15TH INTERNATIONAL COMMAND AND CONTROL RESEARCH AND TECHNOLOGY SYMPOSIUM

"THE EVOLUTION OF C2: WHERE HAVE WE BEEN? WHERE ARE WE GOING?"

HYPOTHESIS MANAGEMENT IN SUPPORT OF INFERENTIAL REASONING

STUDENT PAPER

Suggested Topics:

Topic 3: Information Sharing and Collaboration Processes and Behaviors
Topic 6: Modeling and Simulation
Topic 9: C2 Architectures and Technologies

Richard Haberlin*
Paulo Cesar G da Costa
Kathryn B. Laskey

George Mason University

Systems Engineering & Operations Research Department
4400 University Dr.
Fairfax, Virginia 22030-4444

(703) – 993 – 9989

[rhaberli;pcosta;klaskey]@gmu.edu

Point of Contact: Richard Haberlin

rhaberli@gmu.edu or (571) – 291 – 2576

*Student

ABSTRACT

A key component of Maritime Domain Awareness (MDA), situational awareness supports tactical decision making through fusion of intelligence, geography, environment, and the geopolitical situation. Advanced decision support systems will provide the decision maker with a number of hypotheses from which the evolving situation may be inferred, limited by the computational capacity of today's computer hardware. In this context, a hypothesis can be thought of as a statement of anticipated action in which an actor will conduct an action against a target with a location, time and methodology of his choosing. Hypothesis Management is the control of exponential growth in fusion hypotheses created by incoming data reports, without which the computational capability of hardware is quickly overwhelmed. This paper explores our research on Hypothesis Management techniques in support of inferential reasoning. More specifically, we focus on managing the creation, modification, administration, storage and movement of hypotheses to ensure that only attributes and entities relative to the current context are presented for inferential reasoning. Our approach supports recognition of observed trends and is capable of creating original hypothesis through innovative transformations of existing hypotheses, providing the decision maker with asymmetrical scenario possibilities gleaned from observed attribute data and stored hypothesis histories.

1. Introduction

Hypothesis Management is the control of exponential growth in fusion hypotheses created by incoming data reports, without which the computational capability of hardware is quickly overwhelmed. The Hypothesis Management Module of the PROGNOS[1] Maritime Domain Awareness (MDA) fusion system manages the creation, modification, administration, storage and movement of hypotheses to ensure that only attributes and units relative to the current context are presented for inferential reasoning. Additionally, the Hypothesis Management Module supports recognition of observation trends leading to most likely hypotheses and the discovery of unpredicted hypotheses to provide asymmetric possibilities that match the incoming data.

The Hypothesis Management Module is under development in two phases to support the PROGNOS project. Phase I is the creation of the Hypothesis Management Engine, an essential aspect of the successful operation of the system. It provides the management and administration functions necessary to bind the hypotheses used for inferential reasoning, reducing computational overhead. Phase II is the development and integration of the Hypothesis Discovery Engine which provides innovative System Operator decision support in the form of hypothesis trends and original hypothesis recognition. The two component engines of the Hypothesis Management Module operate independently, allowing success of the PROGNOS project before completion of Phase II.

1.1 Hypotheses in Maritime Domain Awareness

Webster defines a hypothesis as 1) an interpretation of a practical situation or condition taken as the ground for action, or 2) a tentative assumption made in order to draw out and test its logic or empirical consequences. We will use a definition combining aspects of both of the above. For the maritime domain awareness situation assessment problem, a hypothesis can be thought of as a statement of anticipated action. In this context, a hypothesis can be thought of as a specifically

[1] The PROGNOS Project is funded by the Office of Naval Research.

defined plan of execution in which an actor will conduct an action against a target with a location, time and methodology of his choosing. Incoming data arrives from the PROGNOS Knowledge Exchange Module and is captured in the Hypothesis Management Module as an *m-tuple* of attributes in the hypothesis framework described below. A domain-specific inquiry is posed to the PROGNOS system by the System Operator through a hypothesis query which is also captured as an *m-tuple* and compared with the stored metadata. Components of the entire PROGNOS system are delineated in section 2. The hypothesis framework and query hypothesis are described below and their related activities are detailed in section 3.

Hypothesis Framework

Metadata from organic and non-organic information sources arrives in the Hypothesis Management Module via the PROGNOS Knowledge Exchange Module where it is continuously captured and stored in the hypothesis framework described in the following paragraphs and specified in section 3.1. Formally illustrated, each hypothesis, k, will have m attributes associated, $a_1 \ldots a_m$, as shown in Equation (1). The first $m\text{-}1$ attributes represent scenario options which are relevant to the current environment. The m^{th} attribute is reserved for the operational context and will delineate the relevance of the hypothesis to this and future contexts. Each of these attributes is stored in the vector $Hypothesis_k$.

$$\boldsymbol{Hypothesis_k} = \begin{bmatrix} \boldsymbol{a_1} \\ \boldsymbol{a_2} \\ \vdots \\ \vdots \\ \boldsymbol{a_m} \end{bmatrix} \qquad (\mathbf{1})$$

The initial implementation of PROGNOS, incorporating the Hypothesis Management Module Phase I, will be demonstrated in the maritime domain. An example of a 7-tuple attribute field domain and a non-inclusive representation of their possibilities for a North Atlantic smuggling maritime awareness scenario are captured in Table 1. Each attribute category has a number of possible answers and an abbreviated identifier.

Organization	Target	Delivery	Method	Location	Time	Context
AQ Al Quaida	CA Canada	C_Cruise Ship	AD Amphibious Drop	NE Northeast	2W 2 Weeks	A Air
ID Islamic Dawn	MX Mexico	M_Merchant	SB Small Boat Transfer	MA Mid-Atlantic	4W 4 Weeks	L Land
TT Tamil Tigers	PR Puerto Rico	W_Warship	CP Container in Port	SE Southeast	6W 6 Weeks	M Maritime
	US United States			GC Gulf Coast	8W 8 Weeks	S Space
					FU Future	

Table 1 - North Atlantic Smuggling Hypothesis Domain

Similarly, each hypothesis has an associated $m \times 1$ weight vector, Equation (2). The first $m\text{-}1$ rows assign a credibility figure to each of the $m\text{-}1$ attribute categories represented by the fields of the hypothesis. The m_{th} row is an indicator of the relevance of the associated context to the current operational environment as delineated by the System Operator during system configuration. This weight vector captures the credibility and relevance of the hypothesis in the current contextual domain.

$$\boldsymbol{Weight_k} = \begin{bmatrix} \boldsymbol{c_1} \\ \ddots \\ \boldsymbol{c_{m-1}} \\ \boldsymbol{r} \end{bmatrix} \qquad (\mathbf{2})$$

Credibility, represented by the first *m-1* rows in the weight matrix, is an indicator of the trustworthiness of the incoming data and its reporting source for attribute information. For example, reports generated from friendly units will likely be assigned higher credibility weights than those of informants.

Relevance, represented by the m_{th} row in the weight matrix, is an indicator of the significance of the associated hypothesis to the operational environment assigned by the System Operator during startup configuration. The Federal Rules of Evidence, Rule FRE-401 use the following definition:

> …..evidence having any tendency to make the existence of any fact that is of consequence to the determination of the action more probable or less probable than it would be without the evidence [7].

For this maritime domain awareness problem, we substitute evidence for information, and we have a working definition for relevance that identifies information of consequence to establishing the probability of an action.

This framework of hypothesis vector and its associated weight vector will be instantiated as many times as necessary to convey each possible hypothesis representing the incoming metadata. These two storage vehicles capture the story and strength of each hypothesis. The hypothesis *m-tuple* describes a specific instantiation of a possible scenario, and the weight vector allows us to update the scenario with incoming data and compare it to others in response to a query.

