3RD INTERNATIONAL CONFERENCE ON MODEL-BASED SYSTEMS ENGINEERING

A MODEL BASED SYSTEMS ENGINEERING APPROACH TO HYPOTHESIS MANAGEMENT

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 (904) – 742 – 7624

Abstract

Situational awareness supports tactical decision making through fusion of information about intelligence, geography, environment, and the geopolitical situation. Advanced decision support systems can help decision makers explore hypotheses about the evolving situation. Computational resources place limits on the number of hypotheses that can be considered. Hypothesis Management controls the exponential growth in fusion hypotheses created by a deluge of reports. This paper describes the Model Based Systems Engineering (MBSE) approach used to develop a Hypothesis Management Module as part of the Office of Naval Research Probabilistic Ontologies for net-Centric Operations Systems (PROGNOS) project. Specifically, a top-down design process was employed utilizing the Object Management Group Systems Modeling Language. After a brief introduction to the PROGNOS project, a systematic description of the MBSE methodology is provided.

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. PROGNOS (Probabilistic Ontologies for Net-Centric Operations Systems) is an Office of Naval Research sponsored high-level fusion prototype that includes hypothesis management as one of its enabling technologies to cope with model complexity [1]. The focus of this paper is on the use of Model-Based Systems Engineering (MBSE) to support the development of the PROGNOS Hypothesis Management Module (HMM) and its sub-components. The HMM architecture was modeled in the SysML language using MagicDraw 16.6.

The Hypothesis Management Module of the PROGNOS Maritime Domain Awareness (MDA) fusion system manages the creation, modification, administration, storage and movement of hypotheses to focus inference on hypotheses relevant to the current context. Additionally, the HMM supports recognition of patterns of evidence to be passed on to the inferential reasoning module, which evaluates the most probable hypotheses given current evidence. Furthermore, the HMM supports discovery of new, unanticipated hypotheses. The HMM is under development in two phases. Phase I is the creation of the Hypothesis Management Engine, an essential aspect of the successful operation of the overall 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, enabling the development of an operational PROGNOS hypothesis management capability before completion of the second phase.

For our system design, we employ the functional analysis approach based on the OMG Systems Modeling Language Tutorial. The approach consists of the following steps [3]:

1. Organize the model
2. Establish requirements
3. Model behavior
4. Model structure
5. Capture & evaluate parametric constraints
6. Modify the design to meet constraints
7. Model user interaction
8. Modify the design to reflect interaction

The PROGNOS HMM is a piece of software still in its design stage. Steps 1-4 of the above methodology are presented below with appropriate accompanying diagrams.

Hypotheses in Maritime Domain Awareness

For the maritime domain awareness situation assessment problem, a hypothesis delineates a statement of anticipated action. Therefore, a hypothesis summarizes a specifically defined plan of execution in which an adversary will conduct a specified action against a target with a location, time and methodology of his choosing. Incoming data, streaming and aperiodic, arrives from the PROGNOS Knowledge Exchange Module and is passed to the Hypothesis

Management Module, where it is captured and stored in a structured representation called the hypothesis framework. A domain-specific inquiry is posed to the PROGNOS system by the System Operator through a hypothesis query using a similar framework and compared with the stored metadata. The hypothesis framework and query hypothesis are briefly described below and detailed in [2].

Hypothesis Framework

Data 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 as a series of attributes representing features relevant to the current environment. Each of these attributes is stored in a vector $Hypothesis_k$.

Additionally, each hypothesis has an associated weight vector which assigns a credibility figure to each of the attribute categories represented by the fields of the hypothesis. This weight vector captures the credibility and relevance of the hypothesis in the current contextual domain [5]. Credibility is an indicator of the trustworthiness of the incoming data and its reporting source for attribute information. For example, reports generated from organic units will likely be assigned higher credibility weights than those of coalition units.

Relevance is an indicator of the significance of the associated hypothesis to the operational environment assigned by the System Operator during startup configuration. In the maritime domain awareness problem, substituting evidence for information gives a working definition of 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 hypothesis under consideration. These two knowledge structures capture the content and strength of each hypothesis. The hypothesis vector describes a specific instantiation of a possible scenario, and the weight vector allows us to update its credibility 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 data available from organic and non-organic collection systems. To realize the decision support available through the PROGNOS inference algorithm, the System Operator generates a query hypothesis to answer a specific inquiry about the operational environment, using the same framework structure. 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. Associated with the query hypothesis is a priority vector, which allows the System Operator to prioritize attributes, and aids in the development of candidate hypotheses during the retrieval function described below.

Overview of the Paper

In section 1 we introduced hypothesis management and explained why the Hypothesis Management Module is a critical technology for the PROGNOS project. Sections 2-4 detail the MBSE methodology sequence listed above for the design of the HMM and its two components, the Hypothesis Management Engine and Hypothesis Discovery Engine. Section 5 wraps up the paper with conclusions.

