Scientific DataLink

Report 78-24
Stanford -- KSL

Applications of the Contract Net Framework: Distributed Sensing.

Reid G. Smith, Randall Davis,

Dec 1978

78-24.

APPLICATIONS OF THE CONTRACT NET FRAMEWORK: DISTRIBUTED SENSING[1]

Reid G. Smith and Randall Davis

Heuristic Programming Project
Department of Computer Science
Stanford University
Stanford, California, 94305.

Abstract[2]

We describe the initialization and operation of a distributed sensing system based on the contract net framework. In a departure from earlier systems, task distribution is viewed as a local mutual selection process, a discussion carried on between a node with a task to be executed and a group of nodes that may be able to execute the task. This leads to the use of a control formalism based on a contract metaphor, in which task distribution corresponds to contract negotiation.

---

[1] Presented at the *ARPA DSN Symposium*, Carnegie-Mellon University, Pittsburgh, PA, December 7-8, 1978.

[2] This work has been supported in part by the Advanced Research Projects Agency under contract MDA 903-77-C-0322, and the National Science Foundation under contract MCS 77-02712. It has been carried out on the SUMEX-AIM Computer Facility, supported by the National Institutes of Health under grant RR-00785. Reid Smith is now at the Defence Research Establishment Atlantic, Dartmouth, Nova Scotia, Canada. Randall Davis is now at the MIT Artificial Intelligence Laboratory, Cambridge, Mass.

1 Distributed Problem Solving: Overview

Distributed problem solving is carried out in a processor architecture in which the individual nodes include memory as well as processing capability. In such a problem solver, control is decentralized, the nodes are loosely coupled (i.e., they spend a far greater percentage of time in computation than in communication with other nodes), they communicate via messages, and they cooperate in the solution of a single overall problem.

A framework for distributed problem solving has been developed that specifies mechanisms for communications, control, and knowledge organization. The framework is based on the human model of experts that cooperate to solve problems by transfer of messages and use a contract negotiation process to distribute tasks to be executed concurrently. In the distributed problem-solving context, the human experts correspond to processor nodes. Each node contains a number of task-specific knowledge-sources (KSs). In this paper we will focus on the communications and control aspects of the framework.

A key problem that must be resolved in a distributed problem solver is how nodes with tasks to be executed find other nodes capable of executing those tasks. We will call this the *connection problem*. In centralized problem solvers it is called the *invocation problem*; that is, which KS to invoke at any given time for the execution of a task. Because *AI* applications do not generally have well-defined algorithms for their solution, *AI* problem solvers need considerable heuristic knowledge to guide them, making the connection problem crucial.

2 The Contract Net Framework: Communications And Control

The contract net framework includes a *problem-solving protocol* [Smith, 1977] [Smith, 1978b], an extrapolation to the problem-solving level of the standard network communications protocol. This protocol encodes task-independent information that specifies the possible actions and interactions of the nodes of the problem solver and also includes slots for the task-dependent information necessary for the decisions that guide the control. The task-dependent information in the framework is encoded in a *common-internode-language*, understandable to all nodes.

The problem-solving protocol uses an announcement - bid - award sequence of *contract negotiation* to solve the connection problem. It views task distribution as an interactive process, one that that entails a discussion between a node with a task to be executed and nodes that may be able to execute the task. The protocol defines a set of messages that has so far proven adequate for both control and data distribution.

A contract net is a collection of interconnected processor nodes whose interactions are governed by a problem-solving protocol based on the contract metaphor. Each node in the net operates asynchronously and with relative autonomy. The execution of an individual task is handled as a contract. A node that generates a task advertises existence of that task to the other nodes in the net with a *task announcement*, then acts as the *manager* of that task for its duration. In the absence of any information about the specific capabilities of the other nodes in the net, the manager is forced to issue a *general broadcast* to all nodes. If, however, the manager possesses some knowledge about which of the other nodes in the net are likely candidates, then it can issue a *limited broadcast* to just those candidates. Finally, if the manager knows exactly which of the other nodes is appropriate, then it can issue a *point-to-point* announcement.[3] As work on the problem progresses, many such task announcements will be made by various managers.

Nodes in the net have been listening to the task announcements, and have been evaluating their own level of interest in each task with respect to their specialized hardware and software resources. When a task is found to be of sufficient interest, a node submits a ***bid***. A bid indicates the capabilities of the bidder that are relevant to execution of the announced task. A manager may receive several such bids in response to a single task announcement; based on the information in the bids, it selects one (or several) node(s) for execution of the task. The selection is communicated to the successful bidder(s) through an ***award*** message. These selected nodes assume responsibility for execution of the task, and each is called a ***contractor*** for that task.

A contract is thus an explicit agreement between a node that generates a task (the manager) and a node that executes the task (the contractor). Note that establishing a contract is a process of mutual selection. Available contractors evaluate task announcements made by several managers until they find one of interest; the managers then evaluate the bids received from potential contractors and select one they determine to be most appropriate. Both parties to the agreement have evaluated the information supplied by the other and a mutual selection been made.

The contract negotiation process is expedited by three forms of task-dependent information contained in a task announcement. An ***eligibility specification*** lists the criteria that a node must meet to be eligible to submit a bid. This specification reduces message traffic by pruning nodes whose bids would be clearly unacceptable. A ***task abstraction*** is a brief description of the task to be executed, and allows a potential contractor to evaluate its level of interest in executing this task relative to others that are available. An abstraction is used rather than a complete description in order to reduce the length of the message (and hence message traffic).[4] Finally, a ***bid specification*** details the expected form of a bid for that task. It enables a potential contractor to transmit a bid that contains only a brief specification of its capabilities that are relevant to the task (this specification is called a ***node abstraction***), rather than a complete description. This both simplifies the task of the manager in evaluating bids, and further reduces message traffic.[5]

Contracts are queued locally by the node that generates them until they can be awarded. If no bids are received for a contract by the time an ***expiration time*** (included in

---

[3] Restricting the set of addressees of an announcement (which we call ***focused addressing***) is typically a heuristic process, since the information upon which it is based may not be exact (e.g., it may be inferred from prior responses to announcements).