Query Hypothesis

The hypothesis framework described above is the structure used to capture and catalogue metadata available from organic and non-organic collection systems. The System Operator generates a query hypothesis to employ the PROGNOS system as an inferential reasoner to answer specific inquiries about the operational environment. This query hypothesis assumes the same structure as the hypothesis framework shown in Equation (1), above. PROGNOS calls on the Hypothesis Management Module to manage the creation, modification, administration, storage and movement of candidate hypotheses to ensure that only attributes and units relative to the current context are presented for inferential reasoning and to maintain computational viability. In a System Operator query, it is not necessary that every attribute field be assigned a query value. In fact, to open the aperture of candidates, the query hypothesis may be left rather sparse, allowing many hypotheses to match its attributes.

Associated with the query hypothesis is an *mx1* priority vector, Equation (3). The priority vector provides the System Operators prioritization of attributes and aids in the development of candidate hypotheses during the retrieval function described in Section 3.2, below.

$$\boldsymbol{Priority_i} = \begin{bmatrix} \boldsymbol{p_1} \\ \ddots \\ \ddots \\ \boldsymbol{p_m} \end{bmatrix} \quad (3)$$

The *Retrieve Hypothesis* activity uses the query hypothesis and the priority vector to retrieve and prioritize hypotheses stored in the Hypothesis Knowledge Base as outlined in section 3.2.

1.2 A Simple Example

For the remainder of this paper we will reference a simple scenario set in the Mediterranean Sea and North Atlantic Ocean. Agents of the terrorist organization Islamic Dawn operating out of Izmir, Turkey plan to smuggle radiological material into the United States on a bulk cargo vessel to build radiological dispersal devices. They intend to move the material ashore from the motor vessel *Mustafa Kamal* by offloading to commercial fishing craft off the Grand Banks and Cape Hatteras. This hypothesis, highlighted in Table 2 and summarized in Equation (4), represents the actual plan of action and is known only to the terrorists planning the operation.

Organization		Target		Delivery	Method		Location		Time		Context	
AQ	Al Quaida	CA	Canada	C_Cruise Ship	AD	Amphibious Drop	NE	Northeast	2W	2 Weeks	A	Air
ID	Islamic Dawn	MX	Mexico	M_Merchant	SB	Small Boat Transfer	MA	Mid-Atlantic	4W	4 Weeks	L	Land
TT	Tamil Tigers	PR	Puerto Rico	W_Warship	CP	Container in Port	SE	Southeast	6W	6 Weeks	M	Maritime
		US	United States				GC	Gulf Coast	8W	8 Weeks	S	Space
									FU	Future		

Table 2 - North Atlantic Smuggling Example Hypothesis

A hypothesis is created and stored by the Hypothesis Management Module for each ship entered into the PROGNOS system, with the unit name and type represented in the characteristic field of the Delivery attribute. It is possible that a single ship has multiple hypotheses associated with its name as conflicting information arrives which cannot be immediately clarified. For example, if the captain of the *Mustafa Kamal* is affiliated with Islamic Dawn, but a crewmember is affiliated with Tamil Tigers, separate hypotheses must be generated as the ship can be connected to both groups. A hypothesis will be created by the Hypothesis Management Module from incoming data, or it will reside in the Hypothesis Storage Module from a previous episode in a similar environment. This process will be further discussed in section 3 below.

Equation (4) tells the story of the hypothesis as planned by the terrorists using seven variables.

$$\boldsymbol{Hypothesis_{True}} = \begin{bmatrix} \textbf{Islamic Dawn} & \textbf{... is smuggling to the} \\ \textbf{United States} & \textbf{... on the} \\ \textbf{M\_Mustafa Kamal} & \textbf{... using} \\ \textbf{Small boat transfer} & \textbf{... near the} \\ \textbf{Northeast} & \textbf{... in the next} \\ \textbf{6 weeks} & \textbf{... in the} \\ \textbf{Martime Domain} & \end{bmatrix} \quad (4)$$

The weight vector, Equation (5), represents the credibility and relevance of each of the six attributes in the hypothesis and the context in which it occurs. Upon creation for each new surface vessel, these variables will be initialized using weights based on the content and source of the incoming data report. As more data arrive to the Hypothesis Management Module, one or more attribute weight values may be updated, strengthening or weakening the credence of the attribute. Equation (5) is the weight vector of the true hypothesis, under the assumption that (4) is known to represent the actual plan of the terrorists. Obviously in this case every attribute is credible and the context is relevant.

$$\boldsymbol{Weight}_{\boldsymbol{True}} = \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix} \qquad (5)$$

Unfortunately, this level of assurance is available only to the terrorist perpetrators and not to the System Operator. Using the scenario example, perhaps the System Operator is unsure of the vessel name, but has intelligence suggesting Islamic Dawn will smuggle radioactive material into the United States via merchant ship in the next 6 weeks and wants to identify ships that match this profile. The appropriate query hypothesis can be found in Equation (6), where M* represents a merchant ship of any name.

$$\boldsymbol{Hypothesis}_{\boldsymbol{Query}} = \begin{bmatrix} \boldsymbol{Islamic\ Dawn} \\ \boldsymbol{United\ States} \\ \boldsymbol{M}\_* \\ - \\ - \\ \boldsymbol{6\ weeks} \\ \boldsymbol{Martime\ Domain} \end{bmatrix} \qquad (6)$$

Further, the System Operator is positive that Islamic Dawn is behind the threat, and believes strongly that the United States is the target. He is fairly confident that the material will be smuggled by sea in the next six weeks. The associated priority vector is given by Equation (7).

$$\boldsymbol{Priority}_{\boldsymbol{Query}} = \begin{bmatrix} \mathbf{1.0} \\ \mathbf{0.9} \\ — \\ \mathbf{0.75} \\ \mathbf{0.75} \end{bmatrix} \qquad (7)$$

This framework of query hypothesis and priority vector captures the query of the System Operator and identifies which information has the highest priority. The *Retrieve Hypothesis* activity uses this information to retrieve and rank candidate hypothesis for use in the Reasoning Module of PROGNOS.

1.3 Overview of the Paper

In section 1 we introduced hypothesis management and explain why the Hypothesis Management Module is a critical technology for the Probabilistic Ontologies for Net-centric Operations System (PROGNOS) project. Section 2 summarizes the components of the PROGNOS domain, with emphasis on system architecture as it relates to the Hypothesis Management Module and its two components, the Hypothesis Management Engine and Hypothesis Discovery Engine. Section 3 narrows the focus to the PROGNOS Hypothesis Management Engine, which performs the core functions that facilitate coherent inferential reasoning. Similarly, section 4 provides detail on the PROGNOS Hypothesis Discovery Engine, which introduces hypotheses generated on the basis of observed attribute data and stored histories. Finally, building upon the components and respective technologies previously presented, section 5 describes the "big picture" by exploring how the Hypothesis Management Engine interacts with the rest of the system. Section 6 wraps up the paper with conclusions.

The Systems Modeling Language (SysML) is the language chosen to represent the Hypothesis Management Module and its activities for this project. The representation of hardware, software and operator entity interaction in a model-based engineering process is one of the strengths of SysML recognized by the Object Management Group (OMG) and key to the success of this project[5]. PROGNOS involves all of these entity types, as well as semantics, and is most accurately represented by the unique capability of SysML to integrate these systems engineering and mathematical disciplines.

2 The PROGNOS Project[2]

PROGNOS (PRobabilistic OntoloGies for Net-centric Operation Systems) is a proof of concept system under development by George Mason University under contract to the Office of Naval Research. The goal of PROGNOS is "to provide consistent high-level fusion of data through knowledge representation and reasoning and enable predictive analysis with principled hypothesis management [3]." The PROGNOS domain is depicted in Figure 1 and includes the following components:

- *System Operator* – The system operator is a human who interacts with a PROGNOS graphic user interface (GUI) to perform one of two functions:
  - Provide a data report. This is an analyst in a PROGNOS-equipped unit who provides input to the system in a standard format via GUI.
  - Query PROGNOS. This is an operator in a PROGNOS-equipped unit who queries the system to establish the hypothesis that most closely matches the current environment, context, and tactical situation.