The Object Management Group Systems Modeling Language (OMG SysML) is the language chosen to represent the Hypothesis Management Module architecture and its relationship to the PROGNOS system. The available representations 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 [4]. This capability is key to the success of this project. 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.

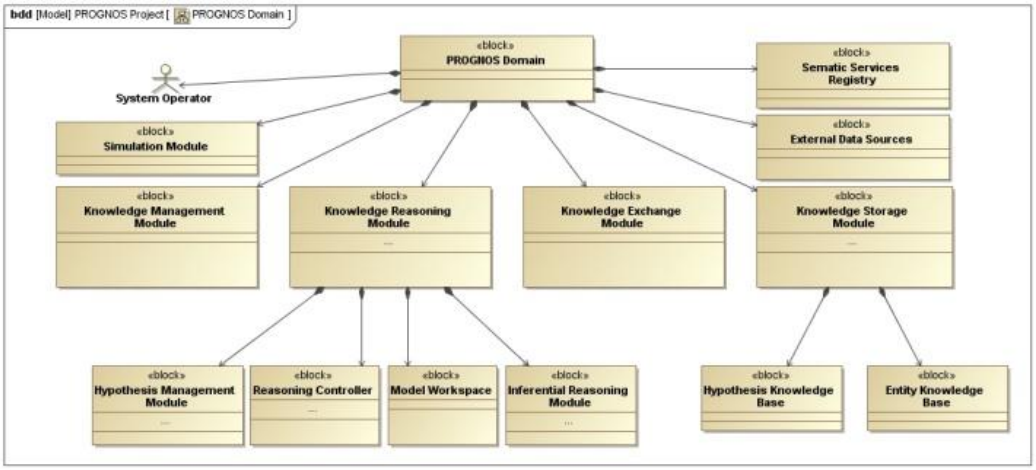

Figure 1 - PROGNOS Domain

2. Model Organization

PROGNOS 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 [1]."

To understand the relationship of the HMM to the entire project it is necessary to understand how the many components interact. Figure 1 depicts the overall PROGNOS domain in the form of a block definition diagram. Each of the major components is under development separately and will be coalesced into a complete system for testing and analysis.

The Knowledge Reasoning Module

One of the primary components of the system is the Knowledge Reasoning Module (KRM), decomposed below in Figure 2. The Knowledge Reasoning Module has been called the heart of PROGNOS [1], as it performs all of the system's reasoning services in response to System Operator queries. The Hypothesis Management Module is a subcomponent of the KRM.

These components coordinate to create, administrate and nominate candidate hypotheses for inferential reasoning in the Situation Specific Bayesian Network (SSBN) in response to an operator query. Further discussion about the components of PROGNOS can be found in [1,2]. The remainder of this paper presents the architecture of the HMM and its two subcomponents, the Hypothesis Management Engine and Hypothesis Discovery Engine.

