Abstract - We report on work on a new approach to counterspace situation awareness, based on variable resolution modeling using distributed relational Bayesian networks. Variable-resolution modeling provides the proven effectiveness of Bayesian data fusion in an efficient, scalable, distributed architecture. The core representational element is the Bayesian network (BN), a graphical representation for probabilistic modeling of large, highly structured problems. A recent breakthrough, relational Bayesian networks, rests on an underlying entity-relationship data model, providing a principled foundation for constructing probabilistic situation awareness. We are developing and evaluating a distributed architecture for scalable, distributed data fusion for counterspace situation awareness. The capabilities include: (1) detection, association, tracking, and assessment of entities, relationships, and overall situations; (2) multiple-hypothesis relationship and situation modeling; (3) distributed situation modeling, robust to single-point of failure in both processing nodes and communication links; and (4) automated management of node processing and storage utilization, as well as link bandwidth utilization.

Keywords: Distributed Fusion, Counterspace, Situation Awareness, Bayesian networks

1 Introduction

The U.S. is more dependent on space than any other nation. At the same time, there are many extant capabilities to deny, disrupt or physically destroy space systems and ground facilities. As a result, recommendations have been made that we invest more heavily in improved space situational awareness and attack warning capabilities, protection/defensive countermeasures, and prevention and negation systems. The foundation of any counterspace capability is space situation awareness. Effective deployment of countermeasures relies on accurate prediction and impact assessment. Increasingly, the targets of enemy offensive counterspace operations, the sensors available to detect such operations, and the assets we can deploy against them are all space-based and distributed. However, current space situation awareness capabilities are ground-based, centralized, and human intensive. This makes them slow, bandwidth-intensive, and vulnerable to events that disrupt space/ground communications.
2 Motivation & Background

Figure 1. Foundation for Effective Counterspace

We argued above that space situation awareness should be a distributed, shared perception over the collection of sensors, command and control elements, and weapons involved. The following scenario, in which the enemy is attempting to disable satellite sensors to prevent observation of ground activity, illustrates the reasons why this is so.

- Ground-based coordinated "dazzle" occurs against ground-looking E/O imaging sensors on two platforms.
- Initial indications are satellite-as-sensor (SAS) – local platform health management reports sensor overload.
- Background intelligence indicates increased ground activity in area traversed at time of dazzle.
- Ground-based radar has been tracking unknowns slowly approaching elements of cluster.

How might distributed counterspace situation awareness respond to this sequence of events? The first indications of a problem in the above scenario occur when health management on one satellite, then on a second, report problems with E/O sensors. Noticing problems on the other satellite, each of the blinded satellites might then create a shared “coordinated attack” hypothesis, to compete with local system problem hypotheses and a shared “space weather” hypothesis. Overall situation interpretation construction might rate simultaneous system problems on two independent satellites unlikely, and there is no other evidence or report of space weather, but coordinated attack is a-priori unlikely as well, so the true situation is unclear. However, the ground node, sharing the coordinated attack hypothesis, might then attempt to locally confirm or deny coordinated attack, revealing that both satellites were visible from the same potentially hostile ground location at time of failure. Notice that little detailed information need be shared between space and ground: only likelihoods about the overall hypothesis itself. Yet, a shared belief in the coordinated attack is evolving, with each participating node knowing the sources of various contributions to that belief, should it require further details.

Next, further inquiry at the ground station uncovers reports of increased ground activity in regions controlled by the hostile. These facts, while not pointing to an unambiguous situation interpretation, raise the likelihood of coordinated attack above alternate interpretations. This triggers a wider search for possible extensions/consequences of the attack. The satellite that has been tracking the approaching unknowns forms a new hypothesis, that the unexpected rogue satellites are part of an extended attack scenario, and defensive actions are initiated in time to thwart the impending physical attack.