[4] One component of the abstraction is the *task type*, or generic classification of the task.

[5] The information that makes up the eligibility specification, task abstraction, and bid specification for any given example must be supplied by the applications programmer. In Section 3 we will see examples of this type of information.

the task announcement) has passed, then the contract is re-announced. This process is repeated until the contract can be awarded.[6]

The award message contains a *task specification*, which includes the complete specification of the task to be executed. After the task has been completed, the contractor sends a *report* to its manager. This message includes a *result description*, which communicates the results that have been achieved during execution of the task.

The manager may terminate the execution of contracts as necessary (with a *termination* message), and further (sub)contracts may be let in turn as required by the size of a contract or by a requirement for special expertise or data that the contractor does not have.

It is important to note that individual nodes are not designated a priori as managers or contractors. These are only roles, and any node can take on either role. During the course of problem solving, a particular node normally takes on both roles (perhaps even simultaneously for different contracts). This leads to more efficient utilization of nodes, as compared, for example, to schemes that do not allow nodes that have contracted out subtasks to take on other tasks while they are waiting for results. Individual nodes, then, are not statically tied to the control hierarchy.

Note also that the idea of transfer of expertise between nodes (via transfer of procedures or data) can be readily handled by the protocol. It can be handled as a standard contract between a node that announces (in effect) *I need the code for <procedure-description>*, and a node that bids on the task by indicating that it has the required information.

To review, the normal method of negotiating a contract is for a node (called the manager for a task) to issue a task announcement. Many such announcements are made over the course of time. Other nodes are listening and submit bids on those announcements for which they are suited. The managers evaluate the bids and award contracts to the most suitable nodes (which are then called contractors for the awarded tasks).

The normal contract negotiation process can be simplified in some instances, with a resulting enhancement in the efficiency of the protocol. If a manager knows exactly which node is appropriate for execution of a task, a *directed contract* can be awarded. This differs from the *announced contract* in that no announcement is made, and no bids are submitted. Instead, an award is made directly. In such cases, nodes awarded contracts must acknowledge receipt and have the option of refusal.

The protocol has also been designed to allow a reversal of the normal negotiation process. When the processing load on the net is high, most task announcements will not be answered with bids because all nodes will be already busy. Hence the protocol includes a *node availability announcement* message. Such a message can be issued by an idle node. It is an invitation for managers to send task announcements or directed contracts to that node.

---

[6] The is a simplified version of the actual process (and does not work in the case of a task that cannot be executed due, for example, to lack of sufficient data); see [Smith, 1978c] for the complete description.

Finally, for tasks that amount to simple requests for information, a contract may not be appropriate. In such cases, a request - response sequence can be used without further embellishment. Such messages (that aid in the distribution of data as opposed to control) are implemented as ***request*** and ***information*** messages. The request message is used to encode straightforward requests for information for which contracting is unnecessary. The information message is used both as a response to a request message and as a general data transfer message.

3 Distributed Sensing

In this section, we demonstrate the use of the contract net framework in the solution of a problem in area surveillance, such as is encountered in ship or air traffic control. The example will help to demonstrate the contract net approach to communications and control.

We consider the operation of a network of nodes, each having either sensing or processing capabilities and all spread throughout a relatively large geographic area. Such a network is called a ***Distributed Sensing System*** (DSS).

The primary aim of the system is rapid, reliable, accurate, and low-cost analysis of the traffic in a designated area. This analysis involves detection, classification, and tracking of vehicles; that is, the solution to the problem is a dynamic map of traffic in the area. Construction and maintenance of such a map requires the interpretation and integration of a large quantity of sensory information received by the collection of sensor elements.

There are many trade-offs involved in the design of a DSS architecture. We present only one possible approach that considers a limited number of these trade-offs.[7] The primary intent of this example is to demonstrate the contract net approach, and we therefore focus on the initialization and communications aspects of the DSS. For a discussion of other aspects of this problem see [Nii, 1978].

3.1 Hardware

All communication in the DSS is assumed to take place over a broadcast channel (using for example, packet radio techniques [Kahn, 1975]). The nodes are assumed to be in fixed positions known to themselves but not known a priori to other nodes in the net. Each node has one of two capabilities: sensing or processing. The sensing capability includes low-level signal analysis and feature extraction.[8] We assume that a variety of sensor types exists in the DSS, that the sensors are widely spaced, and that there is some overlap in sensor area

---

[7] Further discussion of the background issues inherent in DSS design is presented in [Smith, 1978a].

[8] In a real DSS, it is likely that sensors and low-level signal analysis devices would not be considered as statically connected parts, as it is often the case that many different types of analysis are applied to the output of a single sensor, or to that of groups of sensors taken together.

coverage. Nodes with processing capability supply the computational power necessary to effect the high-level analysis and control in the net. They are not necessarily near the sensors whose data they process.

A DSS may have several functions, ranging from analysis of vehicle data in the overall area of coverage to control over the courses and speeds of those vehicles. We consider here the analysis function. In this case the overall area map must be integrated at one node in the system (it could also be integrated by an agent outside the system, like a monitoring aircraft). We therefore distinguish one processor node as the monitor node. Its function is to begin the initialization of the DSS, and integrate the overall area map for communication to an agent outside the DSS. We will see that it does not correspond to a central controller.

