PROGNOS: Predictive Situational Awareness with Probabilistic Ontologies

Rommel N. Carvalho, Paulo Cesar G. Costa, Kathryn B. Laskey, KC Chang
Center of Excellence in C4I / The Sensor Fusion Lab
George Mason University, MS 4B5
Fairfax, VA 22030-4444 U.S.A.
rommel.carvalho@gmail.com, [pcosta,klaskey,kchang]@c4i.gmu.edu

Abstract – *Information in the battlefield comes from reports from diverse sources, in distinct syntax, and with different meanings. There are many kinds of uncertainty involved in this process, e.g., noise in sensors, incorrect, incomplete, or deceptive human intelligence, and others, which makes it essential to have a coherent, consistent, and principled means to represent such phenomena among the systems performing Predictive Situation Awareness (PSAW). PROGNOS is a PSAW system being developed to work within the operational context such as U.S. Navy's FORCENet. It employs probabilistic ontologies in a distributed system architecture as a means to provide semantic interoperability within an intrinsically complex and uncertain environment. This paper explores our current status in developing the system while addressing the major research challenges for making an effective PSAW system to support maritime operations.*

Keywords: probabilistic reasoning, predictive situational awareness, web services, Bayesian networks, MEBN, PR-OWL, probabilistic ontologies, distributed hybrid inference, spatio-temporal hybrid analysis.

1 Introduction

Current technology being employed in modern maritime warfare provides lower-level multi-sensor data fusion through stove-piped systems, which are conveyed to warfighters via high bandwidth Net-Centric systems. As an example, the US Navy's Net-Centric infrastructure, FORCENet [1], can manage thousands of tracks through many and diverse sensors. However, the tacit assumption of today's systems is that humans are responsible for translating the massive amounts of incoming data into knowledge that is useful for their purposes. Assigning to humans the tasks of performing higher-level fusion, creating situational awareness and conducting their own predictive analysis is a recipe for failure. Cognitive overload, among other issues, hamper this process, preventing optimal decision-making and limiting the effectiveness of US military Forces.

To address this problem, our current research on the PROGNOS system (Probabilistic OntoloGies for Net-centric Operation Systems) [2][3] aims to provide consistent higher-level fusion through state-of-the-art knowledge representation and reasoning, as well as to enable predictive analysis with principled hypothesis management.

To present our ideas, this paper is structured as follows. In section 2 we address the major components of PROGNOS, which will provide the necessary contextual background to the ideas supporting the development of the reasoning module, covered in section 3, and of the simulation module, addressed in section 4.

2 PROGNOS General Architecture

The major idea behind PROGNOS is to provide answers to queries from decision makers using distinct, loosely coupled information systems to feed a Bayesian reasoning process. This process must be optimized for both interoperability and operational performance. To address the first we employ Probabilistic Ontologies written in PR-OWL format [4]. To ensure the latter we employ hypothesis management and optimized reasoning algorithms for networks with both continuous and discrete nodes, such as the hybrid message passage algorithm depicted in [5]. PR-OWL is based on Multi Entity Bayesian Networks (MEBN, [6]), a first-order Bayesian logic that combines the flexibility of Bayesian networks with the expressiveness of first-order logic.

Bayesian probability provides a mathematically sound representation language and formal calculus for rational degrees of belief, which gives different agents the freedom to have different beliefs about a given hypothesis. This provides a compelling framework for representing uncertain, incomplete knowledge that can come from diverse agents.

MEBN logic represents the world as consisting of entities that have attributes and are related to other entities. Knowledge about the attributes of entities and their relationships to each other is represented as a collection of MEBN fragments (MFrags). An MFrag represents a conditional probability distribution for instances of its resident random variables given their parents in the fragment graph and the context nodes. An MTheory is a set of MFrags that collectively satisfies consistency constraints ensuring the existence of a unique joint probability distribution over instances of the random variables represented in each of the MFrags within the set. The set of MFrags encoding the knowledge of a given domain is called an MTheory and is stored in PR-OWL format, which can be thus defined as an upper ontology on probabilities based on MEBN

PR-OWL classes and properties allow the ontology engineer to fully specify a MEBN model while maintaining compatibility with the widespread, W3C recommended OWL ontology language. As an ontology language, PR-OWL has support for logical reasoning and thus provides reasoning capabilities such as inferring the class structure from the asserted properties. Therefore, the inherent flexibility of PR-OWL's knowledge modeling paradigm is a good fit for a distributed PSAW system, in which information may come from many systems, possibly following distinct data schemas and semantics.