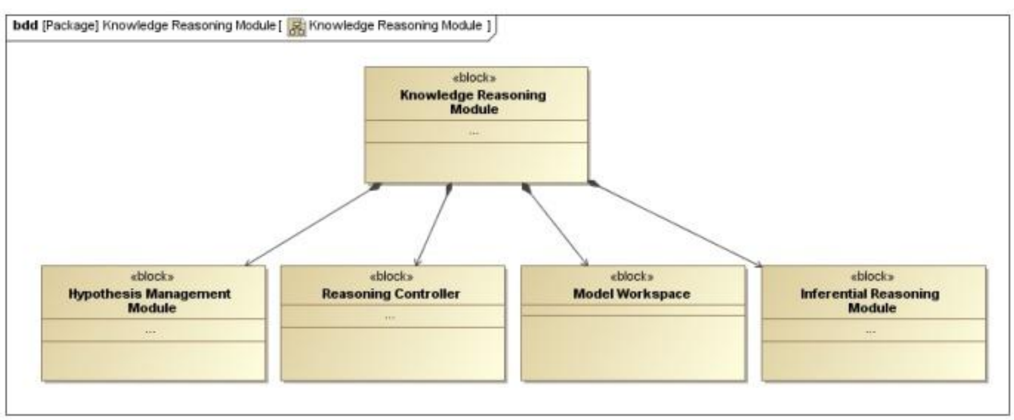

Figure 2 - Knowledge Reasoning Module Domain

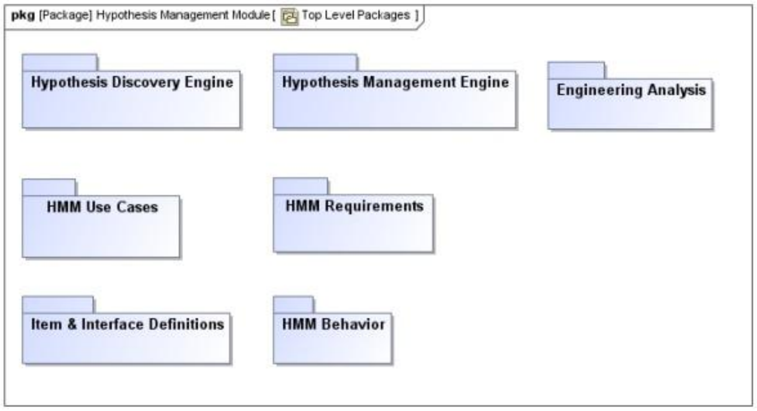

Figure 3 - HMM Top-Level Package Diagram

Hypothesis Management Module

The organization of the HMM begins with the top level packages identified in Figure 3. In addition to packages for its subcomponents, the Hypothesis Management Engine and Hypothesis Discovery Engine, this diagram illustrates top-level HMM Use Cases, Behavior, and overall System Requirements. The Engineering Analysis package includes system constraints and the parametric models used to evaluate system performance. The final package illustrated is the Item & Interface Definitions which documents reusable attributes and port definitions used within the model.

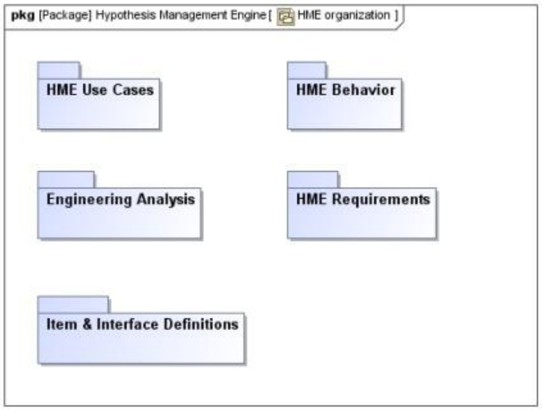

Figure 4 - HME Package Diagram

Decomposing the HMM into its subcomponents, the Hypothesis Management Engine (HME) and Hypothesis Discovery Engine (HDE) yields the package diagrams illustrated in Figure 4 and Figure 5. Organization of the HME includes HME Use Cases and Behavior packages, as well as Requirements specific to this component. Engineering Analysis and Item & Interface Definitions packages are again present with similar functionality to that discussed above for the Top-Level Package Diagram.

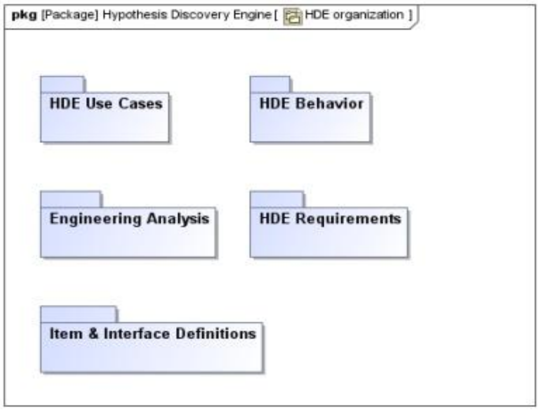

Figure 5 - HDE Package Diagram

Similarly, the HDE package diagram illustrates the components included in this subsystem. These include HDE Use Cases and Behavior packages, HDE Requirements, and the Engineering Analysis Package. The HDE package diagram also includes the Engineering Analysis and Item & Interface Definitions packages introduced in the HMM package diagram, above.

The universal modality of the Engineering Analysis and Item & Interface Definitions packages allows commonality of specifications, parameters and attributes across the entire HMM model. This harmony of effort will be critical during evaluation of the HMM and in modeling the flow of data in and out of ports.

Table 1 - HMM Requirements

#	ID	Name	Text
1	0	PerformHypothesisManagement	Top-level Requirement to perform HM functions required by the PROGNOS project.
2	1	ManageHypotheses	Control exponential growth in fusion hypotheses created from incoming data.
3	1.1	ManageHypothesisCreation	Create hypotheses within a structured framework.
4	1.2	ManageHypothesisModification	Provide method to update hypotheses from incoming data.
5	1.3	ManageHypothesisAdministration	Provide method to archive hypotheses.
6	1.4	ManageHypothesisStorage	Provide interface with Knowledge Storage Module to store/retrieve hypotheses.
7	2	DiscoverHypotheses	Posit most likely and asymmetrical hypothesis from incoming data.
8	2.1	ProposeHypothesis	Identify most likely hypothesis given data to date.
9	2.2	EvolveHypothesis	Produce original hypotheses from existing hypotheses by performing mutation of attributes.

3. Establish Requirements

Establishing requirements ensures all activities in the final model can be traced to a stakeholder demand and lays the foundation for responsible development. Table 1 summarizes the requirements for the HMM and provides a first-level decomposition of its two subcomponents.

