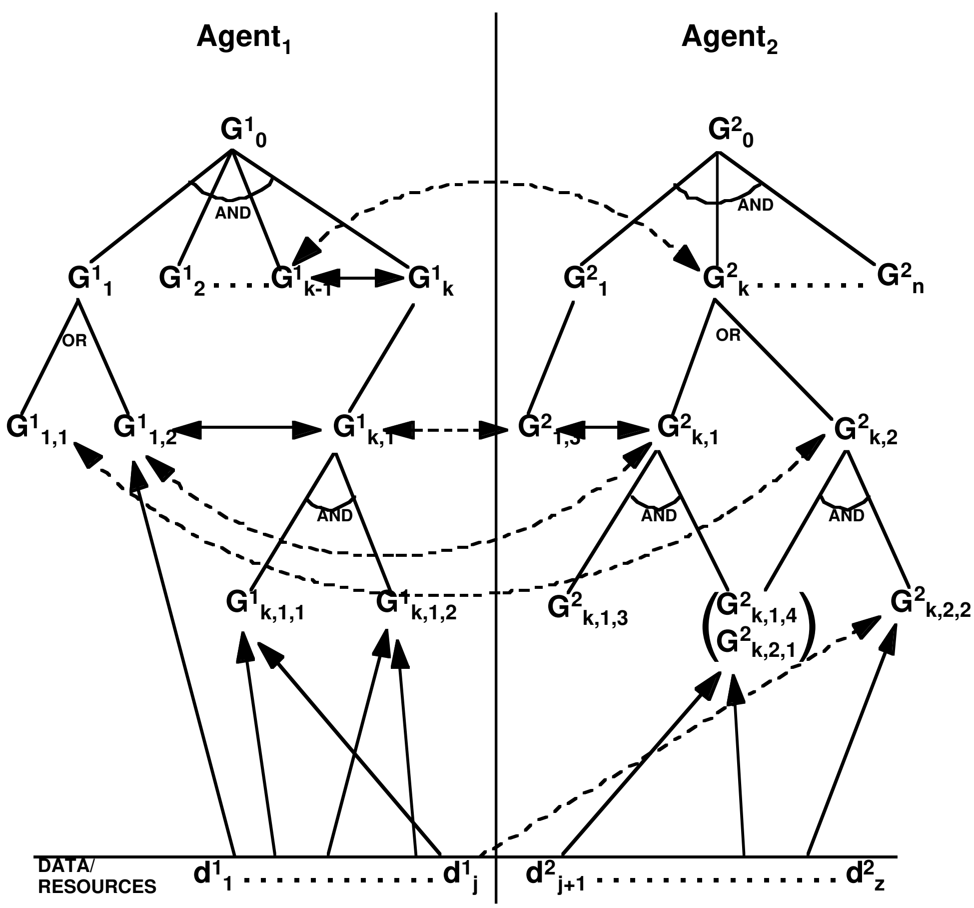

Fig. 3 A distributed goal tree: The goal tree pictured in Fig. 2 is distributed with partial replication between two agents. The dotted arrows indicate interdependencies among goals and data in different agents. The superscripts associated with goals and data indicate the agent that contains that goal or data.

Cooperative Control Uncertainty

Distributing a search among agents where their local search spaces are interdependent introduces an additional class of control uncertainties that are not present in single-agent problem solving. This new class of control uncertainties are called *cooperative control uncertainties*. These uncertainties relate to an agent's view of the state of problem solving in other agents, for example: which agents are working on interacting subproblems (goals); how difficult are these subproblems; when are they expected to be solved; what progress has been made in solving them; if they have been solved what is the character of the solutions; etc. These uncertainties about the state of other agents' problem solving leads to uncertainties in decisions concerning the communication of information among agents and the choice of which goals to pursue locally.

Uncertainties in communication occur in decisions such as to whom should an agent transmit its results and goals, what type of results or goals should it transmit, when and under what conditions should it communicate, what protocols should it use, what credence should it give to information received from specific agents, etc. Additional cooperative control uncertainties occur in decisions such as what type of solution to a goal would be most beneficial to the problem solving of other agents and when should it be generated, what is the appropriate balance between pursuing locally generated activities versus responding to externally received requests, should different priorities be attached to the type of request and the agent doing the requesting, etc. They also occur in task allocation decisions for load-balancing which are based on what type of work agents will do in the near-term and the character of the expected results.

Redundant Computation

Another important issue that arises when a search is distributed is the possibility of deriving solutions redundantly. The system often contains multiple derivation paths for solutions distributed among the agents. This redundancy adds to system robustness in the event of processor, communication, or sensor failure, permits more flexible responses to unbalanced workloads, and introduces additional constraints into problem solving to recover from errorful or incomplete information.

Redundant computation can take many forms in a distributed search. The simplest form is when agents possess exactly the same information and expertise. Another form occurs when the sibling *or* goals are placed in different agents. Thus, if an acceptable solution is generated in one of the agents, it is unnecessary to try to re-achieve the satisfied goal in the other agent until the existing solution is found to be inadequate. To avoid redundant work you must have agents that can assess the implications of received results on their local goal structure. If the entire goal structure is available, a simple process of tracing down the goal-subgoal links of the satisfied goal can be used to recognize all subgoals whose achievement can lead to redundant results. However, for many realistic situations the goal structure is not fully instantiated (especially when problem solving is data-directed) and thus new mechanisms are needed.

Avoiding redundant work also requires some anticipatory component—it is not sufficient to avoid redundant work after a result has been produced because agents may be simultaneously working on the same problem. By the time one agent has solved the goal, another agent may have already spent considerable processing resources on work that is redundant. In order to avoid redundant work, some form of meta-level information about the goal structure of other agents is needed. The more dynamic this information is about the short-term processing plans of agents, the more effective an agent can be in avoiding redundant work while at the same time still maintaining its own processing priorities. However, there is a cost to create, transmit and exploit meta-level information which may mitigate its effectiveness in improving overall problem-solving performance. These points will be more fully elaborated in the next section.

Guaranteeing that there is never redundant activity in the system may not always be the appropriate strategy. Allowing for the possibility of a certain amount of redundant work may be advisable if there are idle processing resources, if the costs of coordination control to prevent the generation of redundant results are high and redundant results are rarely generated, or if there is high solution uncertainty. For example, if there is significant solution uncertainty (or the anticipation of such) it may be appropriate to generate an alternative solution since there is a reasonable likelihood that the alternative solution will need to be eventually examined. This is especially true if it is an important goal whose solution significantly constrains future work.

5. Evolution of Control in the DVMT

The distributed Hearsay-II experiments provided many insights into FA/C problem solving, but further experimentation was difficult. Any new experiments would have required major changes to the system. Instead, we developed a new testbed based on a vehicle monitoring application similar to the application described in section 2. This interpretation application was based loosely on the distributed sensor network task that Smith used in his work on contract-nets [27]. The objective of this application was to identify, locate, and track patterns of vehicles moving through a two-dimensional space using signals detected by distributed acoustic sensors. A system, called the Distributed Vehicle Monitoring Testbed (DVMT) [19], was developed. It simulated a network of problem-solving agents performing this distributed interpretation application. Each agent was a blackboard problem-solving system with knowledge sources and levels of abstraction appropriate for this application.