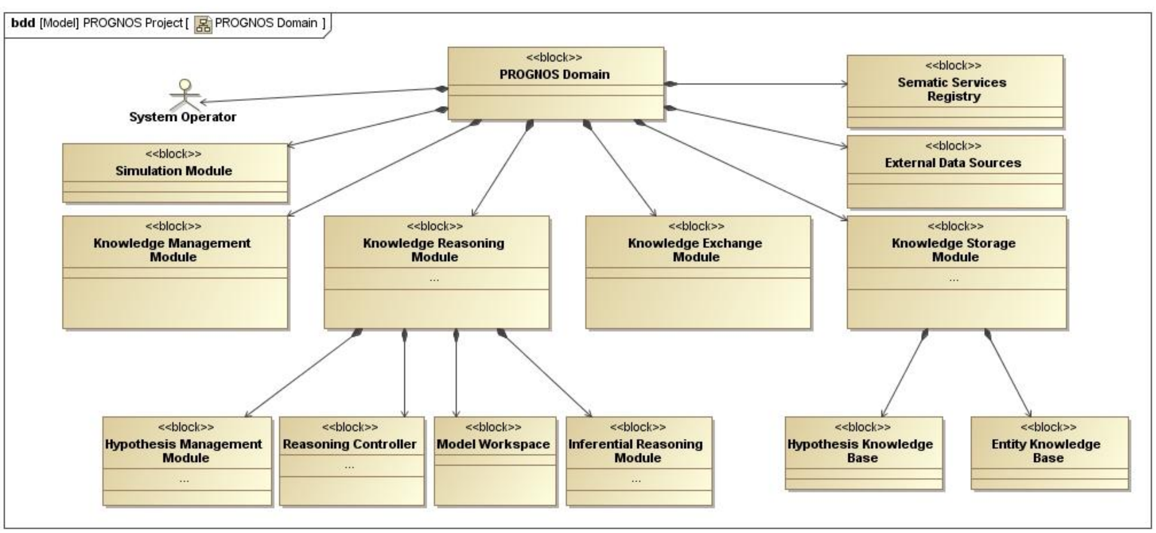

Figure 1– PROGNOS Domain

[2] The description of PROGNOS in this section is summarized from [2,3,4].

- *Simulation Module* – The Simulation Module provides randomly generated units (vessels) to the system with realistic behaviors using intelligent agents. This module is connected to PROGNOS via the Knowledge Exchange Module and provides simulated tracks to support analysis, maintenance and training evolutions.

- *Knowledge Management Module* – The Knowledge Management Module contains Task-specific and Task-neutral probabilistic Ontology Libraries which allow interoperability through shared, machine-interpretable semantics and support principled representation of uncertainty using the mathematical foundation of Multi-Entity Bayesian Networks (MEBN).

- *Knowledge Reasoning Module* – The Knowledge Reasoning Module coordinates the construction and inferential reasoning of a Situation-Specific Bayesian Network (SSBN) by the MEBN Reasoner in response to a query posed by a System Operator. Supporting the SSBN construction and reasoning process are the Reasoning Controller and Hypothesis Management Module which coordinate processes and create, modify, filter and prune candidate hypotheses, respectively.

- *Knowledge Exchange Module* – The Knowledge Exchange Module is the system's interface with the outside world. It receives the various formats of incoming information and ensures that they are PROGNOS-readable and ready for processing. The Knowledge Exchange Module is the conduit for data transmitted between PROGNOS and the Simulation Module, ship sensors, and ForceNet peers connected via the Semantic Services Registry.

- *Knowledge Storage Module* – The Knowledge Storage Module is comprised of the Hypothesis Knowledge Base and the Entity Knowledge Base. The former stores each hypothesis created from incoming data until it is sent to the Model Workspace in response to a System Operator query. Archived hypotheses are also maintained in the Knowledge Storage Module. The Entity Knowledge Base stores entities reasoned about and maintains dynamic links to the Main Probabilistic Ontology of the Task-neutral Probabilistic Ontology Library in the Knowledge Management Module.

- *External Data Sources* – External Data Sources are the myriad producers of data available to update the hypotheses of PROGNOS. Sensors aboard the host unit and those connected by tactical command and control systems arrive directly via the Knowledge Exchange Module. Data sources from other ForceNet peers ride on the Semantic Services Registry which has a conduit through the Knowledge Exchange Module. All data source types must be PROGNOS-readable for processing by the Hypothesis Management Module.

- *Semantic Services Registry* – The Semantic Services Registry is a loosely coupled association between service consumers and providers and rides on a net-centric service oriented architecture (FORCEnet). It allows a push-pull relationship of data exchange between PROGNOS-enabled and Non-PROGNOS units.

Each of these major components is under development separately and will be coalesced into a complete system for testing and analysis.

2.1 The Knowledge Reasoning Module

The Knowledge Reasoning Module has been called the heart of PROGNOS [3], as it performs all of the system's reasoning services in response to System Operator queries. Its major components are illustrated in Figure 2 and described below.

- *Hypothesis Management Module* – The Hypothesis Management Module manages the creation, modification, administration and movement of hypotheses between the Knowledge Storage Module and the Model Workspace in response to tasks assigned by the Reasoning Controller. It also predicts behavior and identifies original hypotheses based on observation of incoming data.

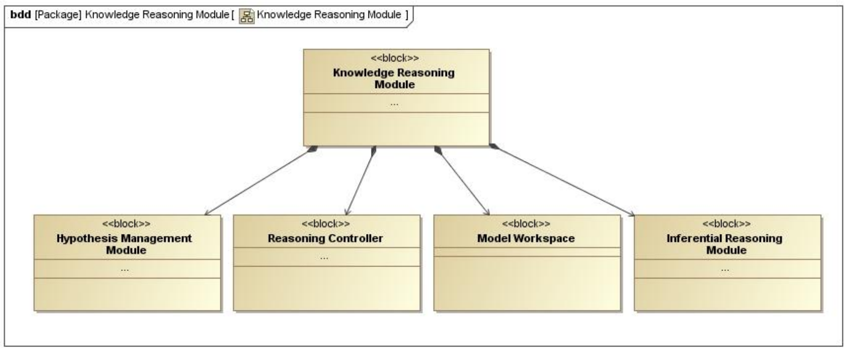

Figure 2 – Knowledge Reasoning Module

- *Reasoning Controller* – The Reasoning Controller manages the flow of data into the Hypothesis Management Module and the flow of hypotheses between the Hypothesis Management Module and Model Workspace in response to queries by the System Operator. It is the conduit of information and controller of all action within the Knowledge Reasoning Module.

- *Model Workspace* – The Model Workspace is the workbench on which the Situation-Specific Bayesian Network is assembled for use by the MEBN Reasoner.

- *Inferential Reasoning Module* – The Inferential Reasoning Module contains the Inference Engine which conducts the continuous cycle of inferential reasoning on the generated SSBN to generate a query response.

These components coordinate to create, administrate and nominate candidate hypotheses for inferential reasoning in the Situation Specific Bayesian Network in response to an operator query. The remainder of this paper presents the Hypothesis Management Module's two components, the Hypothesis Management Engine and Hypothesis Discovery Engine.

3 Hypothesis Management Engine

The Hypothesis Management Engine of the Hypothesis Management Module performs the essential functions of creating, updating, administrating, filtering and routing hypotheses as sub-activities within the major processes of *Process Incoming Data*, *Retrieve Hypotheses*, and

Archive Hypotheses. It coordinates closely with the Hypothesis Knowledge Base of the Knowledge Storage Module for retrieval and storage of hypotheses, both working and archived. The end result is a set of contextually relevant hypotheses built from streaming data that are filtered and pruned for computational efficiency and delivered to the Model Workspace in response to Reasoning Controller demand as a result of an operator query.