These requirements serve as statements to guide the design of the system to meet the vision of the stakeholder. The top-level specification shown in Table 1 is further delineated in separate Requirements Tables for the HME and HDE. As these are redundant in name and ID, they are not shown here.

4. Model Behavior

The *HMM Activities* diagram given in Figure 6 shows the "swim lanes" for each of the HMM activities and specifies component allocations for those processes. In this case parallel control is applied to all activities upon system initiation. External processes that affect the HMM include System Operator initiation of a Query or initiation of a change in environment or context as described in [2]. These input controls are shown as accept-event action nodes. System initialization and startup are not included in this discussion in the interest of brevity, but are a necessary precursor for operations.

From the below figure is it is clear that the HME and HDE can each operate independently with activities that will function in parallel. The following subsections continue the decomposition of these functions.

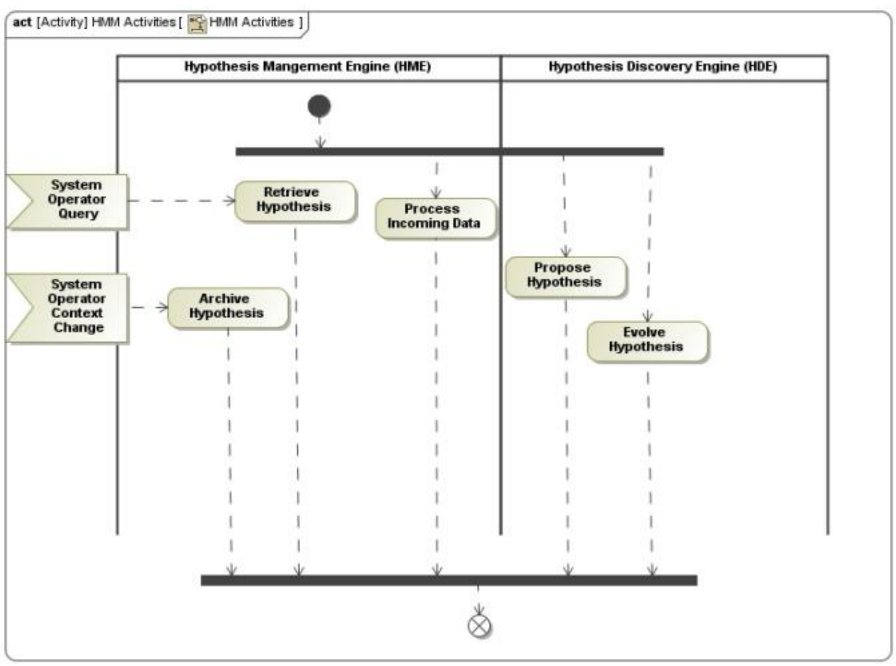

Figure 6 - HMM Activities

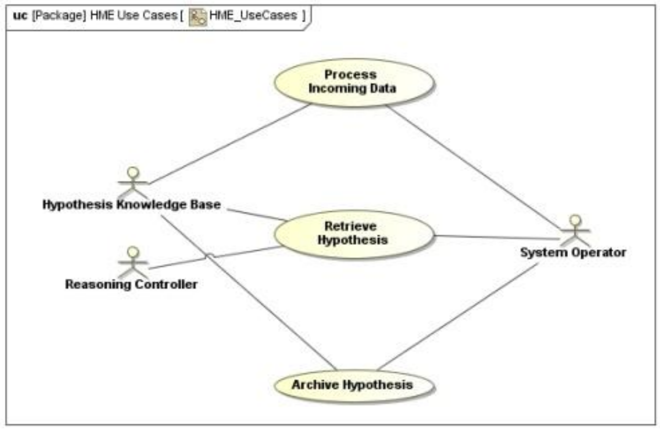

Figure 7 - HME Use Cases

Hypothesis Management Engine

The Hypothesis Management Engine of the HMM 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. Figure 7 captures the Use Cases describing the actions of the HME.

These primary activities lead to the following detailed activity diagrams that will serve as templates for coding HMM software.

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 8. 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.