In this scenario, events and information needed to form a correct interpretation of the situation are distributed over the set of participating nodes. Each node has the information it needs to guide local information gathering and hypothesis formulation actions, with minimal inter-satellite or space-ground communication. Even if ground communications were lost, the space-based nodes would still be able to construct a coordinated attack that included the approaching unknown satellites, albeit at a lower confidence.

3 Distributed Counterspace SA

Situation awareness is inherently a relational task: a situation is not just a set of objects and events, but also the relationships within and among those objects and events. Recognizing events and the relationships among them often requires integrating information from multiple sensors and sources, much of which is imprecise or uncertain. We have begun development and evaluation of relational probabilistic methods for distributed data fusion in support of situation awareness, prediction, and impact assessment. Our distributed modeling technology provides the ability for each node to compose an overall representation from modular, distributed elements so that it has the information it needs at a cost (in local processing, storage and communication bandwidth) it can afford.

Our approach is to apply variable resolution modeling using distributed relational Bayesian networks. Variable-resolution modeling provides the proven effectiveness of Bayesian data fusion in an efficient, scalable, distributed architecture. The core representational element is the Bayesian network (BN), a recent graphical representation for probabilistic modeling of large, highly structured problems. We applied a recent breakthrough, relational Bayesian networks – a modular form of Bayesian networks ideally suited for dynamic tasks such as diagnosis, situation assessment, and distributed target identification. In the remainder of this section, we briefly review the
underlying technology. In the next section, we report progress to date.

3.1 Bayesian Networks

Modern graphical probabilistic representations separate description of the structure of a problem from specification of the numeric details. The discovery of a relationship between conditional independence in probability theory and the absence of arcs in a directed acyclic graph makes it possible to express much of the structural information in a domain independently of the detailed numeric information, in a way that both simplifies knowledge acquisition and reduces the computational complexity of reasoning. The resulting graphical models have come to be known as Bayesian networks.

![Figure 2. Simple Diagnosis Problem](image)

An example Bayesian network (BN) is shown in Figure 2. Intuitively, Figure 3 is a representation of the following model: a bad SensorReport can be caused by a faulty SensorStatus or a faulty SystemStatus, faulty ComponentStatus can lead to faulty SystemStatus, and can also be the cause of bad SelfCheckReport. The missing arcs encode crucial information: the chance of faulty SensorStatus is unrelated to the presence of faulty EquipmentStatus, a faulty sensor doesn’t cause incorrect SelfCheckReport, EquipmentStatus doesn’t affect SensorReport except through SystemStatus, and SensorReport and SelfCheckReport are unrelated except through their shared cause. We have described the structure of the model without specifying a single number, and have done so in an intuitive and easily understandable form. The probability model (the joint probability distribution across the parameters) is completed by specifying a set of local probability distributions, one associated with each node in the graph. Available inference methods permit one to query such a network for beliefs about the value of any variable or combination of variables in the network, in the presence of any combination of evidence [2]. Bayesian networks have transformed many areas of AI, especially including monitoring and diagnosis [5] [6], [3], [4]. An especially fruitful area has been the application of Bayesian networks to data fusion for target identification and situation assessment [1]. For example, in Figure 3, the structure of the model “explains” the correlation between the possible observables (SensorReport and BaselineReport), permitting correct computation of the fused belief despite the correlation between the observations.

BNs also provide a sound basis for decision-making and control. A graphical extension, the Influence Diagram, adds nodes representing decisions to be made and objectives the decision-maker wishes to achieve. Sound and efficient methods exist for using this information, not only to compute optimal action policies, based on the principle of maximizing subjective expected utility but also to information gathering prior to decision-making, through computation of the value of information for alternate information sources with respect to the decision problem at hand [8].