3.1 Process Incoming Data Activity

The Hypothesis Management Engine continuously creates and updates hypotheses from incoming data, as illustrated in the *Process Incoming Data* Activity Diagram, Figure 3. Before system activation, the System Operator selects configuration controls at the GUI which provide input relative to the current geopolitical state and status of the PROGNOS Unit. These data are used in the initialization of the Hypothesis Management Engine and will assist in the creation, update and filtering process. Upon startup, the *Process Incoming Data* activity operations within the shaded interruptible region execute continually on incoming streaming data until a shutdown control is received from the System Operator.

A data token, formatted for PROGNOS consumption by the Knowledge Exchange Module, arrives at the Hypothesis Management Engine via the Reasoning Controller. The frequency of arrival for data tokens is such that this process can be modeled as streaming events. An arriving data token initially travels along three parallel paths from the first fork, *Convert Bool to Control*, *Same Ship* decision node, and *Route to Discovery*.

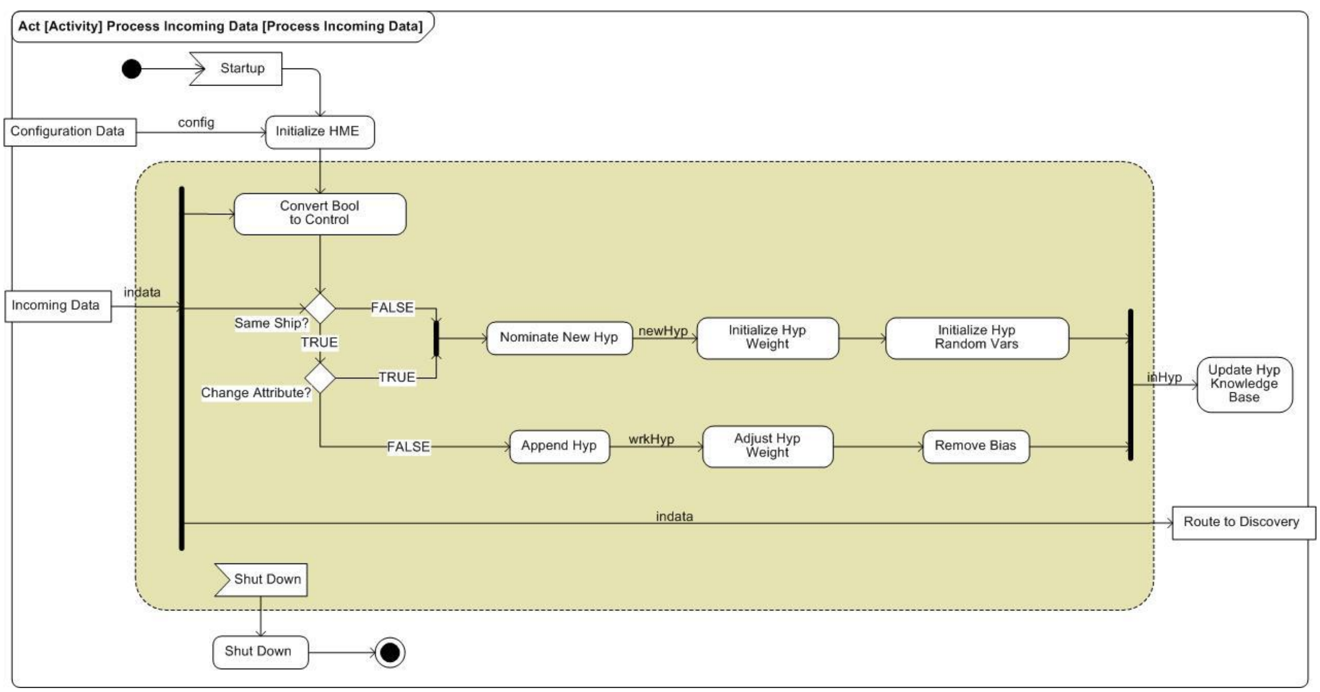

Figure 3 – Process Incoming Data Activity Diagram

In the *Convert Bool to Control* sub-activity, the data token is compared with existing hypotheses in the Hypothesis Knowledge Base to determine if the token will update the hypothesis or its weight matrix. A preliminary course filtering ensures the hypotheses compared are relevant to the current context. The lower bound on comparisons will be the number of ships in the PROGNOS system, if each had only one associated hypothesis. More likely there will be multiple hypotheses for some of the ships being monitored. If an incoming report describes a ship not tracked by PROGNOS, the *Convert Bool to Control* returns a FALSE value for the hypothesis and begins the process to *Nominate New Hyp*. If the new data describes a ship already in the system, an additional check is performed to determine if the information updates an existing hypothesis, or introduces a new one. In the case of the former, the *Append Hyp* path is traveled and in the latter the situation is treated as a completely new hypothesis that must be created using the *Nominate New Hyp*. This control check determines which path, *Append Hyp* or *Nominate New Hyp*, the data travels from the *Append Decision* node.

For each hypothesis flagged for update by the *Change Attribute* decision node, the data token travels the middle sequence of sub-activities in Figure 3. In the first, *Append Hyp*, the hypothesis flagged in the *Convert Bool to Control* sub-activity is called and the new data token is added as an additional attribute of the hypothesis. The output of the *Append Hyp* sub-activity is an appended hypothesis, `wrkHyp`.

The next sub-activity, *Adjust Hyp Weight,* adjusts the credibility for the updated attribute in the weight matrix for the working hypothesis based on the data and its source. It is possible that a hypothesis has additional data associated with it as a result of the *Append Hyp* sub-activity, but because of the source the hypothesis force is lowered, making it less likely in the current situation. On the other hand, similar data arriving from alternate sources may serve to increase the hypothesis weight for one or more attributes.

The final sub-activity in the *Append Hyp* track is *Remove Bias*. There is natural bias associated with the value units associate with data provided from their own sensors. This sub-activity attempts to correct these and other identified sources of bias in the working hypothesis. These three steps are performed on each flagged hypothesis and its corresponding weight matrix.

A data token identified as not relating to any existing hypotheses in the Hypothesis Knowledge Base or altering attributes for a unit in an existing hypothesis travels the upper sequence of sub-activities in Figure 3 in which new hypotheses are nominated and initialized. In the *Nominate New Hyp* sub-activity, a working hypothesis m-tuple is created. The output of the *Nominate New Hyp* sub-activity is a new working hypothesis, `newHyp`.

The next sub-activity, *Initialize Hyp Weight*, evaluates the data source, the System Operator initialization settings, and the current context to produce an initial weight vector for the new hypothesis. The *Initialize Hyp Weight* sub-activity uses much of the same information as the *Remove Bias* sub-activity above and may share a common subroutine.

The final sub-activity in the *Nominate New Hyp* track is to *Initialize Hyp Random Var*. This activity is essential to ensure the new hypothesis has pristine data fields which may be updated as additional data arrives.

Finally, the updated or created hypothesis from either of the above activity tracks is delivered to the Hypothesis Knowledge Base by the *Update Hyp Knowledge Base* sub-activity where it awaits incoming related data for further update, or a call as a candidate for reasoning in the Inference Engine.

On the lowest parallel sequence of the activity, incoming data is passed directly to the Hypothesis Discovery Engine for additional processing. This is a continual function that will facilitate asymmetric hypothesis creation, discussed in section 4.

3.2 Retrieve Hypothesis Activity

In response to a System Operator query, the Reasoning Controller requests candidate hypotheses from the Hypothesis Management Module for use in the creation of the System-Specific Bayesian Network in the Model Workspace. The *Retrieve Hypothesis* activity of the Hypothesis Management Engine coordinates with the Hypothesis Knowledge Base for retrieval, filters and prunes the hypotheses within the context of the query, and forwards the filtered hypotheses to the Model Workspace through the Reasoning Controller, as illustrated in Figure 4 and described below.