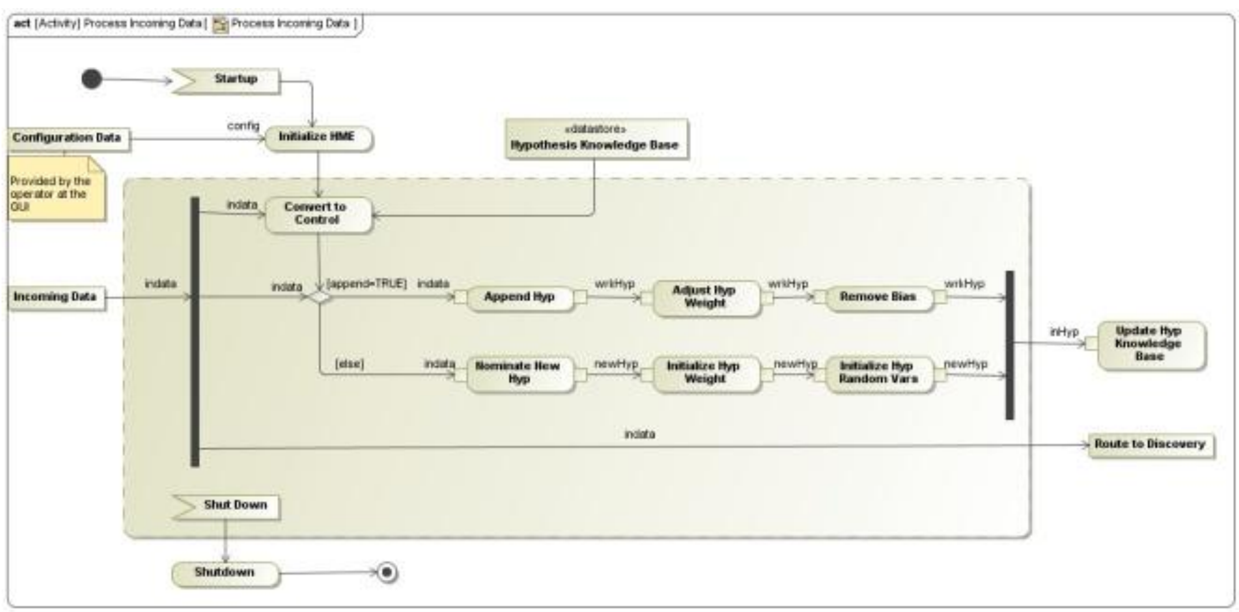

Figure 8 - Process Incoming Data Activity

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 HME via the Reasoning Controller. The frequency of arrival for data tokens is such that this process is modeled as a streaming event. An arriving data token initially travels along three parallel paths from the first fork, *Convert to Control*, *Append Decision* node, and *Route to Discovery*.

In the *Convert 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 and/or its weight matrix. A preliminary coarse filtering ensures the hypotheses compared are relevant to the current context. If an incoming report describes a unit not currently represented within PROGNOS, the *Convert to Control* returns a FALSE value for the hypothesis and begins the process to *Nominate New Hyp*. If the new data describes a unit already in the system, an additional check is performed to determine if the information updates an existing hypothesis, or introduces a new hypothesis. 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 *Append Decision* node, the data token travels the upper sequence of sub-activities in Figure 8. In the first, *Append Hyp*, the hypothesis flagged in the *Convert 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 type of data and its source.

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 middle sequence of sub-activities in Figure 8 in which a new hypothesis is 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 diagram, incoming data is passed directly to the HDE for additional processing.

Retrieve Hypothesis Activity

In response to a System Operator query, the Reasoning Controller requests candidate hypotheses from the HMM for use in the creation of the SSBN in the Model Workspace. The *Retrieve Hypothesis* activity of the HME