Figure 3.1 is a schematic representation of a DSS. Processing nodes are shown in black, and sensing nodes in white. The monitor node is shown in white with an "M" in the middle.

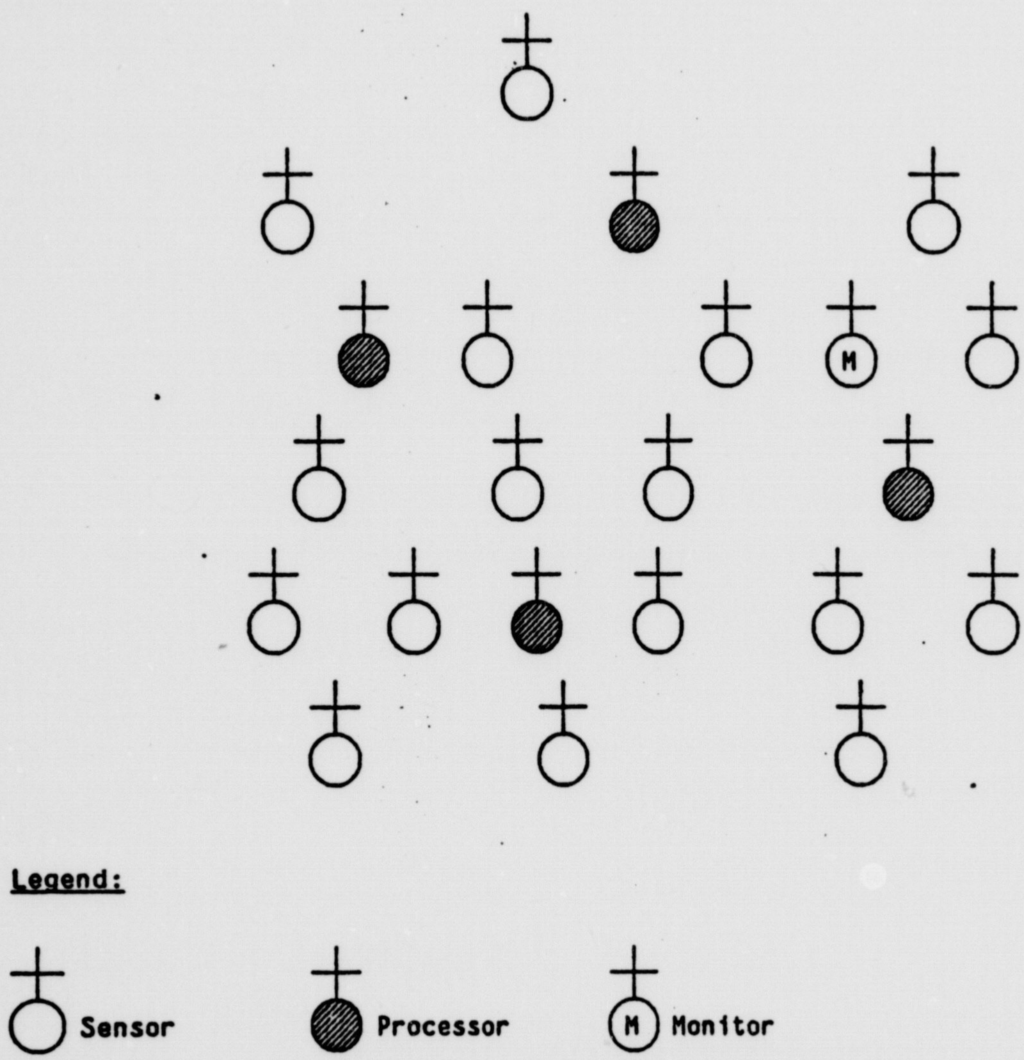

Figure 3.1. A Distributed Sensing System.

3.2 Data And Task Hierarchy

The DSS must integrate a large quantity of signal data, reducing it and transforming it into a symbolic form meaningful to and useful to a human decision maker. We view this process as occurring in several stages, which together form a data hierarchy (Figure 3.2). The hierarchy offers an overview of DSS functions and suggests a task partitioning suitable for a contract net approach. A particular node in the DSS handles data at only one level of the data hierarchy at any given moment, and communicates with nodes at other levels of the hierarchy.

overall area map

area map

vehicle

signal group

signal

Figure 3.2. Data Hierarchy.

For purposes of this example, the only form of signal processing we consider is narrowband spectral analysis. The *signal* has the following features: frequency, time of detection, strength, characteristics (e.g., increasing signal strength), name and position of the detecting node, and name, type, and orientation of the detecting sensor.[9]

Signals are formed into *signal groups* at the second level of the data hierarchy. A signal group is a collection of related signals.[10] For this example, the signal groups have the following features: the fundamental frequency of the group, the time of group formation, and the features of the signals in the group (as above).

The next level of the hierarchy is the description of the *vehicle*. It has one or more

---

[9] The frequency spectrum of noise radiated by a vehicle typically contains narrowband signal components that are caused by rotating machinery associated with the vehicle (e.g., engines or generators). The frequencies of such signals are correlated with the type of rotating machine and its speed of rotation. They are indicators of the classification of the vehicle. Narrowband signals also undergo shifts in frequency, due to doppler effect, or instability and change in the speed of rotation of the associated machine. Alterations in signal strength also occur as a result of propagation conditions and variations in the distance between the vehicle and the sensor.

