An Extended Maritime Domain Awareness Probabilistic Ontology Derived from Human-aided Multi-Entity Bayesian Networks Learning

Cheol Young Park, Kathryn Blackmond Laskey, Paulo C. G. Costa
The Sensor Fusion Lab & Center of Excellence in C4I
George Mason University, MS 4B5
Fairfax, VA 22030-4444 U.S.A.
cparkf@masonlive.gmu.edu, [klaskey, pcosta]@gmu.edu

Abstract*— Ontologies have been commonly associated with representing a domain using deterministic information. Probabilistic Ontologies extend this capability by incorporating formal probabilistic semantics. PR-OWL is a language that extends OWL with semantics based on Multi-Entity Bayesian Networks (MEBN), a Bayesian probabilistic logic. Developing probabilistic ontologies can be greatly facilitated by the use of a modeling framework such as the Uncertainty Modeling Process for Semantic Technology (UMP-ST). An example of using UMP-ST was the development of a probabilistic ontology to support PROGNOS (PRobabilistic OntoloGies for Net-Centric Operational Systems), a system that supports Maritime Domain Awareness (MDA). The PROGNOS probabilistic ontology provides semantically aware uncertainty management to support fusion of heterogeneous input and probabilistic assessment of situations to improve MDA. However, manually developing and maintaining a probabilistic ontology is a labor-intensive and insufficiently agile process. Greater automation through a combination of reference models and machine learning methods may enhance agility in probabilistic situation awareness (PSAW) systems. For this reason, a process for Human-aided MEBN Learning in PSAW (HMLP) was suggested. In previous work, we used UMP-ST to develop the PROGNOS probabilistic ontology. This paper presents an extended PROGNOS probabilistic ontology developed using HMLP. The contribution of this research is to introduce the extended PROGNOS probabilistic ontology and present a comparison between two processes (UMP-ST and HMLP).

Keywords—Probabilistic Ontology; Maritime Domain Awareness; Predictive Situation Awareness; Bayesian Networks; Multi-Entity Bayesian Networks; Uncertainty Modeling Process for Semantic Technology; Human-aided Machine Learning

I. INTRODUCTION

In information science, integration of heterogeneous, distributed, and unstructured information is a difficult and complex challenge. A major issue is ensuring information compatibility, for which ontologies have become a standard solution [18]. Traditional ontologies are limited to deterministic knowledge. Probabilistic Ontologies (POs) move beyond this limitation by incorporating formal probabilistic semantics. Probabilistic OWL (PR-OWL) [19] is a probabilistic ontology language that extends OWL with semantics based on Multi-Entity Bayesian Networks (MEBN), a Bayesian probabilistic logic [1]. PR-OWL has been extended to PR-OWL 2 [14], which provides a tighter link between the deterministic and probabilistic aspects of the Ontologies. MEBN is flexible enough to represent a variety of complex and uncertain situations. MEBN has been applied to systems [2][3][4][5][6][7] for Predictive Situation Awareness (PSAW), providing the ability to estimate and predict aspects of a temporally evolving situation.

Developing probabilistic ontologies can be greatly facilitated by the use of a modeling framework such as the UMP-ST, a modeling process for constructing a probabilistic ontology [13]. The UMP-ST consists of four main disciplines: (1) *Requirement*, (2) *Analysis & Design*, (3) *Implementation*, and (4) *Test*. UMP-ST was used to develop a probabilistic ontology to support PROGNOS (PRobabilistic OntoloGies for Net-Centric Operational Systems), a system to support *Maritime Domain Awareness* (MDA). The existing system for MDA (e.g., US Navy's Net-Centric infrastructure, FORCENet) is used to fuse lower-level multi-sensor data, analyze the fused data by human analysts, and support decision-making for naval operations. However, the era of big data requires greater automation. The PROGNOS probabilistic ontology [7] supports ingestion of lower-level data, fusion of heterogeneous input, and probabilistic assessment of situations to improve MDA. PROGNOS is a prototype system that aims especially to identify threatening targets (e.g., terrorists and terrorist-ships).

Manually developing and maintaining a probabilistic ontology is a labor-intensive and insufficiently agile process. Furthermore, it is important to make use of data when available. Therefore, greater automation through a combination of reference models and machine learning methods has the potential to enhance agility and effectiveness in modeling a probabilistic ontology for PSAW. For this reason, a process for Human-aided MEBN Learning in PSAW (HMLP) has been suggested [20]. HMLP contains three supporting methodologies, MEBN-RM [10], a reference MEBN model for PSAW [8], and MEBN learning algorithms [9][10]. These component methodologies enable efficient and effective modeling. MEBN-RM and the reference model are introduced in Section 2 below.

In previous work, we used UMP-ST to develop the PROGNOS PO. This paper presents an extended PROGNOS PO developed using HMLP. In the following sections, the paper (1) provides background information, (2) introduces the original PROGNOS PO derived from UMP-ST, (3) presents the extended PROGNOS PO derived from HMLP, and (4) compares two processes.

II. Background

This section introduces (1) MEBN, (2) MEBN-RM Mapping Model, (3) A Reference MEBN Model for PSAW, (4) Uncertainty Modeling Process for Semantic Technology (UMP-ST), and (5) Human-aided MEBN learning in PSAW (HMLP). HMLP assumes input data based on the relational model (RM) as its data schema. We choose RM because it is the most popular database model and has the necessary expressive power to represent entities and their relationships. It is necessary to define how to convert elements of RM to elements of MEBN, so a mapping rule between MEBN and RM, called MEBN-RM, was developed. Also, we introduce a reference MEBN model for PSAW which provides a set of basic templates to support the design of a MEBN model for PSAW. HMLP is a modification of UMP-ST, so UMP-ST is introduced in this section. Some of the following background summaries are taken from [20].