The PROGNOS general architecture is depicted in Figure 1. Each module is being developed at a different pace as research progresses. Currently, the primary development focus is on the Simulation and Reasoning modules.

To realize the above architecture we are using the UnBBayes-MEBN framework [7], which implements a MEBN reasoner capable of saving MTheories in PR-OWL format. In this paper, we will focus on the advances we have been achieving in two of the most important modules within this architecture: the reasoning module and the simulation module.

3 The Reasoning Module

The reasoning module is the heart of the PROGNOS system, responsible for performing all of its reasoning services. It is composed of a MEBN reasoner that interacts with the other modules and coordinates the execution of the Situation Specific Bayesian Network (SSBN) construction, which includes interleaved hypothesis management and inference within the constructed SSBN.

The SSBN is the minimum network necessary to answer a specific query given the information available. In other words, the MTheory works as a template to generate a Bayesian network (BN) given the question posed and the information available. Once the SSBN is constructed, all inference is done as it would be in a regular BN. The probabilistic reasoner is a key component of the Reasoning Module. As previously stated, we are actively developing it as an extension of the open source, Java-based Bayesian package UnBBayes, called UnBBayes-MEBN. In the following, we will discuss the main advances we have made in our reasoning module, the design and implementation of the inference and evaluation sub-modules.

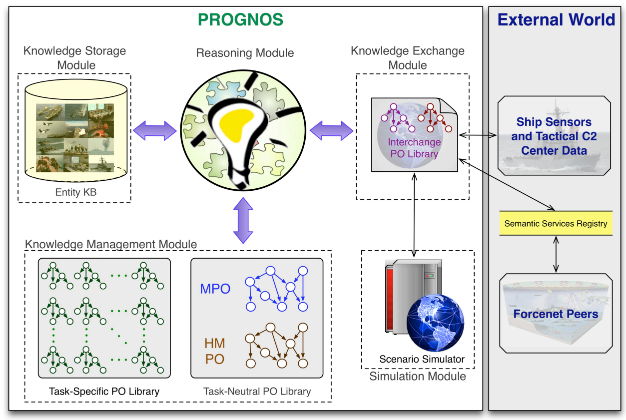

Figure 1. Distributed Predictive Situation Assessment and Impact Assessment – Component Architecture.

3.1 The Inference Sub-Module

The inference sub-module was based on our implementation of SSBN algorithm described in [6]. Figure 2 show an actual SSBN built using the implemented algorithm. To ensure the correctness of the implementation, we developed a set of test cases in which the expected results (i.e. obtained by manual procedure) were compared to those generated by our implementation. The comparison didn't produce any discrepancies related to the algorithm implementation, but it did point out logical inconsistencies in our testing models. In other words, the correct implementation of the algorithm in UnBBayes proved to be helpful in finding flaws in our initial models. This is effective and particularly desirable when modeling in highly expressive languages such as MEBN, which typically yields complex networks such as the one depicted in Figure 2. To mitigate scalability problems due to complexity of the resulting SSBNs, we implemented alternatives to standard exact inference algorithms.

Exact methods for belief updating in BNs commonly demand space requirements that cannot be met by the hardware available, due to the size of the network [8]. When this is the case, the most common solutions are to tradeoff between execution time and memory required, or to trade model size for accuracy by using an approximate inference method such as stochastic simulation.

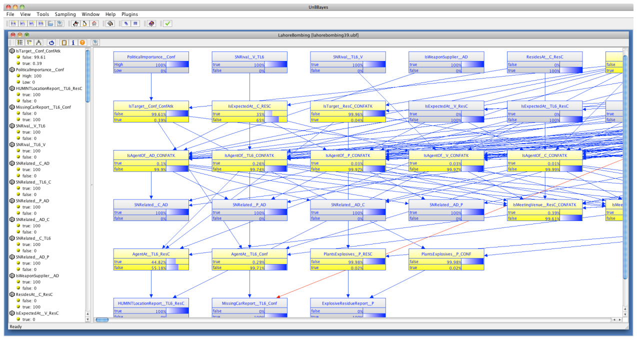

Figure 2. Result of SSBN construction based on [6].

In our development, we addressed this issue by implementing Logic Sampling, Likelihood Weighting, and Gibbs Sampling. Figure 3 shows a graphical depiction of the sub-module's operation.