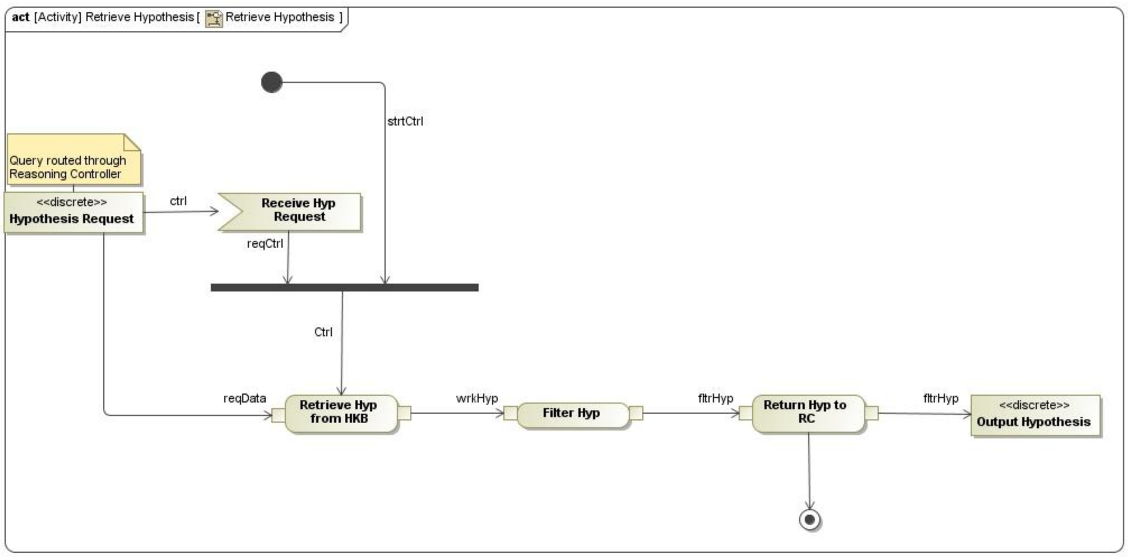

Figure 4 - Retrieve Hypothesis Activity Diagram

The Reasoning Controller generates a request for candidate hypotheses based on a System Operator-generated query hypothesis. This request acts as the control token to begin the *Retrieve Hypothesis* activity. Additional data included in the request is forwarded to the first sub-activity and is used to retrieve the appropriate hypotheses from the Hypothesis Knowledge Base.

The first sub-activity, *Retrieve Hyp from HKB*, uses query hypothesis data from the request to iteratively search for and retrieve one or more hypotheses from the Hypothesis Knowledge Base. This query hypothesis data includes the attributes that represent positive or negative information about the query and the entity of interest. By matching attribute fields with those in stored hypotheses, candidates can be identified that meet a threshold of associated common characteristics. These candidate hypotheses are prioritized by comparison with the priority vector provided by the System Operator. This sub-activity returns one or more working hypotheses, `wrkHyp[i]`.

Arguably the most important activity in the Hypothesis Management Engine is the *Filter Hyp* sub-activity. Even simple Bayesian networks become exponentially large with relatively few nodes. The filtering and pruning function performed in this sub-activity prevents the Situation

Specific Bayesian Network from becoming too large for the computational power of the PROGNOS hardware. The *Filter Hyp* sub-activity performs two serial functions on each `wrkHyp[i]` to produce manageable products, *Filtering* and *Pruning*. Filtering is the process used to weed out data that is not associated with the present query, and therefore not relevant to the Inference Engine. Pruning performs a similar function, but trims attribute fields that do not fit the current context or environment in which the query is being performed.

Using our example, the Mufasa Kamal is traveling West across the Atlantic toward the North American Continent. The System Operator initiates a PROGNOS query to identify the most likely candidate tracks that could be the ship smuggling dangerous cargo from Turkey to the United States. In the *Filter Hyp* sub-activity, the filtering function removes attributes from candidate hypotheses that do not relate to maritime operations, smuggling, or terrorism. Similarly, the pruning function trims attribute fields that are identified with the query hypothesis or do not represent a westbound track.

The output of the *Filter Hyp* sub-activity is a set of filtered hypotheses, fltrHyp[i], which are returned to the Reasoning Controller for transmission to the Model Workspace and use by the Inference Engine. This discrete series of actions is performed at the initiation of each new query and iterated at some fixed time interval to allow updates to the Model Workspace as additional data arrives in PROGNOS.

3.3 Archive Hypothesis Activity

Units often depart operating areas due to a change of mission only to find themselves back in the same area at a later date. Relational data between entities is not likely to change in the short term and should be maintained to expedite unit situational awareness upon return. The *Archive Hypothesis* activity shown in Figure 5 allows non-time sensitive attributes of hypotheses to be archived in the Hypothesis Knowledge Base in anticipation of building upon them, when required.

The *Archive Hypothesis* activity remains dormant until receiving a cue from the System Operator via the GUI that the unit is changing missions. The activity systematically evaluates each hypothesis stored in the Hypothesis Knowledge Base and removes from each all attribute fields associated with spatial and temporal data. For example, attributes of position, course, speed, environment, or time of day may all be deleted. On the other hand, family/social relationships, home port, vessel type and previous erratic behavior may be stored as background data for later use.

Upon System Operator indication that the mission will change, the *Hyp to Check in HKB* sub-activity first determines if there are any hypotheses resident in the Hypothesis Knowledge Base. Then, the system recursively retrieves each stored hypothesis, removes any of its spatio-temporal attributes, and saves it back into the Hypothesis Knowledge Base in the *Retrieve Hyp from HKB, Remove Spatio Temporal Attributes,* and *Save Hyp to HKB* sub-activities.

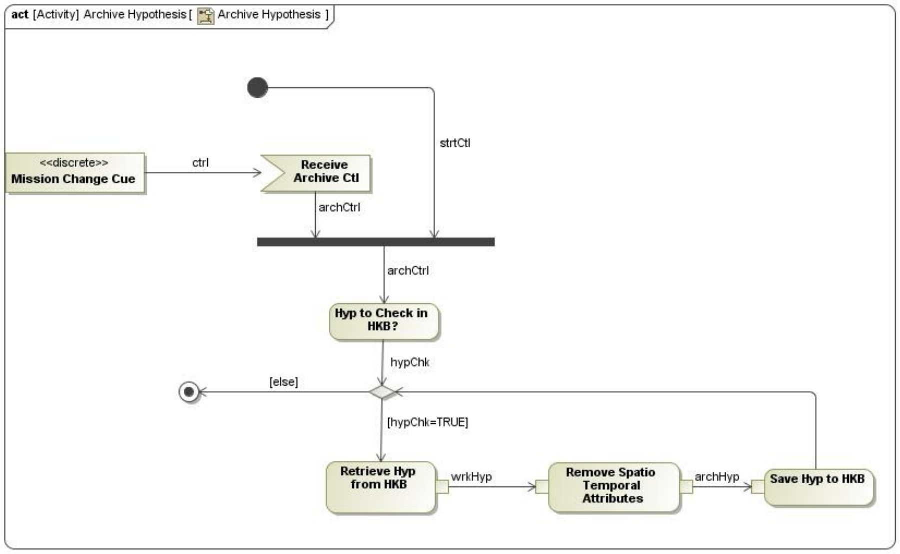

Figure 5 – Archive Hypothesis Activity Diagram

This activity results in a database of hypotheses consisting of useful long-term information about relationships between entities and devoid of any spatio-temporal data. Should the PROGNOS unit return to the same operational setting, these hypotheses are available to the Hypothesis Management Engine to build upon with incoming data.

4 Hypothesis Discovery Engine

The Hypothesis Discovery Engine of the Hypothesis Management Module produces original hypotheses from observed attribute data and recommends which queries the System Operator may desire to pose. The two parallel activities of the Hypothesis Discovery Engine, *Propose Hypothesis* and *Evolve Hypothesis* are shown in Figure 6 and discussed in detail below. These functions support the System Operator in wading through copious data and help to identify potential actions by asymmetric actors attempting to blend into background activities.