3.2 Distributed Bayesian Networks

Attempts at distributed application of the above technology (Figure 3) immediately encounter several difficulties. Obvious alternatives include: (1) duplication of the entire system model on every platform; and (2) partitioning the system model across platforms (e.g., [10]). Decision-making and control would be straightforward in the first architecture, since each platform has a global view of all others. However, one can’t run the entire system model at every node in a distributed system: sensors and effectors are local to specific nodes, and the computational requirements for a fully detailed yet global model often exceed the resources available at any one platform. On the other hand, simple partitioning of a global model among platforms fails due to the high bandwidth requirements needed to achieve globally coherent updating of the model. Local decision-making is difficult because no one platform has global knowledge. In addition, the fully distributed model has no redundancy, making recovery from platform malfunction or failure difficult.

![Figure 3. Model-based Distributed Data Fusion](image)

Our solution is to utilize modular, variable-resolution Bayesian networks to combine the advantages of both approaches while circumventing limitations. Each
platform can model those elements of the overall system needed for its tasks, at the resolution needed to perform those tasks. For example, given two platforms as in Figure 5, platform 1 may need a fully detailed model of its own power system so it can integrate internal sensor data for detailed local health assessment. On the other hand, it may need only a very high level summary view of platform 2 health in deciding whether to request platform 2’s support in a ground-target classification task.

4 Distributed Satellite Data Fusion

Variable-resolution modeling in dynamic environments requires the ability to compose an overall representation from modular elements, so that system control can dynamically select the appropriate resolution for each element to be modeled. Distributed Bayesian modeling using adaptive, variable-resolution models provides a coherent, efficient, scalable architecture for distributed real-time data-fusion.

Based on prior work in the DARPA Dynamic Databases and CyberPanel programs [1] we propose to extend an object-oriented form of relational Bayesian modeling to achieve this capability. There are two representational challenges to multiple-resolution modeling, first the ability to combine multiple element models into an overall model, and second the articulation among models at differing resolutions. Both are addressed by relational Bayesian modeling.

Traditional Bayesian networks model a situation or problem as a collection of related parameters. Absent from this representation is any notion of the objects or entities in the situation. Consider, for example, a simplified satellite-pair health monitoring task. Each subsystem on both platforms could be modeled using BN technology. However, the models would then have to be interconnected, both within a platform and across platforms, to construct an overall system model. Traditional Bayesian network technology provides no larger element than the variable. There is no notion of a “system”, so there is no formal way to describe relationships among systems or subsystems.

Also, different tasks may require different views of the overall system. For example, imagine that a task executive is running on platform 1, allocating payload tasks across platforms 1 and 2. It may need to know overall health and capability of platform 2 without knowing full detail on every platform 2 subsystem. We would like to be able to construct a summary of platform 2 status that is available to the task executive. The resulting set of models running on each platform is shown in Figure 4, where the smaller version of the platform 2 model running on platform 1 schematically illustrates that this is an abstract, summary model. Again, traditional Bayesian networks provide no capabilities to accomplish this.

Figure 4. Variable Resolution Modeling

4.1 Challenges

The challenges to realizing the above capabilities span representation, inference, communication, and control:

- **Representation**: there are two representational challenges to multiple-resolution modeling: first, the ability to combine multiple element models into an overall model; and second, the articulation among models at differing resolutions. Both are addressed by relational Bayesian modeling.

- **Inference**: once a variable resolution model is constructed, it must be distributed. We have explored the sharing of likelihoods over separator sets as a basic communication mechanism.

- **Communication**: separator sets can get large, and inter-platform bandwidth can be limited. Mechanisms are needed to decide what and when to communicate. We have developed bandwidth-agile model-based compression methods for optimizing use of satellite communication bandwidth.

- **Control**: The system needs to determine what to model, and when to update various models. We have developed decision-theoretic control mechanisms for allocating processor and communication resources.

In the remainder of this paper we review progress to date in each of these areas. The overall objectives of our initial effort were to design a distributed data fusion architecture for micro-satellite applications and to evaluate the hypothesis that variable-resolution Bayesian modeling can provide the basis for efficient, effective, scalable distributed data fusion. Our discussion will include discussion of system design, modeling, and evaluation.