The configuration of knowledge sources was designed to be modular so that it was possible to build problem-solving agents that had only a limited set of knowledge sources and levels. This modularity permitted the exploration of a wide range of different organizational structures for configuring agents in the system. Processing modularity was accomplished by structuring knowledge sources so that they would produce the best possible result based on available data and they would be reactivated in the event that new information was placed on the blackboard that could affect previously made decisions. In order to get this asynchronous form of processing to work properly, where assumptions could not be made about the order of arrival of information, a complex design for knowledge sources was required. The testbed was also highly parametrized so that accuracy, placement and range of sensors and the grammar for grouping acoustic signals together to form patterns of vehicles could be adjusted. The DVMT became the major environment for further experimentation with new approaches to control of cooperative problem solving for FA/C systems.

The three major phases in the evolution of mechanisms for the cooperative control will be discussed: integrating data- and goal-directed control, static meta-level control through organizational structuring, and dynamic meta-level control through partial global planning. Each phase builds on mechanisms developed in earlier phases. We will explain the development of this series of increasingly sophisticated cooperative control strategies in terms of how they reduce solution and control uncertainty and minimize redundant computation

Integrating Data- and Goal-Directed Control

The first step in making agents work in a more globally coherent fashion was to extend the data-directed and implicit control regime, used in the distributed Hearsay-II system, so that goals were represented explicitly. This first extension was designed to improve cooperative control by:

- more effectively exploiting constraints implicit in received information which has the effect of making agents' goal structures more consistent in terms of the importance attached to achieving interdependent subgoals; this capability also eliminates the generation of intermediate results that could lead to redundant results;

- permitting agents to request information explicitly from other agents which has the effect of increasing the relevance and timeliness [4] of transmitted information among agents;

- evaluating the importance of externally directed activities separately from that of locally directed activities which has the effect of allowing an agent to be guided by the information generated by the agent that is most likely to produce accurate information; this capability also could be used to ameliorate the effects of distracting communication.

This extension involved modifying the data-directed control cycle of the Hearsay-II system [13]. Instead of directly mapping changes to the blackboard into *knowledge source instances* (KSIs), an intermediate step was introduced. Each time the blackboard was modified, a set of data-directed goals was created representing the potential results that could be *directly* produced using this new information. For example, in Figure 2, the arrival of d_j would trigger the creation of $G_{k,1,1}$ and $G_{k,2,2}$. These goals were placed on a new structure called the *goal blackboard* that used the same levels of abstraction as the data blackboard. Goals were necessarily approximate specifications of the potential results because only a part of the information needed to actually construct the results was used in the mapping of data-to-goals. The mapping was approximate since it was not intended to duplicate the potentially time-consuming computations involved in knowledge source execution. The creation of a goal would trigger a primitive planner (goal scheduler) responsible for creating KSIs that could achieve this goal.

The goals on the goal blackboard can be thought of as a partial goal structure representing the *immediately achievable* goals (short-term problem-solving horizon) in each agent. Based on the use of this explicit representation of the partial goal structure, mechanisms were developed that achieved improved cooperative control in the ways outlined previously. One was a subgoaling mechanism that connected a higher-level goal with the lower-level goals that were necessary for its achievement. The desirability of achieving a lower-level goal was adjusted based on the importance associated with its supergoal(s). KSIs that achieved the lower-level goals were given more credence and were thus likely to be scheduled sooner. For example in Figure 3, if $agent_1$ produced a solution to G^1_{k-1} and transmitted it to *agent*$_2$, then *agent*$_2$ could generate a constrained version of G^2_k based on the fact that there are interdependences between G^1_{k-1} and G^2_k. Suppose the constraints imposed by the solution to G^1_{k-1} could be satisfied only by a solution to $G^2_{k,2}$, and $G^2_{k,1,3}$ and $G^2_{k,2,2}$ were instantiated based on data in *agent*$_2$. Then subgoaling the constrained version of G^2_k raises the rating of $G^2_{k,2,2}$ and increases the likelihood that this subgoal will be solved sooner than $G^2_{k,1,3}$. In this way, both local and cooperative control uncertainty is decreased since the importance attached to achieving interdependent goals in different agents is more consistent.

The ability to transmit goals as well as results among agents also facilitates more coherent cooperative behavior. In situations where agents have significant solution uncertainty, more informed, externally-directed communication can be implemented by agents transmitting results only in response to goals received from other agents. These received goals indicate the type of information that would be most beneficial to the requesting agent's problem solving. The subgoaling of these goals would bias the receiving agent to more quickly generate intermediate results that would lead to the satisfaction of the received goal. In this way, cooperative control uncertainty regarding communication decisions was decreased since it was more likely that the information being generated and communicated was both relevant and timely for the requesting agent.

The ratings on the goals indicate the importance to local problem solving of achieving a solution to this goal. This is in contrast to the rating associated with a partial result which indicates the likelihood of this solution being incorporated into the final answer that the system produces. The problem of distraction in the distributed Hearsay-II experiments was in part caused by there being no distinction between the belief in a received partial result and the importance of incorporating this result into a more encompassing solution. The separation of focusing information specified in the goal rating from the belief specified in the rating of a partial result allows a more explicit form of cooperative control to be implemented. The rating an agent calculates for externally received goals (or goals generated based on received information) determines the balance between responding to the needs of other agents and the perceived directions implied by its own data. The more the balance is biased towards the importance of other agents' information, the more externally-directed the agent becomes and the more likelihood there is for distraction.

On the other hand, the more importance an agent attaches to its own information, the more self-directed the agent is and the less prone to distraction. However, the potential negative consequence of the more self-directed control regime is that local problem solving may have more control and solution uncertainty because the information received from other agents will be only partially exploited. The choice of the right balance depends upon how accurate, from their own local perspective, agents are in judging the belief of their partial results and the importance of their goals from a cooperative perspective. A mechanism for implement this balancing decision will be discussed in the next section. By choosing the appropriate balance, the phenomenon of distraction can be ameliorated while still permitting local problem solving to exploit the received information.

The initial implementation of goal processing also provided a simple form of look-ahead to avoid some redundant processing. When a result is received that is highly rated, the goals on the goal blackboard that are now satisfied based on this result are marked. All KSI's that are pending on the agenda that can achieve these satisfied goals (or their subgoals) are either lowered in rating or eliminated from further consideration. This method prevents redundant work only on the solution paths directly suggested by data currently in the agent. It does not eliminate redundant activity on any of the alternative paths that might at some later time become active due to the generation, or reception of new information. This method for avoiding redundant processing fails in these situations because it is not reinvoked each time the partial representation of the local goal structure is updated to reflect new information on the possible paths available in the local search space.

In order to eliminate the derivation of any intermediate result that could eventually produce an existing high-level result, a separate mechanism, called *inhibiting goals*, was later developed [29]. This new mechanism had to be integrated with KS execution, KSs that satisfied multiple goals, including an inhibiting goal, could still be executed but were not allowed to generate results that were subsumed by the inhibiting goal specification. Since results received from other agents are only tentative, agents have the responsibility of notifying other agents when they modify their belief in a previously transmitted result due to new information. Otherwise, exploration of alternative solution paths to the goal will remain inhibited. The asynchronous mode of problem solving may require a complex decision process for deciding when a goal should no longer considered as being satisfied. It may not be directly discernible whether a partial result used to satisfy a goal was not used in further processing because:

- the appropriate information to extend the partial result has not yet been created since it is too early for the underlying information to have been received;

- the information has just not been processed;
- the desired information will never be generated.