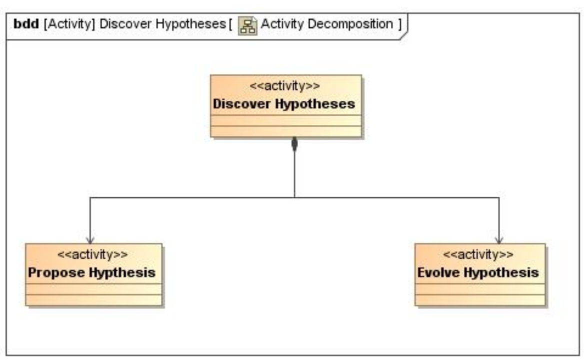

Figure 6 – Hypothesis Discovery Engine Activity Decomposition

Initially, PROGNOS will be introduced with only Hypothesis Management functionality as research continues on the unique functions of the Hypothesis Discovery Engine. By design, the Hypothesis Management Engine operates independently of the Hypothesis Discovery Engine to allow modular insertion of this advanced technology.

4.1 Propose Hypothesis Activity

The *Propose Hypothesis* activity, illustrated in Figures 7 and decomposed in Figure 8, collects statistical information on incoming data and bins it into hypothesis areas. At any given point in time, one bin of associated hypotheses emerges as containing the most likely scenario given the observed data entering the system, weighed appropriately by its relevance, credibility and force. With some regular periodicity this information is delivered to the System Operator in the form of a prioritized list of likely events and associated queries that may be initiated to substantiate a specific threat. The System Operator can use this function as a cueing tool to alert on building evidence.

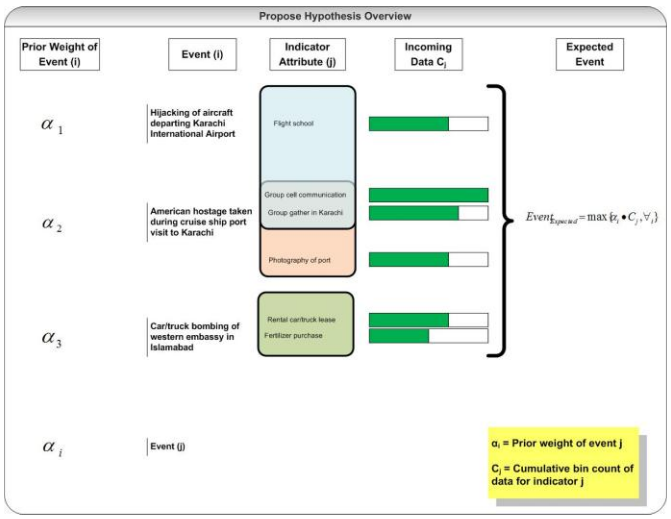

Figure 7 – Propose Hypothesis Overview

The activity diagram illustrated in Figure 8 deconstructs the sub-activities of the *Propose Hypothesis* activity. Streaming data enters the Hypothesis Discovery Engine where it is initially compared to a set of indicators pre-identified by regional subject matter experts as related to specific events of interest in the *Compare to Indicator (j)* sub-activity. If the data fits one or more of these indicators, appropriate counters are incremented by the *Increment Attribute Counter* sub-activity. There are also weights associated with each event that are stored in the Event Data datastore and are used in determining the most likely event based on a weighted "build up" of indicators by the *Determine Expected Event* sub-activity.

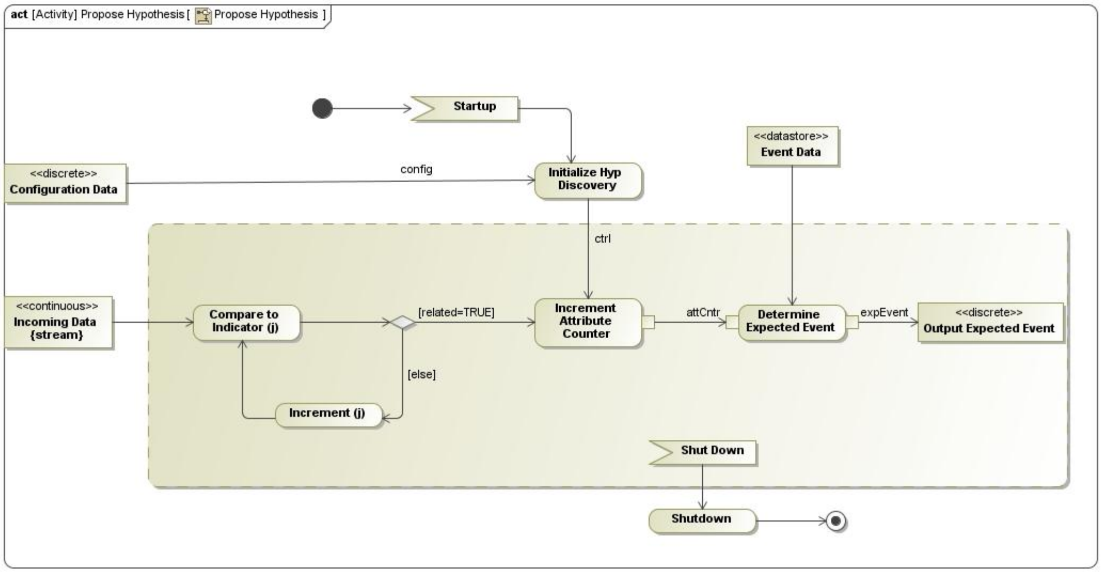

Figure 8 – Propose Hypothesis Activity Diagram

The most likely event is displayed and periodically updated by the GUI to prompt the System Operator to alert him about his environment. This allows the operator to choose the most relevant query to his current situation.

4.2 Evolve Hypothesis Activity

The *Evolve Hypothesis* activity is an original effort to create unforeseen hypotheses which may identify asymmetric actions endangering units. This is accomplished by transforming existing hypotheses resident in the Hypothesis Knowledge Base and checking for feasibility before making them available for update and use in the Inference Engine. Genetic mutation of hypotheses is the transformation planned for initial implementation of the activity.

The *Retrieve Hyps from HKB* sub-activity of the *Evolve Hypothesis* activity selects a number of hypotheses from the Hypothesis Knowledge Base for genetic alteration, as shown in Figure 9. A random number of attributes from these working hypotheses are randomly shifted to the others to produce genetically mutated hypotheses in the *Mutate Hyp* sub-activity which may represent courses of action previously unforeseen. Unfeasible hypotheses, e.g. a unit in two places simultaneously, are filtered by a feasibility check in the *Check Hyp Feasiblility* sub-activity which looks for spatio-temporal or relational errors. For example, a ship cannot be located overland and an individual cannot be both a family member and non-family member. Hypotheses that fail this test are discarded. Those that survive are added to the Hypothesis Knowledge Base by the *Send Hyps to HKB* sub-activity for later update by the *Process Incoming Data* activity or as candidates for use in System Operator queries in the *Retrieve Hypothesis* activity.

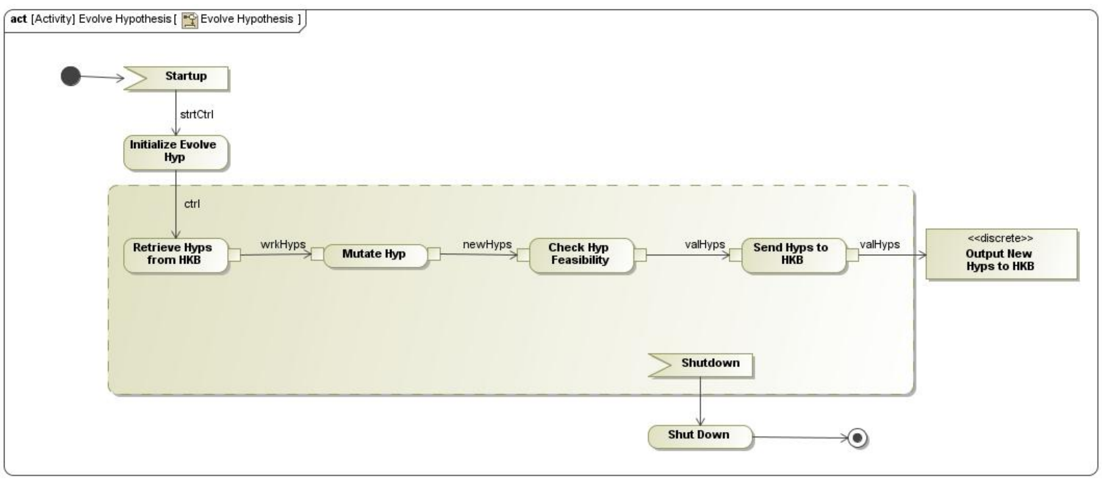

Figure 9 – Evolve Hypothesis Activity Diagram

As computational resources allow, the Hypothesis Discovery Engine iterates the processes in the interruptible region of the *Evolve Hypothesis* activity until PROGNOS is secured. Even with no incoming data, the Hypothesis Discovery Engine can mutate the hypotheses resident in the Hypothesis Knowledge Base as a tool for determining alternate courses of action and relationships.