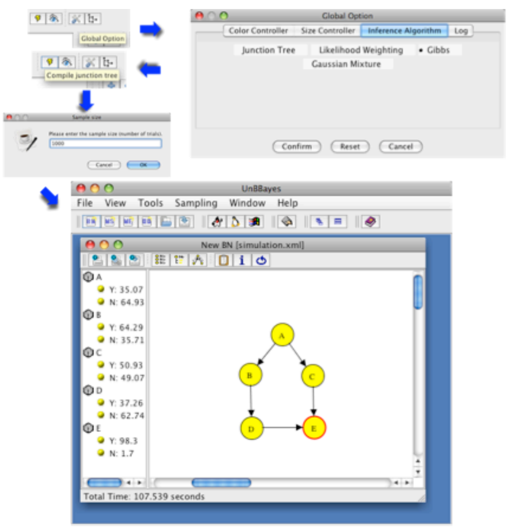

Figure 3. Stochastic simulation in UnBBayes.

First, the user chooses which of the three implemented approximation algorithms to be used. The next step is to compile the network in order to compute the marginal probability distribution. Finally, it is possible to input domain evidence and to compute the posterior probabilities using the parameters already defined by the user.

As the complexity of the SSBN depicted in Figure 2 might suggest, the greater the number of entities involved in the model the greater the complexity of the network. Therefore, it is essential to explore locality by breaking the network into sections. Thus, we have developed and implemented an algorithm for transforming the final SSBN into a Situation Specific Multiply Sectioned Bayesian network (SSMSBN), a very desirable architecture for distributed inference in large Bayesian networks and an important step to ensure scalability in our framework. This SSMSBN implementation involved designing an algorithm to leverage the properties of the MFrags of the MTheory to break down the resulting SSBN into sections that can be seen as instantiations of MFrags. From that point on, we can apply standard MSBN inference algorithms, as described in [9], to allow for an efficient query process. UnBBayes already had an algorithm for MSBN inference, which simplified our implementation process. Figure 4 shows a SSMSBN created via the algorithm.

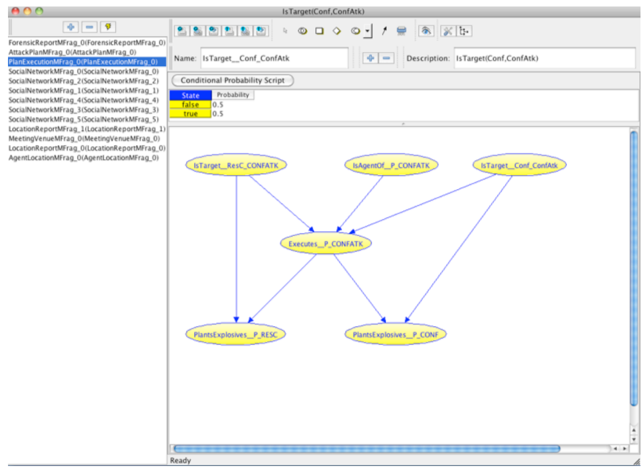

Figure 4. Network generated by the MSSSBN algorithm.

To further ensure scalability, we also carried out empirical studies on the performance of the SSBN construction algorithm. A major motivation for these studies is the fact that the alternatives created to handle scalability are only applied after the algorithm creates the SSBN. In other words, the performance obtained with the approximation algorithms and with the SSMSBN construction algorithms discussed above is only relevant for the inference on the BN generated by the SSBN algorithm. Therefore, it is crucial to ensure that the SSBN algorithm is scalable.

In our empirical studies we used the simulation module described in Section 4 to generate a sizable scenario of 50 ships, 500 people, and 25 organizations. This scenario was then used as the ground truth about those entities and to simulate reports we would get from different sources. The next step was to pose query of the type *IsShipOfInterest*(*shipX*) against the system for all the ships within the scenario. These queries, as explained before, trigger the SSBN construction processes that generate the networks needed to answer each query. For every query we have registered the time it took to generate the SSBN and the number of nodes in the network. Figure 5 shows the number of nodes in the network with the time it took to generate it. As can be seen from the graph, the time to generate the network seems to be linear with respect to the number of nodes in the network. In fact, in 48 out of 50 queries the algorithm performed the task in $.5n$ seconds where n is the number of nodes. This was roughly true except for the two worst cases, in which it took significantly more time.