In the last case, the belief in the correctness of the partial result rating should be lowered.[6] The ease with which these decisions are made is directly related to the explicitness of the evidential representation scheme in the agent [17]. In this way, the amount of redundant computation among agents could be decreased.

In summary, this first stage in the evolution of strategies for cooperative control showed that more coherence in agent activities could be achieved if agents dynamically constructed a partial representation of their problem-solving goal structure. This representation was based on the agent's current set of partial results. The use of this partial goal structure reduces both local control and solution uncertainty and cooperative control uncertainty by providing a framework where information (partial results and goals) from local problem solving and from other agents could be effectively exploited.

Static Meta-level Information for Control of Cooperative Problem Solving

For optimal cooperative problem solving, uncertainties in cooperative control decisions should be eliminated. This implies that these decisions should be constantly reevaluated in the light of new information generated as problem solving progresses throughout the system. Especially for larger systems, the costs to eliminate these uncertainties seem unrealistically high both from a processing and communication perspective.

An alternative approach based on a satisficing approach to cooperative control, which does not require constant reevaluation of all cooperative control decisions and is thus less computationally expensive, is to realize cooperative control decisions through a two-level decision process [3, 30]. One level, called the *organizational design level*, refers to strategic decisions that do not need to be constantly reevaluated in order to get reasonable overall performance. The other level, called the *agent level*, refers to tactical decisions that are constantly reevaluated. The organizational design level defines long-term policies that can be used at the agent level to narrow the set of choices or information that is needed to be evaluated to make decisions that implement cooperative control at each agent. These

[6] The inhibiting goal mechanism was implemented in a single-agent DVMT system but its implications for distributed problem solving were not experimentally evaluated. This is due to never implementing the code necessary to lower the belief in a hypothesis when it could never be extended.

policies define the *organizational structure* of the distributed problem-solving system. In some sense, the organizational structure represents pre-computed decisions about the likely structure of each agent's goal structure, including the type and range of solution and control uncertainties that will occur.

A policy, for example, could dictate for a specific agent the appropriate balance between pursuing local activities versus those activities dictated by external requests. The agent uses this policy together with dynamic information about the importance that the requesting agent attaches to satisfying a particular transmitted goal and the importance of achieving the locally generated goal to choose which goal to pursue. In this way, the local decision process has both a static (strategic) and dynamic (tactical) component and is thus less computationally expensive since not all information has to be computed dynamically. It is not necessary to acquire non-local information in order to estimate the importance of received and locally-generated goals. The policy provides the appropriate information for directly computing this estimate from the goal ratings.

Another example of this two-level decision process is a policy which defines a preference for an agent to work on one class of goals over another. In this case, the policy may be overridden by the local agent when the data available to solve the preferred goal is highly uncertain and incomplete while that of the less preferred goal is highly certain and complete. A third example, which is reminiscent of the focused addressing mode in the contract net protocol [27], is where the organization design level policy indicates that there are three possible agents that can satisfy a goal of a particular type. The local agent negotiates with just these three agents instead of all agents to see which one is currently most appropriate for generating the requested result. A more directed policy, which reduces both the flexibility of the organization and the cost of control, is where there is only one agent that is specified as being able to solve the particular type of goal. Like the two previous examples, the parameters of this policy can be specified so that either agents have considerable latitude in their cooperative control decisions based on local (tactical) considerations or have no real choice.

Individual policies for different aspects of cooperative control like these, taken together specify a long-term, global strategy for achieving coherent problem solving among agents. The policies guide and constrain the control decisions of agents so that their activities adhere to the global strategy. An example of an organizational structure that implements such a global strategy is the following hierarchical structure. There is a two-level hierarchy, the lower-level involves agents that only solve intermediate-level goals that are at most three levels removed from their raw data. They perform this processing opportunistically based on their belief in the raw data and the derived solutions. If a highly believed solution is generated for an intermediate-level goal, the lower-level agent sends this solution to a single higher-level agent who has the responsibility of combining the solutions from intermediate-level goals into a solution for the overall goal. As a result of this integration process, the higher-level agent recognizes the need for additional information which it requests of specific lower-level agents by sending them intermediate-level goals. The lower-level agents perceive the goals they receive from the higher-level agent as important. The lower-level agents use these goals to direct their processing.

This hierarchical organization is based on the following assumptions:

- solution and control uncertainty at the lower-levels of the goal structure is high;
- solution and control uncertainty can be significantly reduced by exploiting constraints among intermediate-level goals;
- problem solving requires the interaction among solutions to multiple, intermediate-level goals to generate strong constraints.

This last assumption is based on the following reasoning: a highly believed solution to an intermediate-level goal still has a reasonable amount of solution uncertainty; a solution for an intermediate-level goal does not significantly constrain other such goals; and sufficiently strong constraints are generated only when a number of solutions to these goals are integrated into (partial) solutions to the overall goal.

Mechanisms to implement such organizational structures were developed for the DVMT. They were built upon the integrated data- and goal-directed control discussed in the previous section. Though we have discussed the development of techniques for integrating data- and goal-directed control and the use of organizational structures as though they occurred in sequence, they were in fact both developed at the same time and are tightly intertwined. A crude form of compiled organizational knowledge was implemented through *interest area specifications* associated with each agent. These data structures specified the following:

- how to modify the belief in received information from specific agents;
- whether to generate processing goals to further extend and refine this received result;

- how to modify the importance attached to received goals from specific agents;
- whether to generate subgoals for these goals;
- how to evaluate the importance of local goals;
- what information and goals to transmit and to whom.

By appropriately setting these specifications, a wide range of static organizational structures could be configured.

We ran a number of experiments using different organizational structures to see whether these structures led to more coherent behavior. One set of experiments compared a hierarchical problem-solving organization with a lateral organization. The experimental results did not indicate clearly which organization was preferable [3, 4, 30]. Before experimentation it was thought that the hierarchical problem-solving organization would perform better. One explanation for the lack of clear-cut experimental results was that the agents' problem solving was not sufficiently sophisticated to take advantage of the more comprehensive information produced by a hierarchical organization. Implicit in the argument for a hierarchical organization is that the single, high-level agent has sufficient inferencing capabilities to exploit the available constraints. This inferencing requires that the agent generate constraints for solutions to intermediate-level goals from partial solutions to the overall goal, reason about inconsistencies in goal solutions, and direct lower-level agents to resolve these inconsistencies. The DVMT agents lacked two elements: 1) knowledge sources to generate partial results when faced with certain types of missing data, and 2) an evidential framework for reasoning about competing, alternative solutions to goals.