4.2 Communication

This section contains the high-level design for the Variable Resolution Modeling for Data Fusion (VRM-DF) platform. VRM-DF is based on CleverSet’s Distributed Modeling and Data Fusion (DM-DF) platform. DM-DF utilizes CleverSet’s relational modeling and inference engine and provides capability for a set of loosely coupled
cooperating agents to build, maintain, and query a distributed model. The design is organized at three levels: the cluster (distributed model); the system, (element of the distributed model running on any one platform); and the unit (an encapsulation of a model instantiated in the underlying relational modeling an inference). A system can contain one or more unit models, and is responsible for articulating among them. The cluster level is responsible for inter-system communication and coordination. There are several issues to be addressed in the design of a distributed modeling capability, including: (1) what to model on each platform; (2) what to communicate among platforms; (3) how to communicate; and (4) when to communicate. This section focuses on the inter-platform communication infrastructure element of issue (3), how to communicate.

VRM-DF provides three levels of organization at each modeling level:

- **Model** – a basic wrapper for a probabilistic relational model. It provides access to basic representation and inference services.
- **Monitor** – a basic control loop for accepting sensor and other input, applying it to the model, and reporting results. Monitors are also responsible for tracking model resource consumption.
- **Manager** – an abstract class responsible for setting resource allocations and performing model instantiation and hypothesis management.

### 4.3 Modeling

The availability of entities and relations permits us to describe systems in a natural way. In our initial effort we modeled only one satellite subsystem in depth, the electrical power system, and performed only the most preliminary modeling of counterspace situations. For example, we modeled an electrical power system as having an overall status, a battery subsystem, a solar array, and one or more buses.

A battery subsystem, in turn, is composed of a set of batteries and a heater, and a battery is modeling as containing a voltage, current, and temperature, corresponding sensors, and a fault state. Each model is composed of two elements: a schema or class description and a relational Bayesian model describing relationships among attributes of the schema elements. In Figure 6, we show the graphical view showing the overall conditional independence structure of the model (upper left); (2) the quantitative conditional probability table associated with one of the model parameters (lower left); and (3) a textual view, showing the graph using the BNIF xml representation for directed graphical probabilistic models (right).

**Figure 5. VRM-DF Organizational Levels**

**Figure 6. Battery Relational Bayesian Model**

The **entities** in the above model include Battery, Voltage, Current, and Temperature. The **relationships** are implicit in the structure of the above diagram: the battery fault status, its voltage, current, and temperature are related through the role each serves in the overall model. This modularity of relational Bayesian modeling permits easy substitution of appropriate modeling elements.

The complete current draft set of probability models over the current ontology and schema is shown in Figure 7.

**Figure 7. Complete Draft Model Set**
4.4 Relational Bayesian modeling and situation awareness

References such as Battery.voltage.value work well when the relations involved are one-to-one (i.e., each battery has only one modeled voltage). However, relations can be one-to-many or many-to-many. For example, the batteries relation for a BatterySubsystem relates a BatterySubsystem instance to a set of component battery instances. Relational Bayesian modeling provides aggregation and selection operators modeling dependencies over these types of relationship. For example, the relationship between an overall BatterySubsystem status and the voltages and currents of its component Batteries is modeled using the Mean aggregator. (i.e., mean battery voltages and currents in the normal range indicate a status of “ok”, and vice-versa).

Another example use of aggregators and selectors appears in the upper right of Figure 9, where the existence of a cluster space-weather problem is correlated with the Min (aggregator) of the existence of problems on cluster satellites, where the problems of interest are those of type=weather (selector). (Again, these are intended only to be suggestive, we have barely begun our development of counterspace situation models.)