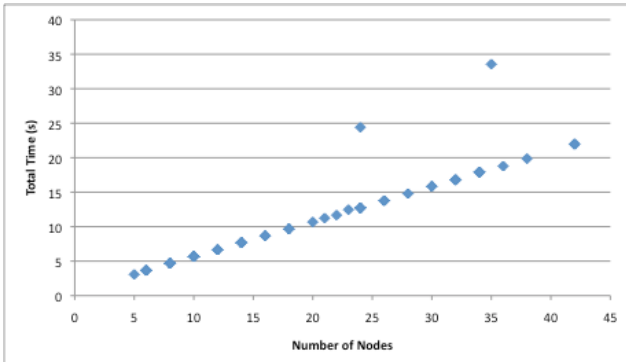

Figure 5. Time to generate the SSBN versus the number of nodes on the generated network.

Even though the empirical results look promising, we still need to conduct an analytical study to validate them. Another pending issue is that although the time is linear to the number of nodes on the generated network, the number of nodes on the network might not be linear in the number of entities involved. Therefore, we are currently performing empirical studies to assess the algorithm behavior as the number of entities involved changes.

3.2 The Evaluation Sub-Module

In PROGNOS, one or a set of random variables in the final SSBN will ultimately answer the queries posed to the system, which can be seen as most likely state of the query node given the information available. This query answering process can also be seen as a classification problem. Therefore, in addition to assessing the probability of a particular random variable to be in a given state, we must have a means of evaluating how accurate that assessment is. The evaluation sub-module is meant to provide not only the likelihood of the states of a random variable, but also the probability of a correct classification for that random variable given the information available. Figure 5 shows the sub-module in action.

In engineering systems, fusion methods have been particularly important in providing system capabilities with multiple sensors that go significantly beyond those of single sensor systems. Multi-sensor data fusion allows for the combination of information from sensors with different physical characteristics, enhancing the understanding of the surroundings and providing the basis for planning, decision-making, and control of autonomous and intelligent machines. It combines information from multiple sensors and sources to achieve inferences that are not feasible from a single sensor or source.

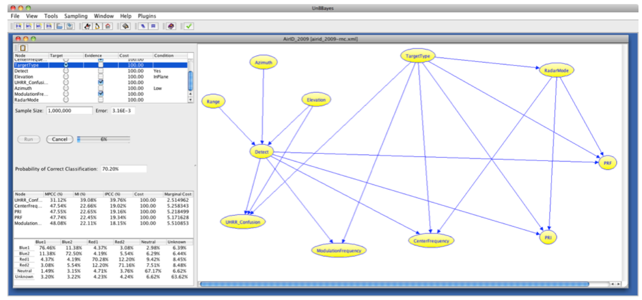

Figure 5. PROGNOS Evaluation sub-module.

In general, although one would expect a fusion system to perform better than any of the individual sensors, it is not clear as how to design such a system in a synergistic fashion. Thus, a systematic approach to evaluate the overall performance of the system is indispensable; since a weakly designed fuser could possibly make the system perform worse than its individual sensors.

To conduct the performance evaluation for the generated SSBN, we rely on the tool developed by Carvalho and Chang [10], which allows a user to evaluate the classification performance of a multisensor fusion system modeled by a Bayesian network. Specifically, the system is designed to answer questions such as: (i) What is the probability of correct classification of a given target using a specific sensor individually? (ii) What if a specific set of sensors together is used instead? (iii) What is the performance gain by adding another sensor to this set? and (iv) Which sensors provide a better cost/benefit ratio? These questions are answered based on the probability of correct classification that can be analytically estimated using Bayesian inference with the given sensor models defined by confusion matrices.

4 The Simulation Module

The simulation module is key to evaluating the architecture and algorithms as research progresses. The module generates simulated scenarios, including entities (e.g., ships, people) and their features, which serves as the ground truth for evaluating performance. It also generates reports of the kind the eventual operational system is expected to receive, thus exercising the interfaces and the reasoning module in a realistic manner.

We have made substantial advances in the development of the simulation module. We began by building a simplified agent-based simulation, which uses intelligent agents to stochastically simulate vessels with different behaviors. Also, we developed the overall concept of operations for the module, which includes the main features and system usage details.