An example indicating the kind of knowledge sources (inferencing capabilities) that were lacking is the following (see Figure 4): suppose there are 1) G_1, G_2 and G_3 at $level_1$ that are *and* subgoals of a $level_0$ goal G_0; 2) G_2 has two *and* subgoals $G_{2,1}$ and $G_{2,2}$,[7]; and 3) solutions for G_1 constrain solutions for $G_{2,1}$; likewise, G_3 constrains $G_{2,2}$, and $G_{2,1}$ and $G_{2,2}$ are mutually constraining. Given solutions to G_1 and G_3, the system should be able to

[7] This example is abstracted from a real situation in the DVMT. The solution to $level_1$ and $level_0$ goals is a track hypothesis, and to $level_2$ goals is a vehicle location hypothesis. The DVMT did not contain knowledge sources that could combine two track hypotheses that were not overlapping in at least one time period into a larger track. For example, if one track hypothesis covered times 3 to 5 and the other track hypothesis covered times 6 to 10, these two hypotheses could not be combined into one covering the times 3 to 10, even though constraints such as velocity, acceleration, spatial location and vehicle type were compatible among the two hypotheses.

generate both a partial result at $level_0$ and a constrained version of G_2 that can focus processing at $level_2$ to generate a solution for G_2 that will be compatible with the solutions for G_1 and G_3. This type of processing is important if there is significant solution uncertainty for $level_1$ goals. Recognizing that solutions to G_1 and G_3 are compatible increases the likelihood that both solutions are correct and that subgoaling of G_2 will produce a correct solution. In addition, generating a partial solution of high confidence for the $level_0$ goal can focus higher level processing on a likely solution path. Continuing with this example, if there were two alternative solutions for G_1 and both were compatible with the solution for G_3, the system should be able to focus processing on generating a solution to G_2 that would differentiate among the two solutions to G_1.

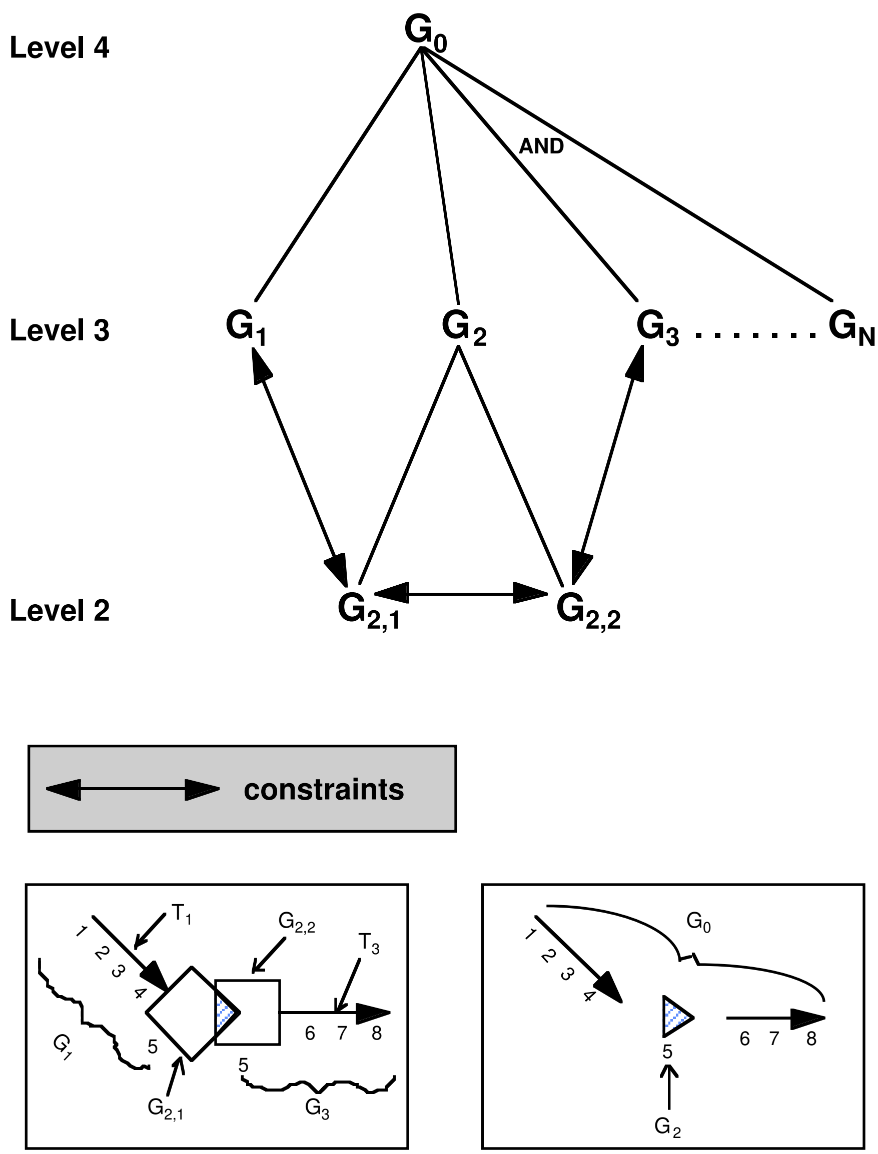

Fig. 4. Constraints in Hierarchical Problem Solving: An example of the need for more sophisticated inferencing.

It was also hoped that organizational policies would decrease the amount of redundant computation by statically prioritizing for each agent the desirability of agents solving specific types of goals. If two agents could both derive a solution to the same goal, the organizational policy could dictate a static preference for one of the agents to work on the goal. This was accomplished by lowering the priority for the other agent to try to achieve this goal. The simple priority scheme worked but it also could delay the generation of important results. The static choice process did not take into account such dynamic factors as an agent's current work load or the quality of its data. A similar phenomenon occurred in defining a static balance between an agent being locally and externally directed. When there was a long-term reason why data and goals from an agent were more or less informed than other agents, this policy avoided distraction. This can occur when one agent's sensors are prone to errors or when one agent has better data due to a more comprehensive view. However, it is common that the relative quality of an agent's information cannot be anticipated. A more dynamic mechanism must be employed to compute the appropriate balance [4].

In summary, using static meta-level information to guide and constrain agent level control decisions only partially achieved coherent behavior among agents in the DVMT. In some situations, the agents lacked the appropriate sophistication necessary to fully exploit a more informed view. In other situations, the agents' goal structures were too dynamic. There are a number of approaches to resolving these problems. One approach is organizational self-design in which the static organizational structure is dynamically adapted or even radically altered as problem solving progresses [25, 26, 31]. Another approach is to augment agent-level control decisions by incorporating dynamic information about a limited number of other agents' goal structures. A third approach is to modify local problem solving so that local problem-solving activities are less dynamic in character. This approach makes it easier to predict likely problem-solving directions. A final approach is to make agents more sophisticated so that they can better exploit available information. The next step in the evolution of cooperative control incorporates aspects of all these approaches.

Dynamic Meta-Level Information for Control of Cooperative Problem Solving

It is clear that for agents to effectively cooperate they require a reasonably accurate view of the evolving goal structure of other agents. They need to be able to recognize and anticipate where there are interacting or redundant subproblems, when agents plan to work on these problems, how difficult these problems will be to solve, the quality of the expected result, what is the current and anticipated work load in the agent, and the flexibility that an agent has in rescheduling its activities. A set of interrelated questions naturally follow, such as: what dynamic aspects of the goal structure are important, how accurate and precise does the view of the goal structure need to be, how much does the future evolution of the goal structure need to be captured.

Approaches for providing and operating on this information are guided by two principles. The first principle is based on a satisficing view of cooperative control. Agents do not have to be totally coherent in their behavior in every situation for an approach to be worthwhile. This principle is important in situations where less than accurate control permits significant reductions in the amount of communication and processing required to implement the cooperative control algorithms. The second principle is that effective cooperative control follows from sophisticated local control. That is, agents must be able to represent and reason not only about their own local problem-solving activities but also how to accommodate problem-solving requirements of other agents.