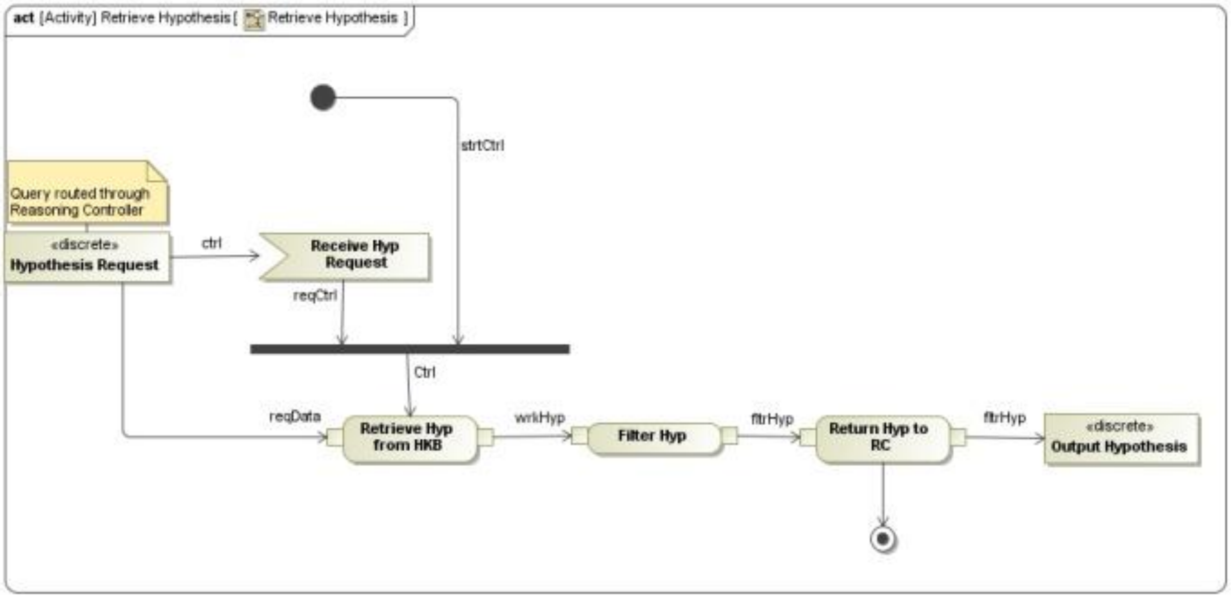

Figure 9 - Retrieve Hypothesis Activity

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 9 and described below. 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, if they exist, 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 applicable 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. Returned candidate hypotheses are prioritized by comparison with the priority vector provided by the System Operator in the query. 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. The filtering and pruning function performed in this sub-activity prevents the SSBN 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. A stylized example of these functions is detailed in [2].

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 a fixed interval to allow updates to the Model Workspace as additional data arrives in PROGNOS.

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

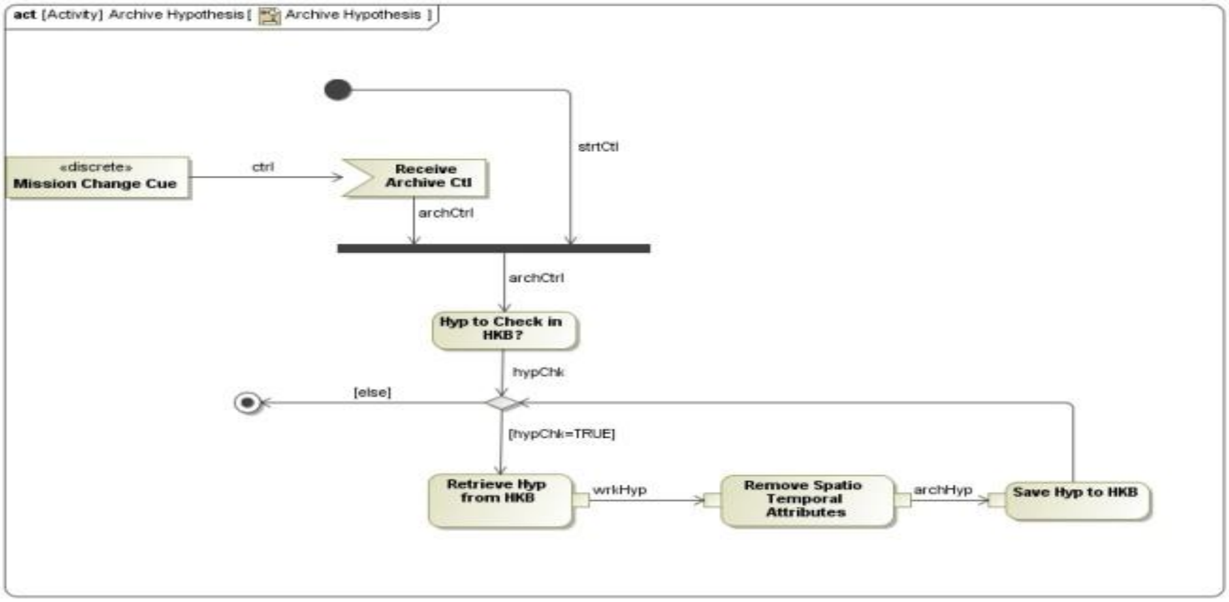

Figure 10 - Archive Hypothesis Activity

Archive Hypothesis activity shown in Figure 10 allows the 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 of the attribute fields associated with spatial and temporal data.

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 that have not been processed. 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. Concurrently, the applicable fields of the weight vector are zeroed to indicate that no data exists in the associated attribute fields.

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 HME to build upon with new incoming data.

Hypothesis Discovery Engine

The Hypothesis Discovery Engine of the HMM produces original hypotheses from observed attribute data and recommends queries for the System Operator to posit. The Use Cases of the Hypothesis Discovery Engine, *Propose Hypothesis* and *Evolve Hypotheses* are shown in Figure 11 and discussed in detail below. These functions support the System Operator in wading through profuse data and help to identify potential actions by asymmetric actors attempting to blend into background activities.

These HDE activities are detailed in the following activity diagrams that will serve as a template for coding the HMM software.