The simulation is based on maritime activities (regular and suspicious) with the objective of prevention and disruption of terrorist attacks, sabotage, espionage, or subversive acts. Therefore, the agents on the simulation tool simulate commercial, fishing, recreational, and other types of ships in their normal and suspicious behaviors. Suspicious behaviors are characterized by ships that do not follow their regular or most probable routes according to their origin and destination, by ships that meet on the middle of the ocean for no apparent reason, etc.

The idea behind the simulation module is that although it is anchored in simulated ground truth, different sources of information will have distinct reports about individual entities (ships, crew member, etc). Therefore, the system must be capable of reasoning with masses of evidence from different domains, coming from different sources, in order to provide effective situation awareness . We seek to achieve this requirement using probabilistic ontologies (POs). One of the goals driving the development of PROGNOS' POs is to support the query process assessing whether a ship is a ship of interest given the information available. In other words, if there is evidence that a ship is suspicious, and it reaches a predefined likelihood threshold, then it will be classified as a ship of interest. POs should include likelihood ratios, relationships among maritime entities, and other aspects that are essential to support this process. We apply the methodology described by Carvalho *et al.* [11] approach for modeling a PO and using it for plausible reasoning.

The Uncertainty Modeling Process for the SW (UMP-SW) presented in Figure 6 is divided into three steps: First we have to model the domain, then we need to populate the KB, and finally we can use the model and KB for reasoning. The modeling step consists of three major stages: Requirements, Analysis & Design, and Implementation. These terms are borrowed from the Unified Process (UP) [12] with some modifications to reflect our domain of ontology modeling instead of development process. The methodology described here is also consistent with the Bayesian network modeling methodology described by [13] and [14].

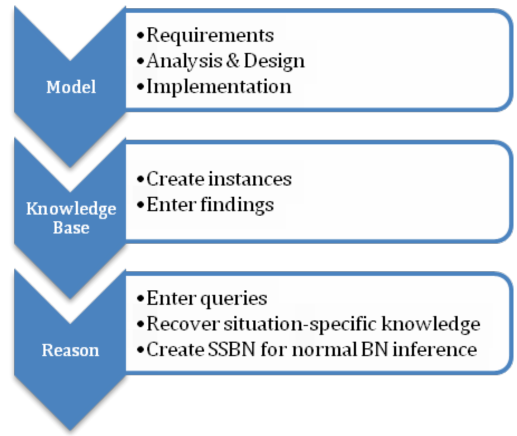

Figure 6. Uncertainty Modeling Process for the SW (UMP-SW) using MEBN.

Figure 7 depicts these three stages of the iterative and incremental Probabilistic Ontology Modeling Cycle (POMC). The basic idea behind iterative enhancement is to model our domain incrementally, allowing the modeler to leverage on what was being learned during the modeling of earlier, incremental, deliverable versions of the model. Learning comes from discovering new rules, entities, and relations that were not obvious in the past, which can give rise to new questions and evidence that might help us achieve our previously defined goal as well as give rise to new goals.

Figure 7. Probabilistic Ontology Modeling Cycle (POMC).

In the POMC depicted in Figure 7, the Requirements stage (blue circle) defines the goals that must be achieved by reasoning with the semantics provided by our model. The Analysis and Design stage describes classes of entities, their attributes, how they relate, and what rules apply to them in our domain (green circles). This definition is independent of the language used to implement the model. Finally, the Implementation stage maps our design to a specific language that allows uncertainty in the SW, which in this case is PR-OWL (red circles).

The objective of the requirements stage is to define the objectives that must be achieved by representing the domain semantics and reasoning with the representation. At this stage, it is important to define the questions that the model is expected to answer (i.e., the queries to be posed to the system being designed). For each question, a set of information that might help answer such question (evidence) must be defined. In our domain we have the following set of goal/queries/evidence:

Identify if a ship is a ship of interest, i.e. if the ship seems to be suspicious in any way.

1. Does the ship have a terrorist crewmember?
   a. Verify if a crewmember is related to any terrorist;
   b. Verify if a crewmember is associated with any terrorist organization.
2. Is the ship using an unusual route?
   a. Verify if there is a direct report that the ship is using an unusual route;
   b. Verify if there is a report that the ship is meeting some other ship for no apparent reason.
3. Does the ship seem to exhibit evasive behavior?
   a. Verify if an electronic countermeasure (ECM) was identified by a navy ship;
   b. Verify if the ship has a responsive radar and automatic identification system (AIS).

The primary hypothesis is shown in the *ShipOfInterest* MFrag in Figure 8.