The approach developed, called *Partial Global Planning* (PGP) [5, 6, 7], is based on each agent constructing and maintaining an intermediate-level view of the likely near-term goal structure. This view also contains the expected order in which the agent pursue these goals, and estimates for each goal's importance, solution time, and solution quality. One can consider this as a high-level plan for local problem solving. It is assumed that an agent does not often need to switch opportunistically among these intermediate-level goals but, rather, can completely solve each goal before tackling the next one. There is an obvious tension between the degree of opportunism in local problem solving (which can significantly affect local performance) and the degree of predictability in the high-level plan (which can significantly affect the cost of cooperative control). The less predictability, the more communication and processing is required by cooperative control to achieve coherent behavior [32].

The agents exchange this partial and augmented view of the expected local goal structure. By combining interacting fragments of different agents' goal structures, an agent constructs *partial global goals and plans*. An agent uses this information to understand how its activities relate to those of other agents. This information guides an agent in reordering the schedule for achieving its local goals so as:

- to exploit partial results available from other agents;

- to provide partial results in a timely manner that can constrain the solution to other agents' goals;
- to avoid redundant goal solution except where desirable;
- to reassess the importance of achieving a goal.

In the last case, we assume that goals that do not interact with goals in other agents are less likely to be correct. This information is also used to implement a sophisticated form of contract-net type task allocation in which agents that are overloaded can transfer tasks to underutilized agents [33], to make communication decisions about when partial results should be transmitted, and to decide where to aggregate partial results into solutions of more encompassing goals.

We also incorporate static organizational knowledge. In part this knowledge is used to define domain-level cooperative problem strategies such as configuring agents to work in a hierarchical organization. Static organizational knowledge also has a new role of defining the organization for performing cooperative control problem solving (meta-level organization). For example, it specifies what agents will communicate meta-level information to whom, which agents will compute the partial global plans, and the criteria for updating partial global plans when problem solving does not progress as expected.

Satisficing cooperative control occurs in many different guises in the PGP algorithms. For instance, we use a hill climbing search to create the multi-agent ordering of goal activities in a partial global plan. This non-exhaustive search can lead to non-optimal cooperative plans. Additionally, computations for different partial global plans in the same agent are not coordinated. A schedule is first produced for the highest rated partial global goal, and then less highly rated goals are scheduled in the context of scheduling decisions made for more highly rated goals. Scheduling two highly rated goals as a unit is much more computationally expensive, but sometimes leads to a better schedule. The partial global plans need not be updated when the estimated time for completion of goals changes slightly. This strategy trades off non-optimal coordination caused by out-of-date information or built-in overestimations of goal completion times against reductions in both frequency of communicating goal structures and recomputing partial global plans [32]. Creating partial global plans is also not synchronized among agents. For limited times, agents can work on partial global plans that are inconsistent since some agents views of other agents is out-of-date. A satisficing approach is also implicit in the amount of effort used to construct an agent's likely intermediate-level goal structure. All these trade-offs introduce the possibility of inappropriate control decisions. However, these inaccurate decisions are generally corrected in short-order and local problem solving can be quickly refocused based on more accurate information. Thus, the use of satisficing control has an overall positive effect on system-wide performance.

Implementing partial global planning in the DVMT required extensive changes to the agent architecture [34]. At the heart of these changes is a new data structure that represents the goals the agent is likely to pursue based on the available data. The agent computes the new data structure by using a simplified form of the knowledge used in normal knowledge-source processing. The output of this processing is an abstracted version of the high-level results likely to be produced. Whereas mapping the date to goals in early versions of the system involves a single step with a single piece of data, this new mapping process involves multiple steps and multiple pieces of data. These results are not solutions because they are ambiguous and imprecise. To speed up processing, the mapping process uses simplified knowledge that does not contain all the constraints needed to completely validate a solution and aggregates solutions at lower levels that only differ slightly. These results became the intermediate-level goals of the agent.

The agent selects an order for achieving its intermediate-level goals, and estimates both how long it will take to achieve each goal and the quality of the expected results. Progress in achieving a goal is monitored so that its estimated completion time can be revised based on actual costs and its desirability can be revised based on partial results produced. This revised agent architecture makes more informed control decisions than the simpler agent architecture used in early work. This is an example in which an agent architecture leads to not only more effective local problem solving but also more effective cooperative control because the agent can reason effectively about its local activities.

This high-level view of the agent's goal structure is near-term since it is only based on the solutions that data in the agent could generate. An agent can construct a more long-term view by predicting which goals are likely to arise based on these near-term goals. Being able to extend in time the near-term view of an agent's goal structure leads to more informed load-balancing decisions. An agent's current near-term view can indicate it is underutilized and therefore a candidate for tasks to be transferred to it. However, the agent making the transfer decision may be able to predict that the apparently underutilized agent will soon have a significant processing load [33].

In summary, using partial global planning in the DVMT leads to much more coherent activity among agents. It provides an elegant and integrated solution to many of the problems that arise when a search is distributed among agents. The key assumptions about local problem solving that make this approach feasible are:

- agents can predict the intermediate-level goal structure that is the focus of their near-term work with some level of accuracy and without significant computation;
- agents can estimate how long it takes to achieve goals;
- agents generally follow the prescribed order for achieving goals;
- agents can recognize the major subproblem/goal interactions among agents using intermediate-level goals;
- agents can transmit intermediate-level goal structure without significant communication costs.

Another assumption which so far has gone unsaid is that agents can tolerate a certain level of uncoordinated behavior. In the distributed interpretation domain, the only effect of uncoordinated behavior is that it takes longer to produce a valid interpretation of the data. This, however, can be a serious problem, especially if agents are easily distracted and take too long to realize that they are pursuing an unfruitful problem-solving path. One of the advantages of agents having sophisticated local control is that they can more rapidly recover from temporarily uncoordinated behavior. This permits satisficing cooperative control as implemented in partial global planning to be cost effective.

The partial global planning framework answered affirmatively the question that was posed at the end of section 3: *A computationally tractable approach to cooperative control can be developed, for a reasonably sophisticated application of distributed problem solving as represented by the DVMT, that maintains both globally coherent behavior among agents and system robustness.* There are, however, many questions still left unanswered with respect to the general applicability of this approach to cooperative control that will be discussed later in “Beyond Partial Global Planning.”

6. New Directions in FA/C Research

Most of the research on FA/C problem solving is based on distributed interpretation applications that use a blackboard problem-solving architecture to implement local agent problem solving. A key question is: are the techniques and intuitions for structuring FA/C systems generally applicable? This question extends not only to different application areas with different underlying problem-solving architectures but also to more complex and sophisticated interpretation applications.

We will approach this question from several different perspectives. First, we will examine intuitions about the general applicability of the FA/C paradigm based on its use in other application areas. Second, we will discuss application characteristics that require significant extensions to the partial global planning framework. Third, we will describe in some detail the research in progress to extend this framework as well as initial ideas for new coordination strategies. The section ends with some comments about the applicability of this paradigm to real world problems.