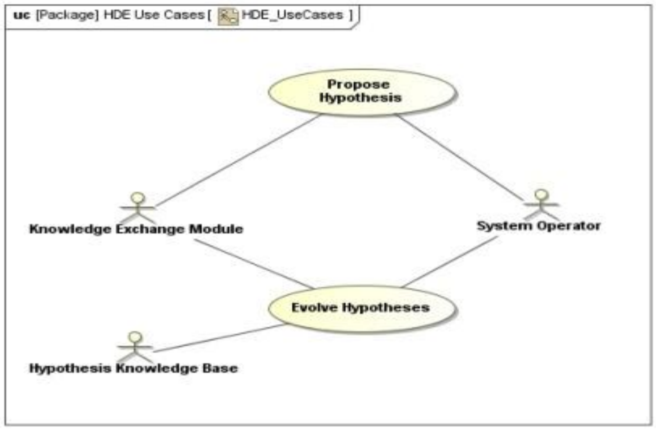

Figure 11 - HDE Use Cases

Propose Hypothesis Activity

The *Propose Hypothesis* activity, illustrated in Figure 12, 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 and credibility. 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. An example of this activity is provided in [2].

Evolve Hypothesis Activity

The *Evolve Hypothesis* activity is an original effort to create unforeseen hypotheses which may identify asymmetric actions that present a danger. 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