[10] A signal group that is often used to integrate narrowband signal data, for example, is the harmonic set, a group of signals that are harmonically related (i.e., the frequency of each signal in the group is an integral multiple of the lowest, or fundamental frequency). A single rotating machine often gives rise to several narrowband signals that form a harmonic set.

signal groups associated with it and is further specified by position, speed, course, and type.[11] Position can be established by triangulation, using matching groups detected by several sensors with different positions and orientations. Speed and course must be established over time by tracking.

The *area map* forms the next level of the data hierarchy. This map incorporates information about the vehicle traffic in an area. It is an integration of the vehicle level data. There will be several such maps for the DSS, corresponding to areas in the span of coverage of the net.

The final level is the complete or *overall area map*. In this example, the map is integrated from the individual area maps by the monitor node.

As indicated above, the hierarchy of tasks follows directly from the data hierarchy. The monitor node manages several *area* contractors. These contractors are responsible for the formation of traffic maps in their immediate areas. Each area contractor, in turn, manages several *group* contractors that provide it with signal groups for its area (Figure 3.3). Each group contractor integrates raw signal data from *signal* contractors that have sensing capabilities.

The area contractors also manage several *vehicle* contractors that are responsible for integration of information associated with individual vehicles. Each of these contractors manages: a *classification* contractor that determines vehicle type; a *localization* contractor, that determines vehicle position; and a *tracking* contractor, that tracks the vehicle as it passes through the area.[12]

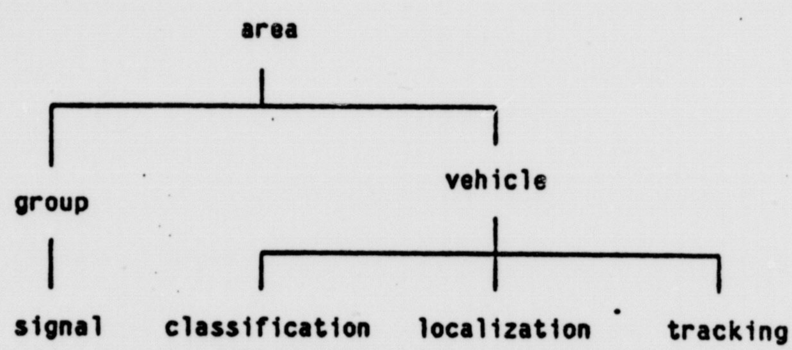

Figure 3.3. Area Task Partitioning.

---

[11] For simplicity, we have ignored a level in the hierarchy that can be called *component sources of signal*, as in [Nii, 1978]. At this level, a DSS would normally try to attribute signals and signal groups to particular pieces of machinery associated with a vehicle.

[12] In a real solution to the DSS problem, it is possible that not all of these tasks would be large enough to justify the overhead of contracting; that is, some of them might be done in a single node. Note also that some of the tasks in the hierarchy are *continuing* tasks (e.g., the area task), while others are *one-time* tasks (e.g., the localization task).

Note that this particular partitioning of tasks is only one of many possibilities that might be specified by the system designer.

3.3 Contract Net Implementation

This section reviews in qualitative terms how the DSS problem can be attacked using the contract net approach and illustrates several of the ideas central to its operation. Appendix A gives specific examples of the message traffic that is described here.

3.3.1 Initialization

The monitor node is responsible for initialization of the DSS and for formation of the overall map. It must first select nodes to be area contractors and partition the system's span of coverage into areas, based on the positions of those selected nodes. For purposes of illustration we assume that the monitor node knows the names of nodes that are potential area contractors, but it must establish their positions in order to partition the overall span of coverage. Hence, it begins by announcing contracts for formation of area maps of the traffic. Because the monitor node knows the names of potential area contractors, it can avoid a general broadcast and can instead use a focused addressing scheme. The announcement contains the three components described in Section 2: a task abstraction, an eligibility specification, and a bid specification. The task abstraction is simply the task type, and the eligibility specification is blank (because the monitor node knows which other nodes are potential contractors and addresses them directly). The bid specification is of primary interest for this task. It informs a prospective area contractor to respond with its position. Remember that the purpose of a bid specification is to enable a manager to select, from all of the bidders, the most appropriate nodes to execute a contract. Node position is the information required by the monitor node to make that selection. Given that information, the monitor node can partition the overall span of coverage into approximately equal-sized areas, and select a subset of the bidders to be area contractors. Having decided upon a partitioning, the monitor node broadcasts an information message to the other nodes in the system. This message defines the names and specifications (in terms of latitude and longitude ranges) of the individual areas. Each selected area contractor is then informed of its area of responsibility in an award message.[13]

The area contractors' purpose is to integrate vehicle data into area maps. They must first establish the existence of vehicles on the basis of signal group data. Therefore, each area contractor solicits other nodes to provide signal group data. In the absence of any information about which nodes are suitable, each area contractor announces the task using a general broadcast. The task abstraction in these announcements is the type of task. The eligibility specification is the area for which the individual area contractor is responsible;

---

[13] The full announcement-bid-award sequence is necessary (rather than a directed contract) because the monitor node needs to know the positions of all of the potential area contractors in order to partition the overall span of coverage of the DSS into manageable areas. Note that this means that the DSS can adjust to a change in the number or position of potential area contractors.

that is, a node is only eligible to bid on this task if it is in the same area as the announcing area contractor. This restriction helps to prevent a case in which a signal group contractor is so far away from its manager that reliable communication is difficult to achieve. The bid specification is again node position. Potential group contractors respond with their respective positions, and, based on this information, the area contractors award signal group contracts to nodes in their areas of responsibility.