Using FA/C in Other Application Domains

The FA/C paradigm has been studied in two other application areas. These studies did not emphasize cooperative control (i.e., reducing control uncertainty) but how agents can structure their local problem solving while constrained by limited communication so that non-local constraints can be acquired and exploited (i.e., reducing solution uncertainty). Both areas are concerned with distributed resource allocation.

The first, the Multi-Fireboss Phoenix application, requires cooperative planning among agents to arrive at an acceptable plan to fight the outstanding fires [35]. Agent interaction focuses on loaning bulldozers by one agent to another agent that has insufficient local resources to effectively fight the fires in its region.

The second, and more developed application, involves restoring transmission paths for dedicated circuits in a long-haul communications network [36]. The communications network is controlled by a group of agents, each of which are responsible for allocating circuits to specific regions of the network. Multiple circuit restoral goals, located in different agents in the network, may be currently active. Often there are alternative paths through the network for restoring a circuit. The distributed problem-solving task is to find a network path for each goal. This application shows the complexities of distributed search when there is an indirect chaining of constraints among partial solutions in different agents. It also shows how difficult it is to detect an overconstrained problem while at the same time trying to solve the problem.

The focus of these studies is on techniques for performing distributed search where key constraints are not locally available. For example, the choice of which local strategy to use to fight a fire may depend on how many bulldozers can be loaned by the other agent, when they are available, and for how long. In turn, these questions can be answered in different ways depending upon how much flexibility both agents have to modify their own fire-fighting plans. Likewise, one agent's solution of a subgoal to restore a path through its region may have implications for how another agent goes about solving its part of a different goal. Both applications also present situations in which the necessary resources to satisfy all the constraints specified in the goal structure are lacking. This requires the agents to both recognize an overconstrained situation and to find a satisficing solution.

This FA/C style of cooperation is effective in both cases. The basic means for cooperation among agents in both applications is an iterative exchange of tentative solutions. The exchange results in each agent acquiring a more global view of the constraints that affect the suitability of its local solutions. Both implementations indicate the importance of agents being able to understand the state of their local search and how partial solutions received from other agents alter further local search. In summary, these two new applications reinforce the observation that an agent's ability to understand both its own search process and the impact of the results produced by other agents is one of the keys to effective FA/C style problem solving.

Beyond Partial Global Planning

The partial global planning framework represents an important milestone in our understanding of how to achieve coherent cooperative behavior in an FA/C system. However, we did not address many issues in this work. Some issues were not dealt with because of limitations in the evidential reasoning capabilities in the underlying problem-solving architecture, the simplification of the application as embodied in the DVMT, and the lack of adequate processing power to simulate large and complex environments. Other issues did not arise because the nature of the application made it unnecessary to solve them in order to get effective cooperation.

Issues of Scale and Cost

One set of issues concerns large systems. Even with thirteen-agent systems [7], it is not computationally feasible to exchange partial global plans among every agent. Instead, such systems use a hierarchical meta-level organization in which top-level agents compute partial global plans for lower level agents. Can this idea be extended to networks of hundreds of agents? Alternatively, can agents dynamically form cooperative groups that coordinate closely through partial global planning and more loosely with other groups? Can the ideas of partial global planning be extended to more abstract views of search space and implicitly more abstract views and long-term views of the coordination relationships among agents? The problem of organizational self-design that permits both domain- and meta-level organizations to be defined dynamically is closely related to these questions. These questions are beginning to be addressed [25, 26]. Another related issue concerns whether the computational and communication costs of sophisticated cooperative control are always justified. Situations can arise that, based on the characteristics of the data, indicate this cost is not always warranted [4]. One such situation occurs when the goal structures of local agents are not strongly connected—there is neither a significant amount of goal redundancy nor strong constraint interactions among goals in different agents. Can these situations be recognized dynamically so that lower overhead cooperative control regimes can be used?

Complex Agent Interaction

Another set of issues relates to the type of interactions among agents that are needed for effective problem solving. In the DVMT, it was sufficient for agents to communicate only high-level partial results. In other domains, it may be necessary to selectively communicate information and goals at detailed levels. Agents may also want to engage in multi-step communication protocols and, depending upon the results, temporarily abandon the solution of certain goals. We have seen these more complex protocols arise when agents have more sophisticated evidential reasoning capabilities [17]. How does this increase in opportunism in local problem solving affect the suitability of partial global planning? Further, there may be more complex relationships among goals such as the *favor relation* [23]. This type of goal relationship and others of similar character require agents to not only reorder the sequence of goal achievement but to engage in joint planning to *reorganize* their goal structures (such as splitting and merging of goals) to take advantage of opportunities presented by these more complex goal relationships. The partial global planning task negotiation algorithm [19] uses a simplified form of this replanning activity.

Another example of more complex relationships occurs when there are real-time constraints among goals. Agents may have alternative methods for solving a goal that trade off the precision, completeness, or certainty of a solution against the time to generate a solution [37, 38]. Also, in this situation, agents may need to interact in ways not currently possible in the partial global planning framework in order to appropriately reorganize their goal structure and dynamically change the criterion for acceptable answers.

Terminating Cooperative Problem Solving

Related to the issue of more complex and dynamic relationships among agents' goal structures is the issue of terminating cooperative problem solving. How can the partial global planning framework handle complex criteria for deciding when problem solving is complete? Can such termination algorithms be built on top of the current framework or will they need to be integrated into the basic coordination strategy? Termination of system problem solving requires similar reasoning capabilities needed for effective coordination. This includes understanding what parts of the goal structure have been explored, where solution uncertainty is present in these goals and what its character is, and whether all the system-wide constraints have been developed that could significantly affect confidence in a solution [17]. The agents in the network use this information to control further cooperative problem solving in order to reach an acceptable solution. How can this information be abstracted so that it can be transferred among agents? Continuous problem-solving systems also present issues. We currently do not know what the implications of having this type of system are for coordination strategies. New issues also arise in more heterogeneous systems where agents may have differing criteria as to which goals are most important [22, 39, 40].

Ongoing and Future Research

We are approaching these issues in multiple ways. One approach is to develop a generic version of partial global planning [22]. Partial global planning, as implemented in the DVMT, used highly specific domain relationships among goals. The emphasis in this new work is to define the existing goal relationships generically and to add new relationships so that more complex interactions among the local search spaces can be captured. Four categories of goal relationships that are being used: *domain relations*, *graph relations*, *temporal relations* and *non-computational resource constraint relations*. The most interesting of these is domain relations, which include the following goal relationships: *inhibits*, *cancels*, *constrains*, *predicts*, *causes*, *enables* and *subgoal*. This approach also permits goals at different levels of abstraction to be transmitted dynamically based on the needs of cooperative control (see [25, 26] for another approach to communicating control information at multiple levels of abstraction). Other extensions include provisions for real-time cooperative problem solving and heterogeneous agents that have differing criteria for local problem solving. This work also addresses the use of negotiation to coordinate heterogeneous agents.