These operators permit the definition of models at multiple resolutions and articulation among them. For example, let’s return to our challenge of constructing a comprehensive multi-satellite system model on satellite 1, for use by the task executive running on that satellite. Assume we have a detailed power system model for each satellite. We don’t want to duplicate the detailed system model for satellite 2: communication bandwidth requirements to send all the sensor data would be too high, as would computational requirements to keep the redundant model current. However, we can define a lower resolution model that contains the key information needed by the task executive, namely the overall power system status on satellite 2. We can accomplish this simply instantiating on satellite 1 only the overall ElectricalPower model for satellite 2. The communication needed to keep this model current is automatically established and maintained by our distributed communication and control layers, described later.

4.5 Scenario demonstration

Consider the scenario introduced earlier: one satellite, then a second, experiences sensor overload, while SOPS sees no evidence of untoward space weather. The screen shot below shows the final fused analysis of the above scenario phases of the above scenario, as currently implemented. The two panels at the top of the screen shows the two satellites, the lower right panel shows the ground SOPS.

Figure 8 shows conclusions following a ground update regarding the absence of adverse space weather reports. As can be seen, ground is now confident that there is a coordinated attack under way, and each satellite is confident that it is under attack, although some probability is reserved for the possibility that any one satellite may be experiencing local weather effects.

5 Performance Evaluation

In order to obtain scalable, generalizable results, we decided to use an abstract problem to evaluate bandwidth/accuracy trade-offs in distributed fusion. In particular, we chose a bipartite graph, where the root nodes are unobserved and the leaf nodes are observed. An example graph is shown in Figure 9 below.

The network in Figure 9 is an instantiation of the canonical “simplest” multiple-cause causal model, where the letter nodes represent unobserved root causes and the numbered nodes represent observed symptoms. Real models such as those we are developing for counterspace scenarios have considerably more structure, but we believe results obtained for this topology will apply to more realistic topologies. This simple topology provides a smooth experimental space in which we can scale a number of interesting parameters, including variable
domain size, number of causal nodes, number of evidence nodes, average number of incoming arcs to evidence nodes, and number of platforms. The two boxes above illustrate the two-platform case, where one platform observes evidence nodes 1-4 and the second platform observes evidence nodes 5-8.

5.1 Bandwidth

In our evaluation design, we compared the accuracy of the posterior distribution across the unobserved nodes (a-f in Figure 9), given observations on the evidence nodes (1-8 in Figure 9), as the experimental parameters above, as well as the allowed bandwidth, vary. The base case for comparison is always a single, centralized platform observing all evidence nodes. In our initial experiment, we tested a 2-platform case with a network containing 10 cause nodes, 14 evidence nodes (7 per platform) and all nodes having a domain size of two. We evaluated the accuracy of distributed fusion by computing the KL-divergence between the centralized platform and the two-platform solution, using the worst-case of the two platforms (i.e., the one with beliefs furthest from the true values). Table 1 below shows the results of our initial evaluation. It should be noted that KL-divergence of the joint distribution is an extremely demanding evaluation criterion – many decisions depend only on individual marginal probabilities, and few if any depend on the full joint. However, this is a good indicator of the ability of an individual platform to construct a full, instantiated, situation description.

Table 1: KL-Divergence between distributed and centralized fusion as a function of bytes communicated.

<table>
<thead>
<tr>
<th>Bandwidth Penalty</th>
<th>Number of Probabilities Communicated</th>
<th>KL-Divergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2040</td>
<td>0.000006</td>
</tr>
<tr>
<td>1</td>
<td>224</td>
<td>0.005</td>
</tr>
<tr>
<td>1000</td>
<td>176</td>
<td>0.0039</td>
</tr>
<tr>
<td>10000</td>
<td>118</td>
<td>1.55</td>
</tr>
<tr>
<td>25000</td>
<td>74</td>
<td>2.61</td>
</tr>
<tr>
<td>50000</td>
<td>52</td>
<td>19.5</td>
</tr>
</tbody>
</table>

Notes:
- The minimum possible communication for this problem, the set of marginal probabilities across the nodes involved, would be 40 probabilities.
- The average KL-divergence between a set of randomly generated distributions and the true joint posterior was 16.76, indicating that the last row in the above table, the most extreme compression, was worse than random!
- The KL-divergence of the “0” bandwidth penalty case is measuring of floating-point roundoff error.