The ultimate goal of the *Evolve Hypothesis* activity is to identify potential hypotheses not imagined at the time of system setup for the regional context. By constantly observing and transforming real-time and archived data, the Hypothesis Discovery Engine introduces asymmetric hypotheses into the candidate hypothesis set used to answer System Operator queries.

5 HMM Interaction with PROGNOS

The Hypothesis Management Module interacts with the rest of the PROGNOS system primarily through the System Operator-induced query process. While the *Process Incoming Data*, *Archive Hypotheses, Propose Hypothesis* and *Discover Hypothesis* activities receive continuous controls and data during system operation, it is the *Retrieve Hypothesis* activity that requires true interaction between all parts of the PROGNOS system, as shown in Figure 10 below.

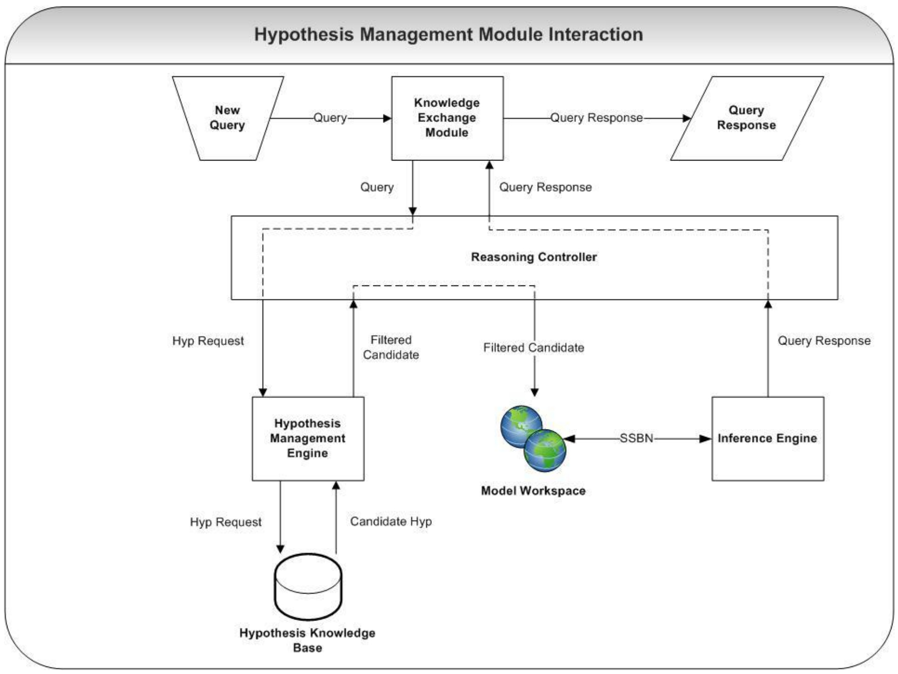

Figure 10 – HMM Interaction During PROGNOS Query

The simplified flow of a query outlined in Figure 10 begins with a *New Query* to PROGNOS initiated by the System Operator at a GUI. This flows through the Knowledge Exchange Module to the Reasoning Controller where it is converted to a *Hypothesis Request*. The request is sent to the Hypothesis Management Engine of the Hypothesis Management Module which coordinates with the Hypothesis Knowledge Base to select one or more *Candidate Hypotheses*.

Before departing the Hypothesis Management Engine, these *Candidate Hypotheses* are filtered and pruned to maintain computational viability before transfer to the Model Workspace via the Reasoning Controller. The Model Workspace and Inference Engine work to create the Situation Specific Bayesian Network and conduct the inferential reasoning that determines a *Response* to the query. The *Response* is returned to the System Operator through the Reasoning Controller and Knowledge Exchange Module.

Without introduction of a query by a System Operator, the Hypothesis Management Module continuously performs three major functions on incoming data. It processes incoming data, proposes hypotheses, and discovers hypotheses, as discussed in Sections 3 and 4, above.

For the North Atlantic smuggling example, Locher proposed the model shown in Figure 11 as a naïve Bayes classifier for reasoning about surface ships [6]. To make use of such a model in PROGNOS, a copy of this reasoning model would be generated for each surface ship in the system, updated with incoming data, and maintained in the Entity Knowledge Base. This is a simple example of the kind of model that the PROGNOS reasoning module will apply. More sophisticated models will consider relational information involving multiple vessels and/or actors. Incorporating such relational reasoning is much more computationally demanding, and increases the need for effective hypothesis management.

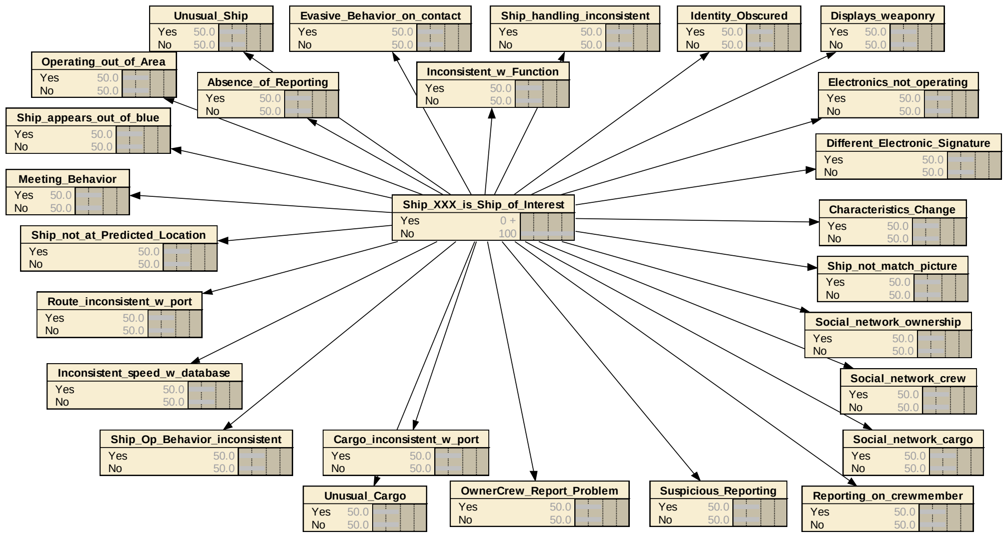

Figure 11 – Naïve Bayesian Network of Merchant Ship

As these reasoning rules become more complex and the contacts more numerous, they will quickly overwhelm the computational capability of the PROGNOS hardware. The Hypothesis Management Engine prunes the hypotheses returned to the reasoning module by eliminating information that is not relevant to the current query or context. One example of a mapping of hypothesis attributes to reasoning identifiers is shown in Figure 12.