The signal group contractors' task is to integrate signal data from sensor nodes into signal groups. Therefore, they must first find nodes that will provide raw signal data. This is done with signal task announcements. The eligibility specification in these announcements indicates that only those nodes located in the same area as the announcer and having sensing capabilities should bid on this task. The task abstraction indicates the task type and position of an individual signal group contractor. This information assists potential signal contractors in determining the group contractors to which they should respond.[14]

The potential signal contractors listen to the task announcements from the various group contractors. They respond to the nearest group contractor with a bid that supplies their position and a description of their sensors. The group contractors use this information to select a set of bidders that covers their immediate vicinity with a suitable variety of sensors, and then award signal contracts on this basis. The awards specify the sensors that each signal contractor is to use to provide raw data to its managing group contractor. Figure 3.4 depicts the exchange between one group contractor (the black node), and several potential signal contractors (the white nodes). Successful bidders are connected by solid lines to the group contractor and unsuccessful bidders are connected by dashed lines.

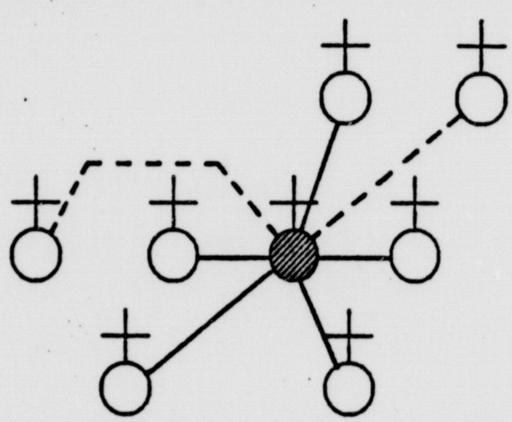

Figure 3.4. Signal Contract Negotiation.

[14] The signal task is to detect signals and report them to a signal group contractor in a particular position. Hence the position of the group contractor is reasonable as part of the abstraction for the task.

There are some potential problems of asynchrony in the receipt of announcements from group contractors. A potential contractor for signal tasks must determine the group contractor that is closest to it by listening to several signal task announcements. The potential signal contractors use the expiration times of the announcements (i.e., the times after which no further bids will be accepted) as a guide to the length of time during which they can listen to announcements before submitting a bid. In the best case, the expiration times are long enough to allow announcements from group contractors to reach all local potential signal contractors so that optimum partitioning can be achieved. In the worst case, however, a potential signal contractor may submit a bid to a group contractor that is not the closest one to it (because the task announcement of the closest group contractor is not received until after a bid has already been submitted to another group contractor). The result is a sub-optimal partitioning.

The signal contract is a good example of the contract negotiation process, illustrating how the matching of contractors to managers is an interactive process. It involves a mutual decision based on local processing by both the group contractors and the potential signal contractors. The potential signal contractors base their decision on a distance metric and respond to the closest manager. The group contractors use the number of sensors and distribution of sensor types observed in the bids to select a set of signal contractors that ensures that every area is covered by every kind of sensor. Thus each party to the contract evaluates the proposals made by the other, using a different evaluation function, and a task distribution agreement is completed via mutual selection.

Reviewing the status of the DSS, we have a single monitor node which manages several area contractors. Each area contractor manages several group contractors, and each group contractor manages several signal contractors. The data initially flows from the bottom to the top of this hierarchy. The signal contractors supply raw signal data; each group contractor integrates the raw data from several signal contractors to form a signal group, and these groups are passed along to the area contractors, which eventually form area maps by integrating information based on the data from several group contractors. All the area maps are then passed to the monitor which forms the final traffic map.

As we have noted, in this example one area contractor manages several group contractors and each group contractor in turn manages several signal contractors. It is possible, however, that a single group contractor should supply information to several area contractors, and a single signal contractor should supply information to several group contractors. It may be useful, for instance, to have a particular group contractor near an area boundary report to the area contractors on both sides of the boundary. This is easily accommodated within our framework.

3.3.2 Comments On The DSS Organization

We have taken a top-down, distributed approach to initializing the DSS. An alternative approach might involve acquisition of the positions of all nodes at a very early stage, followed by area definition, award of area contracts, and so on. This would involve a more global approach to the problem of initialization, using a single node that initialized the net by gathering together all the necessary data. We have not pursued this approach for several reasons, primarily because it would tell us little about solving problems in a distributed manner. An underlying theme of this research is a search for ways in which to effect distributed problem solving--rather than ways to do traditional problem solving in a distributed architecture.

Moreover, there are two practical difficulties with the global approach. First, it concentrates a large amount of message traffic and processing at a single node (say the monitor node) because such a node would be responsible for accepting position messages from every other node in the net. Second, it may not be possible for any single node to communicate directly with all other nodes in a widely separated collection. This would mean that either indirect routing of messages would be required for communication, or that each of the nodes would require powerful transmitters. This is one of the advantages of the distributed and dynamically-defined organization we have adopted: only pairs of nodes that are close together enough to communicate directly are linked together with contracts.

3.3.3 Operation

We now consider the activities of the system as it commences operation.

When a signal is detected or when a change occurs in the features of a known signal, the detecting signal contractor reports this fact to its manager (a group contractor) (Figure 3.5). This node, in turn, attempts to integrate the information into an existing signal group or to form a new signal group.