We are also developing a new architecture, called RESUN [42], for local problem solving in a DVMT-like task. This architecture views interpretation as a process of gathering evidence to resolve particular sources of uncertainty in the interpretation hypotheses. We developed an evidential representation that includes explicit, symbolic statements about the sources of uncertainty in the evidence for the hypotheses. A script-based, incremental control planner focuses system activity based on information in the evidential representation. The planner' goals are constructed from the (partial) solution uncertainties that need to be resolved in the current context in order to reach an acceptable overall level of uncertainty for the final solution. The RESUN architecture can support the sophisticated, evidential reasoning that is crucial to the implementation of high-level communication protocols that implement distributed differential diagnosis, multi-sensor fusion, selective communication of information among agents at different levels of detail, complex system-wide criteria for termination of problem solving, etc. This architecture also provides the necessary inferencing capabilities that were lacking in the DVMT to exploit the more comprehensive view provided in a hierarchical organization.

The DRESUN architecture [17] is an extension of RESUN for distributed problem solving. The key to achieving the necessary complex and dynamic interactions among RESUN agents is to make the solution convergence process explicit. In our approach, this has been done by giving each agent an explicit representation of the goals that must be satisfied in order to meet the criteria for termination of (global) problem solving. Termination criteria that are not satisfied or have not been verified as satisfied, are viewed as sources of uncertainty about the global correctness of local solutions. Goals representing the need to resolve these uncertainties are posted and drive the overall problem solving process. Communication between agents results from the agents taking actions to meet these goals. Because the goals are explicit and detailed, communication between agents can be very directed. That is, instead of simply exchanging information about partial solutions, agents communicate specific *evidence* that can be used to satisfy goals of resolving particular uncertainties. Another way of viewing our approach is that we have made explicit the need to enforce constraints between possibly interdependent subproblems of the agents. We recognize (possibly) interdependent subproblems and post goals to resolve uncertainty about whether the relevant partial solutions are consistent. Thus, the DRESUN architecture permits an explicit and knowledge-based approach to resolving inconsistency that occurs among solutions to interacting goals among different agents.

A next step in our research is to symbolically represent and reason about control uncertainty. This leads to a model of FA/C problem solving in which control and solution uncertainty is explicitly represented. Based on the character of these uncertainties, agents develop plans for resolving them, change the criterion for an acceptable solution, or dynamically reorganize.

For FA/C style problem solving to be applied to real world problems, it is crucial that the designer be able to provide bounds on the quality and resources required to produce solutions. This is especially difficult for non-deterministic search algorithms that are at the heart of many knowledge-based problem-solving approaches. For this reason, real-time control with its associated techniques such as approximate processing and anytime algorithms is crucial for the practical use of the FA/C paradigm.

Quantitative Models

There is also the need for a quantitative approach to the design of the FA/C system so that performance characteristics can be anticipated. For example the following are some qualitative intuitions on the application specific requirements for the use of the FA/C paradigm:

- partial solutions to subproblems constructed through local problem solving are more often valid than not or there are sufficient number of subproblems that can be solved correctly through just local problem solving, thus seeding system-wide problem solving with enough correct constraints;
- there exist sufficiently strong constraints among the solutions to subproblems such that commonly resulting errors in local problem solving due to the use of errorful or incomplete information can be detected fairly quickly before the invalid partial solution is spread widely;
- local problem-solving performance is not overly sensitive to the order of arrival of information and a high percentage of invalid local search paths can be fairly quickly pruned or recognized as having low credibility.

The last point relates to the need for a sufficient level of internal constraints associated with each subproblem such that the use of invalid information in local problem solving does not lead to a large set of partial solutions that seem plausible. Each of the above points relates to restricting the frequency and scope of dissemination of errorful partial solutions so that convergence to a valid overall solution can occur using an acceptable level of processor and communication resources. Can we develop some theory that can be used to quantify these intuitions?

We see this ability to bound and to predict the performance characteristics of FA/C systems as one of the major objectives of future research.

7. Summary

It has been ten years since the original paper on the FA/C paradigm was published. Great strides have been made in our understanding of the issues that arise when this paradigm is used. The focus of this article is on the solution and control uncertainties that occur when a search is partitioned among agents. We examined this issue from the conceptual perspective of a goal-based search and from the more realistic perspective of a distributed interpretation task. The presence of significant amounts of solution and control uncertainties in agents' local searches gives rise to uncoordinated behavior among the agents. We described a series of increasingly sophisticated mechanisms for decreasing these uncertainties with the consequent increase in the coherence of agent activities. They include integrating data- and goal-directed control, using static meta-level information specified by an organizational structure, and using dynamic meta-level information as developed in the partial global planning framework. Each of these mechanisms provides information that reduces solution and control uncertainty.

The developments of these cooperative control mechanisms is based on three guiding principles. The first is to make local control in each agent more sophisticated so that available information about the local search is better utilized. The second is for agents to exchange meta-information about their local search space so that local control decisions can be made in the context of a more comprehensive view of the composite search space. In our experience, these first two principles are complementary since the better the local control strategy is in understanding its own search space the easier it is to construct meta-information that abstractly represents the key aspects of its search space that are important to another agent. In a similar vein, local control must have a certain level of sophistication in order to exploit the meta-information about other agents' search spaces. The third principle is satisficing control in which significantly reduced computational costs to implement cooperative control is traded-off for less than optimal but still acceptable levels of coordination among agents.

There are many outstanding issues that need to be solved before this paradigm can be practically applied to a wide class of problems. In particular, the complicated and asynchronous nature of agent interactions in an FA/C distributed search makes it hard to understand and predict the system's performance characteristics. We hope that insights from this article will inspire researchers to develop a more comprehensive theoretical perspective for viewing distributed search, which in turn will aid us in more rigorously analyzing the FA/C paradigm.

8. Acknowledgements

I want to acknowledge the importance of my collaborative research with colleagues and students in shaping the views expressed in this paper. Each of them has contributed important insights, beginning with my first published article on distributed problem solving with Lee Erman, through the long-term joint research efforts with Daniel Corkill and Edmund Durfee, and continuing in my recent work with Norman Carver, Susan Conry, Zarko Cvetanovic, Keith Decker, Kazuhiro Kuwabara, Brigitte Laasri, Hassan Laasri, Robert Meyer, and Theresa Moehlman. I also want to thank Norman Carver, Keith Decker, Edmund Durfee, Lee Erman, Les Gasser and the anonymous reviewers for their thoughtful critiques of this paper.

9. References

1. Lesser, V.R. and Corkill, D.D. "Functionally-Accurate Cooperative Distributed Systems," *IEEE Transactions on Systems, Man, and Cybernetics* — Special Issue on Distributed Problem solving, Vol. SMC–11, No.1, January 1981, pp. 81–96.

2. Lesser, V.R. and Erman, L.D. "Distributed Interpretation: A Model and an Experiment," *IEEE Transactions on Computers* — Special Issue on Distributed Processing, Vol. C–29,12, December 1980, pp. 1144–1163.

3. Corkill, D.D. and Lesser, V.R. "The Use of Meta-Level Control for Coordination in a Distributed Problem Solving Network," *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, August 1983, pp. 748–756, Karlsruhe, FRG.

4. Durfee, E.H., Lesser, V.R. and Corkill, D.D. "Coherent Cooperation Among Communicating Problem Solvers," *IEEE Transactions on Computers*, Vol. 36, Issue 11, November 1987, pp. 1275–1291.

5. Durfee, E. H. and Lesser, V. R. "Using Partial Global Plans to Coordinate Distributed Problem Solvers," *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Vol. 2, August 1987, pp. 875–883.