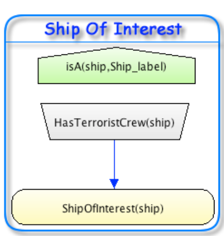

Figure 8. MFrag for identifying the ship of interest.

The hypothesis related to the identification of a terrorist crewmember is presented in the *HasTerroristCrew*, *TerroristPerson*, and *ShipCharacteristics* MFrags in Figure 9.

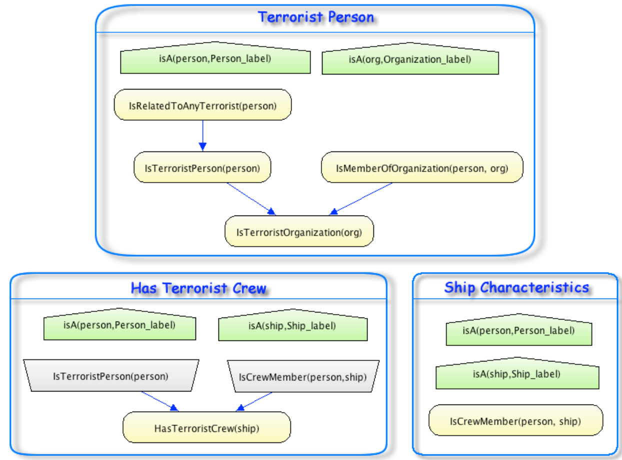

Figure 9. MFrags for identifying a terrorist crewmember.

The hypothesis related to the identification of unusual routes is presented on the *UnusualRoute* and *Meeting* MFrags in Figure 10.

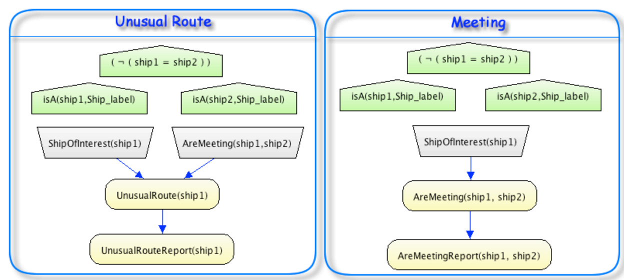

Figure 10. MFrags for identifying the ship with unusual route.

Finally, the hypotheses related to identification of evasive behavior is shown in the *EvasiveBehavior*, *EletronicsStatus*, and *Radar* MFrags in Figure 11.

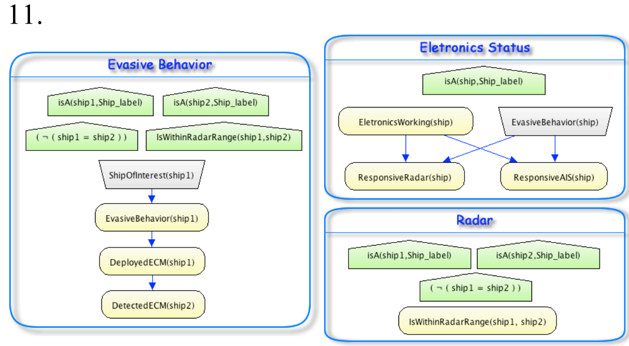

Figure 11. MFrags for identifying the ship with evasive behavior.

4.1 The Sampling Sub-Module

The method adopted to generate the ground truth for the simulation scenario is based on stochastic sampling. For the scenario "Ship of Interest" we have used Procedure 1, the BN shown in Figure 12, and the stochastic sampling method similar to the one presented in Figure 13. The procedures *createPeo-*

ple, *createOrganizations*, and *createShips*, only instantiate the number of entities desired. The procedure *createSocialNetwork* creates random relations between the people instantiated, which can be represented as a graph, generating a social network.

Procedure 1 simulateGroundTruth

Input: bn – Bayesian network used for sampling ground truth
nPeople – number of people to generate
nOrgs – number of organizations to generate
nShips – number of ships to generate

Output: db – data base containing the ground truth

```
db = new DataBase();
people = createPeople(nPeople);
orgs = createOrganizations(nOrgs);
createSocialNetwork(people);
db.addPeople(people);
db.addOrganizations(orgs);
ships = createShips(nShips);
createShipsCrew(people, ships);
db.addShips(ships);
for person in people do
        simulatePersonGroundTruth(person, bn);
end for
for org in orgs do
        simulateOrganizationGroundTruth(org, bn);
end for
for ship in ships do
        simulateShipGroundTruth(ship, bn);
end for
return db;
```

The stochastic sampling method is used on procedures *simulateShipGroundTruth*, *simulateOrganizationGroundTruth*, and *simulatePersonGroundTruth*. For the person ground truth, the random variables *IsRelatedToAnyTerrorist_person* and *IsTerroristPerson_person* from the BN in Figure 12 are used.