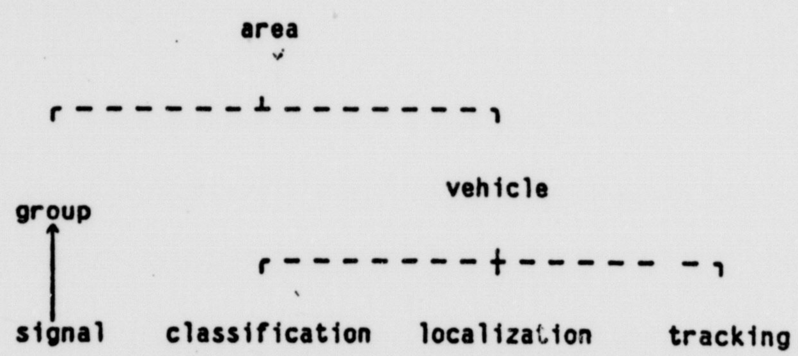

Figure 3.5. Signal Contract Reporting.

A group contractor reports the existence of a new signal group to its manager (an area contractor) (Figure 3.6).

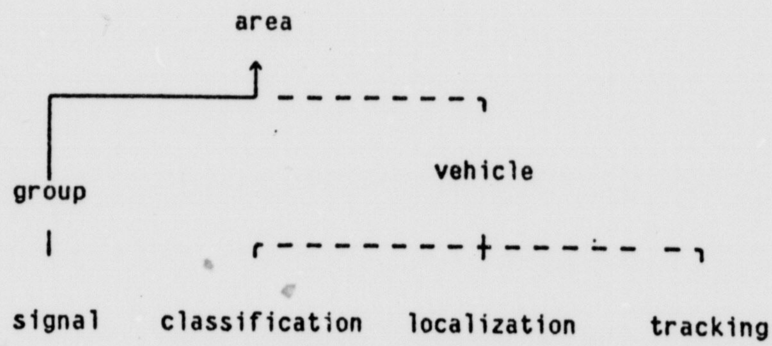

Figure 3.6. Group Contract Reporting.

Whenever a new group is detected, the managing area contractor attempts to find a node to execute a vehicle contract (Figure 3.7). The task of a vehicle contractor is to classify, localize, and track the vehicle associated with the signal group. Since a newly detected signal group may be attributable to a known vehicle, the area contractor first requests from the existing collection of vehicle contractors a measure of confidence in the fact that the new group is attributable to one of the known vehicles. Based on these responses, the area contractor either starts up a new vehicle contractor or augments the existing contract of the appropriate existing vehicle contractor, with the task of making certain that the new group corresponds to a known vehicle. This may entail, for example, the gathering of new data via the adjustment of sensors or contracts to new sensor nodes.

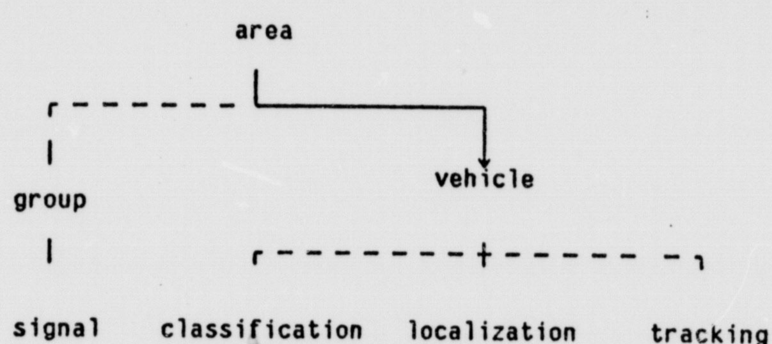

Figure 3.7. Vehicle Contract Initiation.

The form of the signal group confirmation request demonstrates a tradeoff that arises in many distributed problem solving applications--a tradeoff between communication and local processing. The area contractor has the option of transmitting to the existing vehicle contractors either the complete signal group or an abstraction of it (e.g., its fundamental frequency). In either case, the response to the request is a measure of the individual vehicle contractor's confidence that the group corresponds to a vehicle it knows about. If a vehicle contractor returns a high confidence measure in its bid, then in the first case (complete signal group announced) the area contractor has the information it wants. In the second case, however, the area co tractor must still transmit the complete signal group description and await a further report.

The first approach has the disadvantage of using up more local processing time in each of the vehicle contractors than does the second, in that the original request message is more complex. The question of interest is, *Under what conditions should a complete signal group description be announced instead of an abstraction?* The answer appears to depend on the quality of the abstraction. If the abstraction is good enough that the vehicle contractors are able to make definitive statements on the basis of its use, then the second approach is likely better than the first, in that it minimizes local processing time. On the other hand, if the abstraction does not allow definitive statements to be made, then its use will result in increased message traffic and local processing time, since the area contractor will not be certain as to the best course of action as a result of uncertain responses from the vehicle contractors.

The vehicle contractor then makes two task announcements: vehicle classification and vehicle localization. The task abstraction of the classification task announcement is a list of the fundamental frequencies of the signal groups currently associated with the vehicle. This information may help a potential classification contractor select an appropriate task (a contractor may, for example, already be familiar with vehicles that have signal groups with the announced fundamental frequencies). The eligibility specification is blank. The bid specification indicates that a bidder should respond with a tentative classification and an associated confidence measure. This measure is used to select a classification contractor--the bidder with the highest confidence is chosen. The award is the complete current description of the vehicle.

A classification contractor may be able to classify directly, given the signal group information; or, on the other hand, it may require more data, in which case it can communicate directly with the appropriate sensor nodes (Figure 3.8).