6. Durfee, E. H. *Coordination of Distributed Problem Solvers*, Kluwer Academic Publishing, 1988.

7. Durfee, E. H. and Lesser, V. R. “Partial global planning: A coordination framework for distributed hypothesis formation,” *IEEE Transactions on Systems, Man, and Cybernetics*, 21(5):1167–1183, Sept/Oct 1991.

8. Klein, M. “Supporting Conflict Resolution in Cooperative Design Systems,” *IEEE Transactions on Systems, Man, and Cybernetics,* 21(6):1347–1362, Nov/Dec 1991.

9. Sycara, K. “Resolving Goal Conflicts via Negotiation,” *Proceedings of the National Conference on Artificial Intelligence*, August 1988, pp. 245–250.

10. Werkman, K. “Knowledge-Based Model of Negotiation Using Shareable Perspectives,” *Proceedings of the Distributed AI Workshop*, Ch.6, October 1990.

11. Bridgeland, D.; and Huhns, M. “Distributed Truth Maintenance,” *Proceedings of the National Conference on Artificial Intelligence*, August 1990, pp. 72–77.

12. Courand, G. “Cooperation via Consensus Formation,” *Proceedings of the Distributed AI Workshop*, Ch.10, October 1990.

13. Doyle, J., and Wellman, M. “Rational Distributed Reason Maintenance for Planning and Replanning of Large-Scale Activities (Preliminary Report),” *Proceedings of the Workshop on Innovative Approaches to Planning, Scheduling and Control Workshop*, November 1990, pp. 28–36. Morgan Kaufmann.

14. Mason, C.; and Johnson R. “DATMS: A Framework for Distributed Assumption Based Reasoning.” In Les Gasser and Michael Huhns, eds., *Distributed Artificial Intelligence, Vol. II*, 1989, pp. 293–317. Morgan Kaufmann.

15. Gasser, L. “Social Conceptions of Knowledge and Action,” *Artificial Intelligence* 47, 1991, pp. 107–138.

16. Hewitt, C. “Open Information Systems Semantics for Distributed Artificial Intelligence,” *Artificial Intelligence* 47, 1991, pp. 79–106.

17. Carver, N., Cvetanovic, Z., and Lesser, V. R. “Sophisticated Cooperation in FA/C Distributed Problem Solving Systems,” *Proceedings of the National Conference on Artificial Intelligence*, July 1991.

18. March, J. and Simon, H. *Organizations*, John Wiley & Sons, 1958.

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

20. Erman, L., Hayes-Roth, F., Lesser, V., and Reddy, R., “The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty,” *Computing Surveys*, 12(2):213–253, June 1980.

21. Lesser, V.R., Corkill, D.D., Hernandez, J. and Whitehair, R.C. “Focus of Control Through Goal Relationships,” *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence,* Vol. 1, Detroit, August 1989, pp. 497–503.

22. Decker, K., and Lesser, V. "Extending the partial global planning framework for cooperative distributed problem solving," *Proceedings of the Workshop on Innovative Approaches to Planning, Scheduling and Control*, 1990, pp. 396–407. Morgan Kaufmann.

23. Von Martial, F. "Coordination of Plans in Multiagent Worlds by Taking Advantage of the Favor Relation," *Proceedings of the Distributed AI Workshop*, October 1990, Ch.21.

24. Laasri, B., Laasri, H., and Lesser, V. "Negotiation and Its Role in Cooperative Distributed Problem Solving," *Proceedings of the Distributed AI Workshop*, October 1990, Ch.9.

25. Durfee, E. and Montgomery, T. "A Hierarchical Protocol for Coordinating Multiagent Behaviors," *Proceedings of the National Conference on Artificial Intelligence,* 1990.

26. Durfee, E. and Montgomery, T. "Coordination as Distributed Search in a Hierarchical Space," *IEEE Transactions on Systems, Man, and Cybernetics,* 21(6): 1363–1378, Nov/Dec 1991.

27. Smith, R., "The contract net protocol: High-level communication and control in a distributed problem-solver," *IEEE Transactions on Computers*, C-29(12),1104–1113, December 1980.

28. Corkill, D.D., Lesser, V.R., and Hudlicka, E. "Unifying Data-Directed and Goal-Directed Control: An Example and Experiments," *Proceedings of the 1982 National Conference on Artificial Intelligence*, August 1982, pp. 143–147.

29. Lesser, V.R., Corkill, D.D., Hernandez, J. and Whitehair, R.C. "Focus of Control Through Goal Relationships," *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* Vol. 1, Detroit, August 1989, pp. 497–503.

30. Corkill, D. "A Framework for Organizational Self-Design in Distributed Problem Solving Networks." PhD thesis, University of Massachusetts, February 1983.

31. Ishida, T., Yokoo, M. and Gasser, L. "An Organizational Approach to Adaptive Production Systems," *Proceedings of the National Conference on Artificial Intelligence,* 1990.

32. Durfee, E. and Lesser, V. "Predictability versus responsiveness: Coordinating problem solvers in dynamic domains," *Proceedings of the National Conference on Artificial Intelligence*, pp. 66–71, August 1988.

33. Durfee, E.H. and Lesser, V.R. "Negotiating Task Decomposition and Allocation Using Partial Global Planning," *Distributed Artificial Intelligence*, M. Huhns and L. Gasser (eds.), Vol. 2, Pitman Publishing Ltd., London, England, 1989, pp. 229–244.

34. Durfee, E.H. and Lesser, V.R. "Incremental Planning to Control a Time-Constrained, Blackboard-Based Problem Solver," *IEEE Transactions on Aerospace and Electronic Systems*, special issue on Space Telerobotics, Vol. 24, No. 5, September 1988, pp. 647–662.

35. Moehlman, T. and Lesser, V. "Cooperative Planning and Decentralized Negotiation in Multi-Fireboss Phoenix," *Proceedings of the Workshop on Innovative Approaches to*

Planning, Scheduling and Control Workshop, November 1990, pp. 144-159. Morgan Kaufmann.

36. Conry, S. E., Kuwabara, K., Lesser, V.R., and Meyer, R. A. "Multistage Negotiation in Distributed Constraint Satisfaction," *IEEE Transactions on Systems, Man, and Cybernetics,* 21(6):1462–1477, Nov/Dec 1991.

37. Lesser, V.R., Pavlin, J. and Durfee, E.H. "Approximate Processing in Real-Time Problem Solving," *AI Magazine*, Vol. 9, No. 1, Spring 1988, pp. 49–61.

38. Decker, K.S., Lesser, V.R. and Whitehair, R.C. "Extending a Blackboard Architecture for Approximate Processing," *The Journal of Real-Time Systems*–Special Issue on AI, Vol. 2, Kluwer Publishers, 1990, pp. 47–79.

39. Lander, S. and Lesser, V. R. "Negotiation among cooperating experts," *Proceedings of the 1988 Distributed AI Workshop*, May 1988.

40. Lander, S. and Lesser, V. R. "The Use of Problem Solving Context in Choosing Conflict Resolution Strategies for Cooperating Expert Agents," *Proceedings of the International Working Conference on Cooperating Knowledge-Based Systems*, Keele, United Kingdom, October 1990.

41. Carver, N. and Lesser, V. R. "A New Framework for Sensor Interpretation: Planning to Resolve Sources of Uncertainty," *Proceedings of the National Conference on Artificial Intelligence*, July 1991.