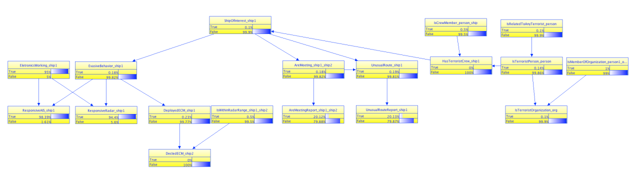

Figure 12. BN generating the simulation ground truth.

For the organization ground truth, the random variables *IsTerroristPerson_person*, *IsMemberOfOrganization_person1_org*, and *IsTerroristOrganization_org* from the BN in Figure 12 are used. For the ship ground truth, all the other random variables not yet mentioned from the BN in Figure 12 and *IsTerroristPerson_person* are used.

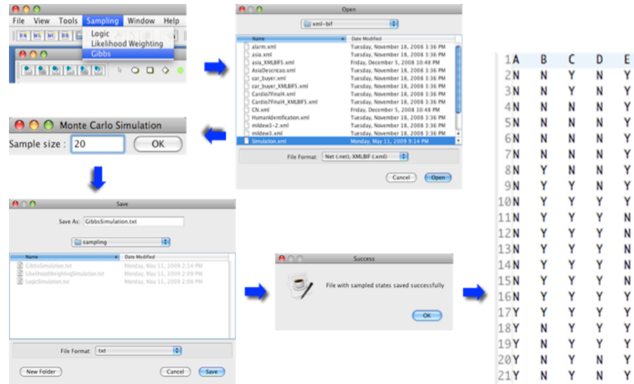

Figure 13. Stochastic sampling method used to generate the simulation ground truth.

4.2 The Scenario Execution Sub-Module

Figure 14 shows the PROGNOS simulation module running a scenario with 50 ships, 500 people (crew members), and 25 organizations. Each ship follows a random pattern based on a specific behavior.

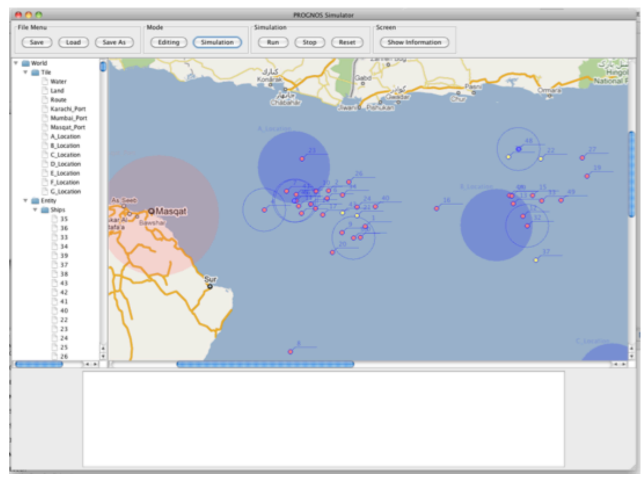

Figure 14. PROGNOS Simulation Module.

The simulation system then generates a "ground truth" that is randomly sampled and passed to the reasoning module, which uses this partial data as its knowledge base. Figure 15 shows the SSBN generated from a query posed against the MTheory depicted in Figures 8-11 and using the data extracted from the scenario in Figure 14.

The right panel of the window shows the nodes of the SSBN, which can be analyzed as any Bayesian network. The flow of events happening in PROGNOS when providing a maritime Common Operational Picture (COP) can be summarized in the following process: 1) A collection of MFrags is stored as a set of POs in PR-OWL format; some domain-specific (e.g. task-related ontologies), some conveying aspects of general knowledge (e.g. probabilistic mappings between ontologies); 2) upon receiving a query (e.g. a request for information about a given vessel), the system starts a SSBN construction algorithm, which will look for existing information and define which sensors (or systems) should be queried to provide extra knowledge; 3) after receiving this extra information, an SSBN is instantiated to provide the best possible answer to the original query.