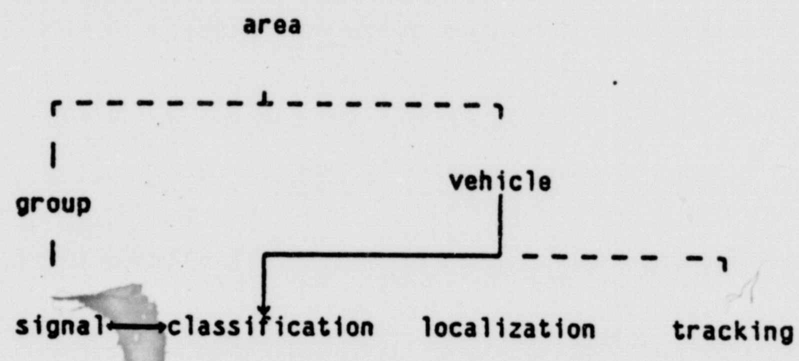

Figure 3.8. Classification Contract Communication.

The task abstraction for a localization task announcement (Figure 3.9) is a list of positions of the nodes that have detected a vehicle. The eligibility specification and bid specification are blank. The bid is simply an affirmative response to the announcement and the contract is awarded to the first bidder, which does the required triangulation to obtain the position of the vehicle.

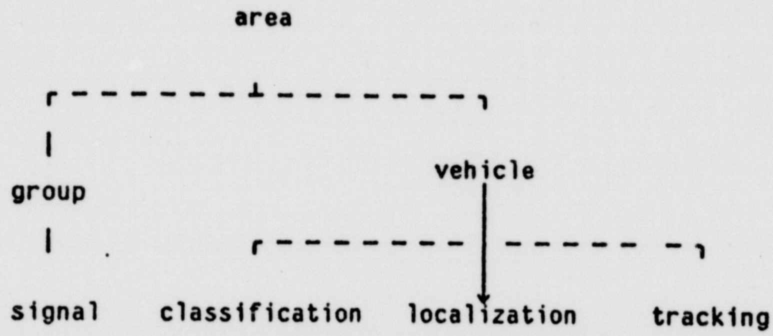

Figure 3.9. Localization Contract Initiation.

Once the vehicle has been localized, it must be tracked. This is handled by the vehicle contractor by entering into follow-up localization contracts from time to time and using the results to update its vehicle description (Figure 3.10). As an alternative, the area contractor could award separate tracking contracts. The decision as to which method to use depends on loading and communications. If, for example, the area contractor is very busy with integration of data from many group contractors, then it seems more appropriate to isolate it from the additional load of tracking contracts. If, on the other hand, the area contractor is not overly busy, then we can let it handle updated vehicle contracts, taking advantage of the fact that it is in the best position to integrate the results and co-ordinate the efforts of multiple tracking contractors. In this example, we assume that the management load would be too large for the area contractor.

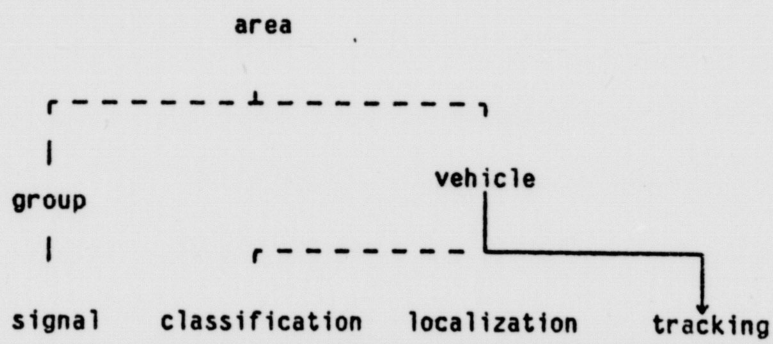

Figure 3.10. Tracking Contract Initiation.

There are a variety of other issues that must be considered in the design and operation of a real distributed sensing system. Most of these are quite specific to the DSS application, and would take us away from our primary concern with the use of the contract net framework. One issue, however, presents an interesting example of the relative utility of the use of contracting compared to the use of information messages. Consider the case of a vehicle that is moving from the area of one area contractor to that of another. How is responsibility for tracking the vehicle to be transferred?

There are two possibilities. First, the area contractor that is currently responsible for the vehicle could send out an information message to neighboring area contractors. This message would serve to set up expectations in the neighboring area contractors that a vehicle was about to pass into one of their areas. The processing is fully decentralized in this approach.

Second, the area contractor currently responsible for the vehicle could send a report to its manager (the monitor node) that contains the same information. The monitor node could then award a new contract to one of the neighboring area contractors to handle the vehicle when it passes out of the original area. This is a hierarchical control approach to the problem. It entails more (centralized) processing by the monitor node.

With the first approach there is no guarantee that another area contractor will pick up on the information message and immediately take responsibility for the vehicle, where the second approach does provide such a guarantee. The lack of guarantee is not generally serious, but may result in excess processing because the vehicle must be re-detected in the new area, classified, localized, and so on. The tradeoff is thus one of more processing by the monitor node against (possibly) more processing by nodes in the new area. In the simulation we have used the hierarchical control approach because of its guarantee of transfer of responsibility.

4 Conclusion

The use of the contract net framework in the DSS enables the implementation of a dynamic configuration, depending on the actual positions of sensor and processor nodes and the ease with which communication can be established. Such a configuration offers a significant operational improvement over a static a priori configuration; specifically, it ensures that nodes that must cooperate for the solution of the area surveillance problem are able to communicate with each other. This avoids the necessity for either indirect routing of messages or powerful transmitters for all nodes.

The distributed control enabled by the framework enhances both reliability and equalization of the processing load. It is also possible to recover from the failure of nodes, because there are explicit links between nodes that share responsibility for execution of tasks (managers and contractors). The failure of a signal contractor, for example, can be detected by its manager, and the contract for which it was responsible can be re-announced and awarded to another node.