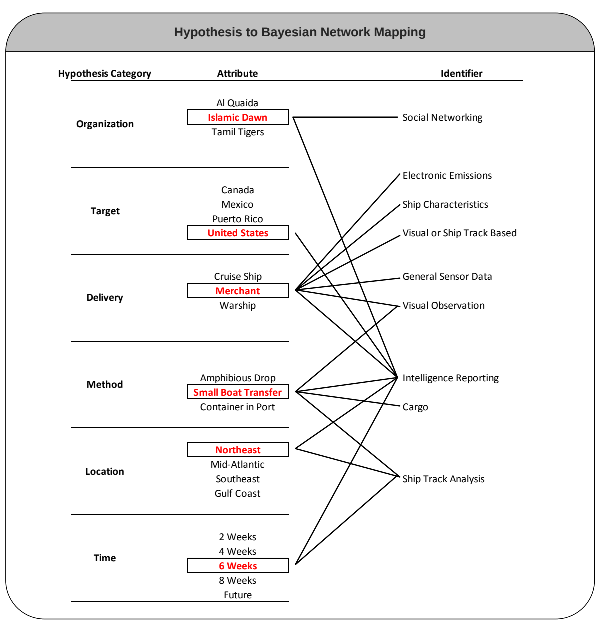

Figure 12– Hypothesis to Bayesian Network Mapping

This mapping indicates which reasoning rules are relevant based on the attributes identified in the hypothesis query. To illustrate this explicitly, assume the System Operator initiates a query with the attribute vector given by $Hypothesis_{Query1}$ in Equation 8:

$$\boldsymbol{Hypothesis_{Query1}} = \begin{bmatrix} \boldsymbol{Islamic\ Dawn} \\ \boldsymbol{United\ States} \\ \boldsymbol{M\_*} \\ \_ \\ \_ \\ \boldsymbol{6\ weeks} \\ \boldsymbol{Maritime} \end{bmatrix} \qquad (\mathbf{8})$$

This equation summarizes the situation in which Islamic Dawn is attempting to smuggle material into the United States on a merchant vessel in the next 6 weeks. The method of delivery from the ship to land and the location is not known. The system operator initiates a hypothesis query to locate the ship or ships that best match $Hypothesis_{Query1}$. In this case, the mapping indicates that only the *cargo* identifier can be eliminated from the returned hypotheses. As an alternative, assume that the smuggling timeline is also unknown, giving the $Hypothesis_{Query2}$ attribute vector, shown in the Equation 9:

$$\boldsymbol{Hypothesis_{Query2}} = \begin{bmatrix} \boldsymbol{Islamic\ Dawn} \\ \boldsymbol{United\ States} \\ \boldsymbol{M\_*} \\ \_ \\ \_ \\ \_ \end{bmatrix} \qquad (\mathbf{9})$$

Now the mapping indicates that the *cargo*, and *ship track analysis* attributes are not relevant. Therefore, the *Retrieve Hypothesis* activity would identify which hypotheses in the Hypothesis Knowledge Base best matched $Hypothesis_{Query2}$ and would deliver the pruned reasoning model to the Inference Engine. Figure 13 illustrates the pruned reasoning network associated with $Hypothesis_{Query2}$.

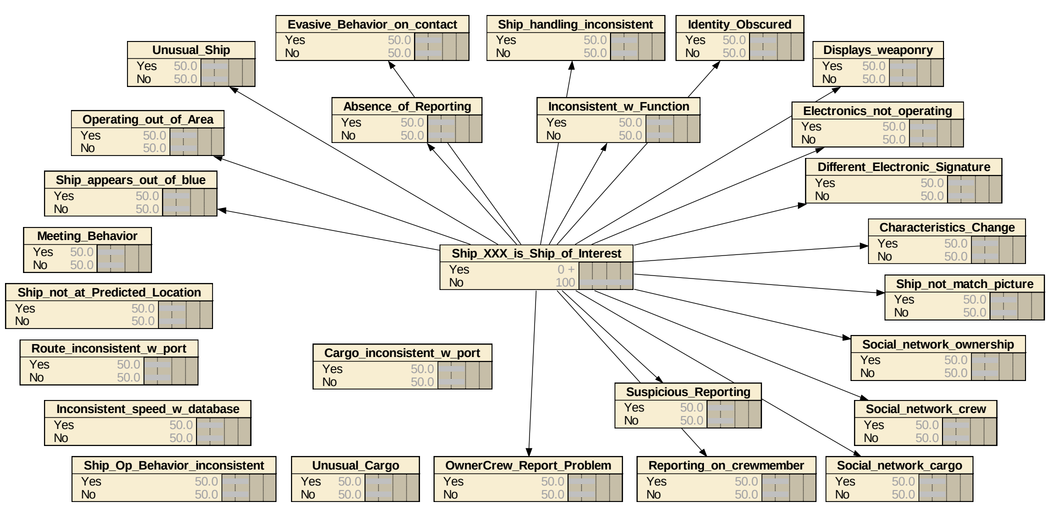

Figure 13 – Pruned Bayesian Network for Smuggling Example

In this case, since the System Operator is unsure of the method, location, or timeline of the smuggler's arrival, the seven reasoning rules associated with these unknowns are not included in the data supplied to the Inference Engine. With thousands of ships at sea every day, this pruning can result in significant savings in computational ability.

6 Conclusions

The Hypothesis Management Module of the PROGNOS system controls the exponential growth in candidate hypotheses delivered to the Reasoning Controller for inferential reasoning. By managing the creation, modification, administration, storage and movement of hypotheses and ensuring that only attributes and units relative to the current context are presented for inferential reasoning, it controls the number of hypotheses created as PROGNOS receives incoming observation data. The final system not only performs the critical functions of efficient creation, revision, movement, filtering, and archiving of hypotheses, but introduces features revolutionizing the situation awareness of tactical Systems Operators and allows focus on corrective action, vice data fusion.

The two-phase design and implementation approach to the Hypothesis Management Module ensures that PROGNOS is able to provide its primary inferential reasoning output before development of the hypothesis discovery functions. Phase I develops the Hypothesis Management Engine which provides the management and administration functions necessary to bind the hypotheses used for inferential reasoning, reducing computational overhead. Phase II develops and integrates the Hypothesis Discovery Engine which supports recognition of observation trends leading to most likely hypotheses and the discovery of unpredicted hypotheses to provide asymmetric possibilities that match the incoming data. Together, these components of the Hypothesis Management Module facilitate high-level data fusion by the PROGNOS maritime domain awareness system.

7 References

1. Alberts, D., Garstka, J., & Stein, F. (1999). Network Centric Warfare: Developing and Leveraging Information Superiority. *CCRP Publication Series* , 256.

2. Costa, P., Chang, K., Laskey, K., & Carvalho, R. (2009). A Multidisciplinary Approach to High Level Fusion in Predictive Situational Awareness. *Proceedings of the 11th International Conference of the Society of Information Fusion (FUSION 2009).* Seattle, WA, USA.

3. Costa, P., Laskey, K., & Chang, K. (2009). PROGNOS: Applying Probabilistic Ontologies to Distributed Predictive Situation Assessment in Naval Operations. *Proceedings of the 14th International Command and Control Research and Technology Conference* . Washington D.C., USA.

4. Costa, P., Laskey, K., Laskey, K., & Wright, E. (2007). Probabilistic Ontologies: the Next Step for Net-Centric Operations. *Proceedings of the 12th International Command and Control Research and Technology Symposium.* Newport, RI, USA.

5. Friedenthal, S., Moore, A., & Steiner, R. (2008). *A Practical Guide to SysML.* Boston, MA: The MK/OMG Press.

6. Locher, M. (2010). *Defining Reasoning Rules for a Naval Information Fusion System-PROGNOS.* George Mason University, Fairfax, VA.

7. Schum, D. (1994). *Evidential Foundations of Probabilistic Reasoning.* New York, NY: John Wiley & Sons, Inc.

Richard Haberlin is a retired US Naval Flight Officer with extensive experience in anti-submarine warfare and airborne intelligence, surveillance and reconnaissance operations in the Arctic, Atlantic, Mediterranean and Middle East. He is currently a senior analyst and military subject-matter expert for Enterprise Management Solutions Inc. in Alexandria, Virginia. An active member of the Institute for Operations Research and the Management Sciences, he holds an M.S. in Operations Research from the Naval Postgraduate School and a B.S. in Ocean Engineering from the United States Naval Academy.