We see from the above that, even for this small initial study problem, our compression methods achieve over an order of magnitude reduction in communication bandwidth with minimal loss of accuracy. Perhaps more significant, the penalty 1000 case is only 4 times the minimal communication size, again with minimal accuracy loss.

5.2 Distributed alerting

In this study, we compared the performance of centralized versus distributed fusion on a simple alerting task. In this task, we used the networks described above, and considered a cause node “faulted” if the posterior probability of its fault state was higher than that of its true state, given the evidence. Evidence was generated by randomly assigning a subset of the cause nodes to false and computing the posterior probabilities of the evidence nodes. The “bandwidth penalty” was set to unity for this task. Since the conditional probabilities of evidence given cause were deterministic, this led to a difficult diagnostic task in which causes were easily confused, even in the case of centralized fusion. The results are presented in Table 2 as confusion matrices. We can see from the preliminary results in Table 2 that performance of the distributed platform on a simple alerting task is comparable to that of centralized fusion.

Table 2. Comparison of Centralized and Distributed Fusion on a Simple Alerting Task

<table>
<thead>
<tr>
<th></th>
<th>Centralized Platform</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>True Positive: 31</td>
<td>False Positive: 8</td>
</tr>
<tr>
<td></td>
<td>False Negative: 13</td>
<td>True Negative: 148</td>
</tr>
<tr>
<td></td>
<td>Overall correct: 89.5%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Distributed Platform 1</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TP: 26</td>
<td>FP: 11</td>
</tr>
<tr>
<td></td>
<td>FN: 20</td>
<td>TN: 143</td>
</tr>
<tr>
<td></td>
<td>Overall correct: 84.5%</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Distributed Platform 2</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TP: 19</td>
<td>FP: 14</td>
</tr>
<tr>
<td></td>
<td>FN: 24</td>
<td>TN: 143</td>
</tr>
</tbody>
</table>

1 A confusion matrix compares the probabilities of true positive and true negative (on the diagonal) with those of false positives and missed detections (off diagonal).
6 Conclusions

Our objectives were to design a distributed data fusion architecture for satellite constellation data fusion applications and to evaluate the hypothesis that variable-resolution Bayesian modeling can provide the basis for efficient, effective, scalable distributed data fusion. Evidence to date supports the effectiveness of distributed variable-resolution modeling for distributed fusion and situation awareness tasks. In controlled studies, results from distributed modeling and fusion were very close to those from centralized fusion, on both inferential and decision tasks, while communication costs remained low.

While we did complete the distributed data fusion architecture design, study problems used during this initial study were too small-scale to fully exercise the architecture. In particular, the "system" level, introduced to manage computation on a single platform, was unnecessary. We remain confident, however, that the capabilities designed at that level will be needed for larger problems. At the same time, as the design proceeded we identified an important element of the distributed architecture that we were not able to fully address within the scope of the initial effort: distributed name-space management. Part of this task is well-understood in the distributed computing community, where it is seen as a name-space management problem (i.e., name-space management for pre-existing objects such as satellite subsystems). Another part is understood by the tracking and fusion community under the heading of track correlation (i.e., name-space management for dynamically detected objects such as targets).

While computational costs of computing compact communication representations were low for our test problem, we believe these may be a bottleneck for full-scale application of the techniques, and so this should be an area of further research. Similarly, the distributed name-space problem should be a focus of future effort. It provides the opportunity to unify distributed computing and fusion perspectives and methods for object recognition and identification, potentially to the benefit of both.

References