The framework is also advantageous for this problem because it enables addition of new nodes to the net, even after operation has commenced. This is possible for two reasons. First, the nodes communicate in a common protocol. This protocol enables a new node to interpret the task-independent portions of messages (e.g., that a particular message specifies a task to be executed). Second, the protocol is augmented by a common internode language. This enables a new node to identify, for example, the specific information that it must have to execute a particular task. The protocol and language also enable a new node to request this information from other nodes in the net.

We have shown the effectiveness of interactive mutual decisions (by those with tasks to be executed and those in a position to execute the tasks) in distributing tasks throughout the system. This was used, for example, in setting up signal contracts. The potential signal contractors used the information in the signal task announcements to select the closest managers to which to respond. The managers in turn used the number of sensors and the distribution of sensors in the returned bids to make a final selection of signal contractors.

We have also noted the ways in which the contract net protocol helps to reduce message traffic and message processing overhead--through the use of task abstractions, eligibility specifications, and bid specifications in task announcements, through the use of focused addressing, and through the use of specialized interactions like directed contracts and requests.

The contract net framework is in general applicable to problems that use a hierarchy of tasks and levels of data abstraction. The manager-contractor structure provides a natural way to effect hierarchical control (in the distributed case, it's actually concurrent hierarchical control), and the managers at each level in the hierarchy are an appropriate place for data integration and abstraction. It should also be noted that the control hierarchies in the contract net framework are not simple vertical hierarchies, but are the more complex generalized hierarchies discussed by [Simon, 1969]. The manager-contractor links are not the only means of communication. Nodes are able to communicate horizontally with related-contractors or indeed any other nodes in the net, as we saw in the DSS example, where classification contractors were able to communicate directly with signal contractors.

The announcement - bid - award sequence of contract negotiation enables more information, and more complex information to be transferred in both directions (between caller and respondent) before KS-invocation occurs. The computation devoted to the selection process, based on the information transfer noted above, is more extensive and more complex than that used in traditional approaches, and is *local* in the sense that selection is associated with and specific to an individual KS (rather than embodied in, say, a global evaluation function). As a result, the framework is most useful when the specific KS to be invoked at any time is not known a priori, and when specific expertise is required.

It also follows that the framework is also primarily applicable to domains in which the subtasks are large (in the loose-coupling sense), and in which it is worthwhile to expend a potentially non-trivial amount of computation and communication to invoke the best KSs for each subtask, so as to prevent backtracking as much as possible.

Appendix A

DSS Sample Messages

This appendix includes abbreviated sample messages for the signal task in the DSS example. For brevity, the messages shown contain only the information mentioned in Section 2. Terms written in upper case are included in the core internode language, while terms written in lower case are specific to the DSS application.

Italicized statements are commentary about the content and sequence of messages.

<Announcements of the following form are transmitted by the various group contractors.>

To: * *<"*" indicates a general broadcast.>*
From: node-sg1
Type: task announcement
Contract: s
Message:

<Needed - signal data for traffic. My position is p. If in possession of sensors, and located in area A, respond with position, and type and number of sensors.>

<u>Task Abstraction:</u>

TASK TYPE signal
NODE NAME sg1 POSITION p

<u>Eligibility Specification:</u>

MUST HAVE DEVICE TYPE sensor
MUST HAVE NODE NAME SELF POSITION area A

<u>Bid Specification:</u>

NODE NAME SELF POSITION
EVERY DEVICE TYPE sensor TYPE NUMBER

<Nodes with sensors respond to the nearest group contractor.>

To: node-sg1
From: node-s1
Type: bid
Contract: s
Message:

<u>Node Abstraction:</u>

NODE NAME s1 POSITION q
sensor TYPE S NUMBER 3
sensor TYPE T NUMBER 1

<Several similar awards are transmitted.>

To: node-s1
From: node-sg1
Type: award
Contract: s
Message:

<Report signals. Use sensors S1 and S2.>

Task Specification:
sensor NAME S1
sensor NAME S2

References

[Kahn, 1975]
R. E. Kahn, The Organization Of Computer Resources Into A Packet Radio Network. *NCC Proceedings*, Vol. 44, Montvale, N. J.: AFIPS Press, 1975. Pp. 177-186.

[Nii, 1978]
H. P. Nii and E. A. Feigenbaum, Rule-Based Understanding Of Signals. In D. A. Waterman and F. Hayes-Roth (Eds.), *Pattern-Directed Inference Systems*. New York: Academic Press, 1978. Pp. 483-501.

[Simon, 1969]
H. A. Simon, *The Sciences Of The Artificial*. Cambridge, Mass.: MIT Press, 1969.

[Smith, 1977]
R. G. Smith, The CONTRACT NET: A Formalism For The Control Of Distributed Problem Solving. *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, Cambridge, Mass., August 1977, p. 472.

[Smith, 1978a]
R. G. Smith, *Issues In Distributed Sensor Net Design*. HPP-78-2 (Working Paper), Heuristic Programming Project, Dept. of Computer Science, Stanford University, January 1978.

[Smith, 1978b]
R. G. Smith and R. Davis, Distributed Problem Solving: The Contract Net Approach. *Proceedings of the Second National Conference of the Canadian Society for Computational Studies of Intelligence*, Toronto, Canada, July 1978, pp. 278-287.

[Smith, 1978c]
R. G. Smith, *A Framework For Problem Solving In A Distributed Processing Environment*. Ph.D. Dissertation, Dept. of Electrical Engineering, Stanford University, December, 1978.

Copyright © 1985 by KSL and
Comtex Scientific Corporation

FILMED FROM BEST AVAILABLE COPY