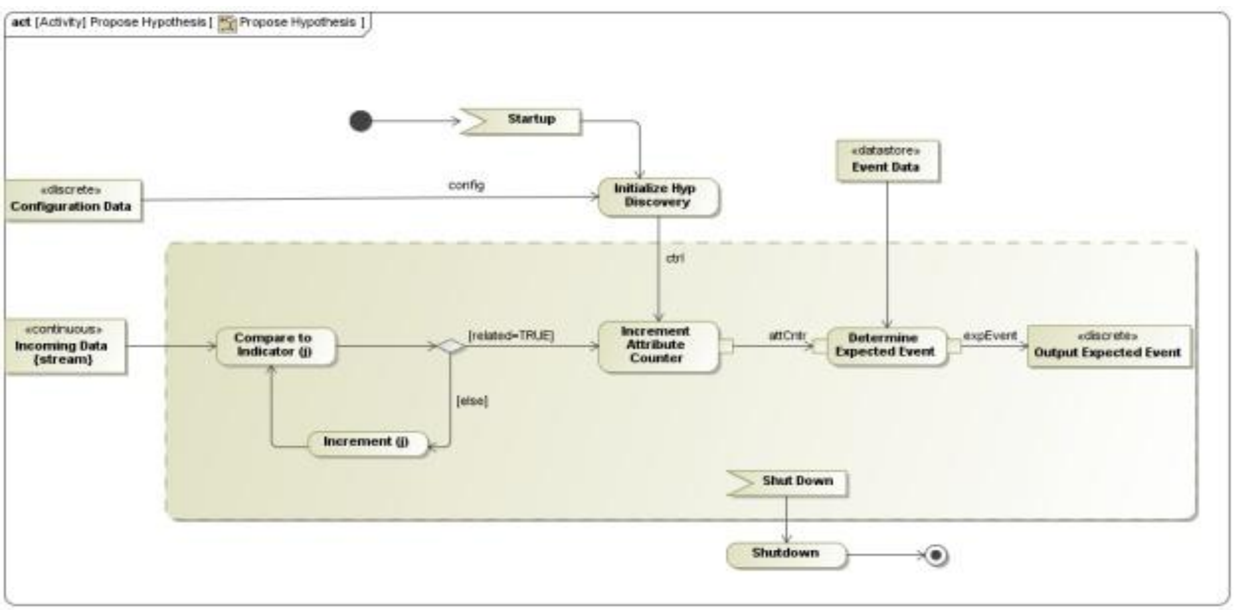

Figure 12 - Propose Hypothesis Activity

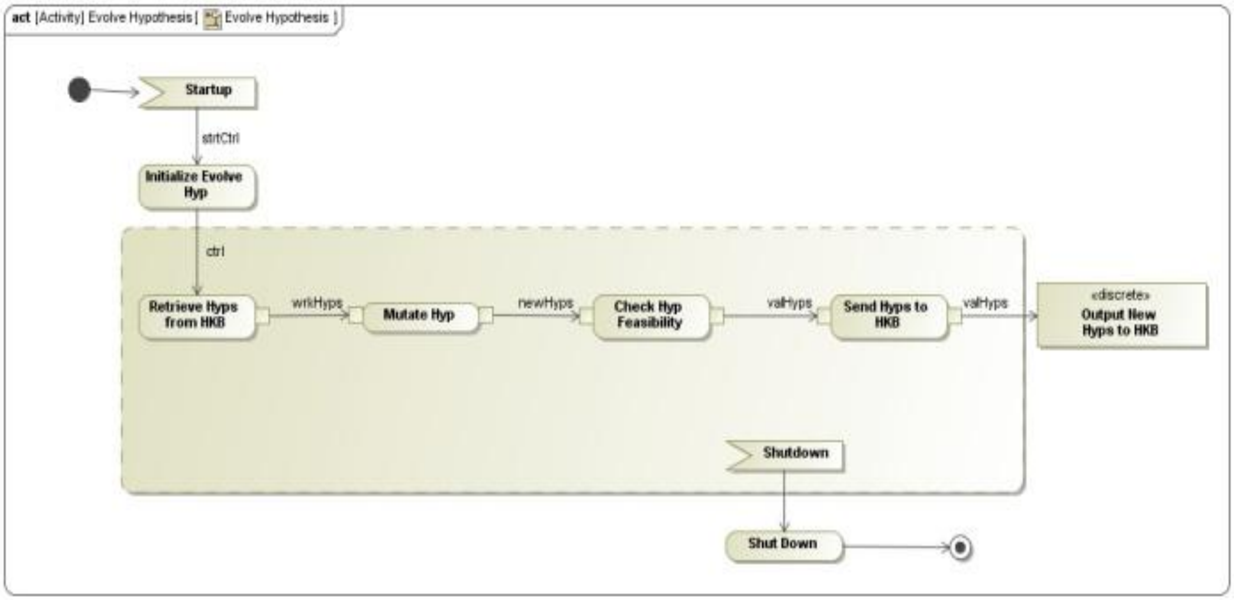

Figure 13 - Evolve Hypothesis Activity

hypotheses from the Hypothesis Knowledge Base for genetic alteration, as shown in Figure 13. A 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.

Infeasible 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. 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.

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

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.

HMM Interaction with PROGNOS

The Hypothesis Management Module interacts with the rest of the PROGNOS system primarily through the System Operator-induced query process as detailed in [2]. 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.

5. Conclusions

Using the OMG MBSE methodology, we have developed a top-level architecture for a Hypothesis Management Module for the ONR PROGNOS system. The architecture meets the requirements set forth in the PROGNOS introductory presentation. As the project moves toward coding and interaction with other system components, three focus areas emerge as the next steps in the model.

Information flow

Decomposition of model behavior will be further detailed through the use of internal block diagrams specifying control token and data flow between system components.

Parametrics

The Engineering Analysis and Item & Interface Definition packages will be further populated and specified to ensure appropriate metrics are available for system configuration and testing. Additionally, these attributes will be aligned with traits of the overall PROGNOS project to ensure the HMM performs satisfactorily within the scope of the entire project.

Simulation

Prior to inclusion in the PROGNOS domain, the HMM must be tested to minimize risk that it will adversely affect the rest of the project. This will be accomplished through a simulation that generates inputs simulating the actions of the other PROGNOS components and the external environment. A subset of diagrams will detail the composition of the simulation, its processes, and interconnections with the HMM.

The PROGNOS Hypothesis Management Module is designed to effectively manage the creation, modification, administration, storage and movement of hypotheses and to ensure that only attributes and units relative to the current context are presented for inferential reasoning. Through rigorous application of a detailed MBSE methodology, the final system will perform the critical functions of efficient creation, revision, movement, filtering, and archiving of hypotheses, providing a key enabler for the overall PROGNOS capability to support situation awareness of tactical Systems Operators.

6. References

1. 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 Symposium* . Washington D.C., USA.

2. Haberlin, R. J., Costa, P., & Laskey, K. (2010). Hypothesis Management in Support of Inferential Reasoning. *Proceedings of the 15th International Command and Control Research and Technology Symposium.* Santa Monica, CA, USA.

3. Friedenthal, S. M. (2008). *OMG Systems Modeling Language (OMG SysML) Tutorial.* The Netherlands: INCOSE.

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

5. 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.

Kathryn Blackmond Laskey is Associate Professor of Systems Engineering and Operations Research and Associate Director of the Center of Excellence in Command, Control, Communications, Computing and Intelligence at George Mason University. Dr. Laskey teaches courses in systems engineering, computational decision theory, and decision support. She has published extensively in the areas of inference, knowledge representation, learning, and information fusion. She developed Multi-Entity Bayesian Networks (MEBN), a language and logic that extends first-order logic to support probability. She was a key contributor to the development of PR-OWL, an upper ontology that allows MEBN theories to be represented in OWL ontologies.

Paulo Cesar G Costa is Affiliate Professor of Systems Engineering and Operations Research and Research Assistant Professor of the Center of Excellence in Command, Control, Communications, Computing and Intelligence at George Mason University. Dr. Costa is a retired Brazilian Air Force Flight Officer with extensive experience in electronic warfare, C4I, operations research and military decision support. He teaches courses in decision theory and systems engineering, and has developed PR-OWL, a probabilistic extension of the OWL ontology language. As an invited professor at University of Brasilia, he was a key contributor to the development of UnBBayes-MEBN, an implementation of the MEBN probabilistic first-order logic.