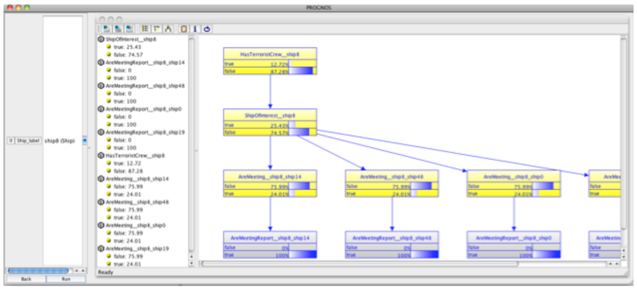

Figure 15. Result from a query to the simulated scenario.

Clearly, this is an oversimplification of all the processes happening before the final output as a SSBN, yet it provides a good picture of how a modular, composeable COP system can be built using the mathematically supported representational framework of PR-OWL/MEBN.

5 Conclusions

Although PROGNOS is still being actively under development, the partial results and insights reported in this paper are promising. It is our intention to continue developing the system and we hope this paper will bring a valuable contribution to related efforts in the Fusion community.

Acknowledgments

Research on PROGNOS has been partially supported by the Office of Naval Research (ONR), under Contract#: N00173-09-C-4008.

References

[1] M. Sharp (RADM, USN), “FORCEnet – Engineering & Architecting the Navy’s IT Future”. Presentation at the *NMCI Industry Symposium*, June 17-19, New Orleans, LA, USA.

[2] P.C.G. Costa, K.B. Laskey, and KC Chang, “PROGNOS: Applying Probabilistic Ontologies To Distributed Predictive Situation Assessment In Naval Operations.” *Proceedings of the 14th Int. Command And Control Research and Technology Symposium*, Washington, D.C., USA, 2009.

[3] P.C.G. Costa, KC Chang, K.B. Laskey, and R.N. Carvalho, “A Multi-Disciplinary Approach to High Level Fusion in Predictive Situational Awareness.” *Proceedings of the 12th Int. Conf. on Information Fusion*, Seattle, WA, USA, 2009.

[4] P.C.G. Costa, *Bayesian semantics for the Semantic Web*, Doctoral Dissertation, Volgenau School of Information Technology and Engineering, George Mason University, Fairfax, VA, 2005.

[5] W. Sun, *Efficient Inference For Hybrid Bayesian Networks*. Doctoral Dissertation. Volgenau School of Information Technology and Engineering, GMU, Fairfax, VA, 2005.

[6] K.B. Laskey, “MEBN: A Language for First-Order Bayesian Knowledge Bases”, *Artificial Intelligence*, 172(2-3), 2007.

[7] R.N. Carvalho, L.L. Santos, M. Ladeira, and P.C.G. Costa, “A Tool for Plausible Reasoning in the Semantic Web using MEBN”. In *Proceedings of the Seventh International Conference on Intelligent Systems Design and Applications*, 2007, IEEE Press, pp. 381-386.

[8] A. Darwiche, “Modeling and Reasoning with Bayesian Networks.” ISBN 978-0-521-88438-9, Cambridge University Press, New York, NY, USA, 2009.

[9] Y. Xiang, Y. Xiang, and V. Lesser, “Justifying Multiply Sectioned Bayesian Networks,” *COMPUTATIONAL INTELLIGENCE*, vol. 9, 2000, pp. 680--687.

[10] R.N. Carvalho and KC. Chang, “A performance evaluation tool for multi-sensor classification systems,” Proceedings of the 12th International Conference on Information Fusion, Seattle, Washington, USA: 2009, pp. 1123-1130.

[11] R.N. Carvalho, K.B. Laskey, P.C.G. Costa, M. Ladeira, L.L. Santos, and S. Matsumoto, “UnBBayes: Modeling Uncertainty for Plausible Reasoning in the Semantic Web,” Semantic Web, INTECH, 2010, pp. 1-28.

[12] I. Jacobson, G. Booch, and J. Rumbaugh, “The Unified Software Development Process.” Addison-Wesley Professional, ISBN 978-0201571691.

[13] K.B. Laskey and S.M. Mahoney, “Network Engineering for Agile Belief Network Models.” IEEE Transactions in Knowledge and Data Engineering 12(4): 487-498, 2000.

[14] K.B. Korb, and A.E. Nicholson, “Bayesian Artificial Intelligence.” Chapman and Hall, Boca Raton, 2003.