A. Multi-Entity Bayesian Networks

MEBN is a compact model combining Bayesian networks (BN) with First-order logic (FOL) to represent repeated structures in a joint distribution representing domain knowledge. MEBN is a highly expressive model for treating uncertainty and complex forms of data and information. A MEBN model, called an MTheory, is composed of fragments, called MFrags. An MFrag consists of a set of resident nodes, a set of context nodes, a set of input nodes, an acyclic directed graph for the nodes, and a set of class local distributions (CLD) for the nodes. A resident node is a random variable which is associated with a function or predicate of FOL and whose class local distribution is resident in an MFrag. A context node is derived from a resident node and determines conditions under which the class local distribution defined in the MFrag is valid. An input node has its distribution defined elsewhere and conditions the class local distribution defined in the MFrag. Nodes for an acyclic directed graph are associated with resident and input nodes. An FOL function or predicate of a resident node contains ordinary variables, which can be replaced with entity identifiers to generate multiple instances of the RVs. MFrags in an MTheory are used to generate instances of fragments of BN. The fragments of BN are combined to form a Bayesian network, called a situation-specific Bayesian Network (SSBN). An MTheory can be used to generate an unbounded number of different SSBNs. Further information about MEBN can be found in [1].

B. MEBN-RM Mapping Model

MEBN-RM [10] is a mapping model which provides a specification for how to convert relational databases [11][12] to MTheories [1]. The relational model (RM) is the most popular database model. MEBN-RM provides an entity mapping between a relation in RM and an entity in MEBN, a resident node mapping between an attribute in RM and a resident node in MEBN, an MFrag mapping between a relation in RM and an MFrag in MEBN, and an MTheory mapping between an RM and an MTheory. An MTheory can be constructed automatically from a relational database by using mapping rules in MEBN-RM. Therefore, MEBN-RM can support a MEBN learning algorithm, which develops an MTheory from a dataset, or an MTheory developer, who aims to develop an MTheory using domain knowledge and MEBN knowledge. HMLP exploits MEBN-RM for efficient development of an MTheory.

C. A Reference MEBN Model for PSAW

A reference model is an abstract framework to which a developer refers in order to develop a specific model. A reference MEBN model for PSAW is a reference model for a PSAW-MTheory which specifies references for MFrags, RVs, relationships of RVs, and entities. The reference MEBN model for PSAW can support the design of a PSAW-MTheory and improve the quality of the PSAW-MTheory. The references for entity are classified into five categories (Time entity *T*, Observer entity *OR*, Sensor entity *SR*, Target entity *TR*, and Reported target entity *RT*). Entities derived from these categories describe a situation in which an observer *OR* observes a target *TR* and interprets it as a reported target *RT* using a sensor *SR* at a certain time *T* [20]. The reference MEBN model for PSAW provides some referring random variables (RV), called PSAW-RVs. PSAW-RVs are classified into five categories (Observing condition RV, Reported object RV, Target object RV, Situation RV, and Context RV). These PSAW-RVs are defined in five types of MFrags (Observing condition MFrag, Report MFrag, Target MFrag, Situation MFrag, and Context MFrag). An observing condition RV defined in an observing condition MFrag represents probabilistic knowledge about conditions of a sensor (e.g., maintenance conditions for a sensor). A reported object RV defined in a report MFrag represents probabilistic knowledge about a relation or an attribute of observed targets (e.g., a reported target size). A target object RV defined in a target MFrag represents probabilistic knowledge about a relation or an attribute for actual targets (e.g., an actual target size). A situation RV defined in a situation MFrag represents probabilistic knowledge about situations of targets (e.g., a collaborating situation for targets). A context RV defined in a context MFrag represents probabilistic knowledge about conditions under which the class local distribution defined in the MFrag is valid. For example, an RV Predecessor(*pre_t*, *t*) can be a context RV. The context RV Predecessor(*pre_t*, *t*) means that the time interval *pre_t* occurs immediately before the time interval *t*. More specific information for the reference MEBN model for PSAW can be found in [20].

D. Uncertainty Modeling Process for Semantic Technology (UMP-ST)

UMP-ST is a framework to support the design of a probabilistic ontology [13]. The PROGNOS probabilistic ontology was developed using UMP-ST. UMP-ST provides processes for constructing a probabilistic ontology through four disciplines: (1) *Requirement*, (2) *Analysis & Design*, (3) *Implementation*, and (4) *Test*.

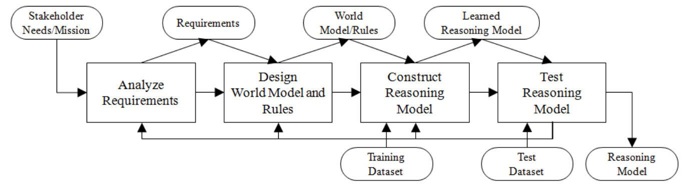
In the *Requirement* discipline, requirement statements are defined. The requirement statements can contain goals, queries, and evidence for a probabilistic ontology. Objectives to be achieved by reasoning with the probabilistic ontology are specified by statements for goals (e.g., detect a ship of interest). To achieve the objectives, specific query statements are specified in this discipline (e.g., what is the type of a ship?). To support the queries, evidence associating with the queries is determined in this discipline (e.g., an appearance of a ship). In the *Analysis & Design* discipline, entities, attributes, relationships, and probabilistic rules are defined. These are used to support the goals, queries, and evidence. For example, we are developing a probabilistic ontology, which aims to detect a ship of interest (the goals). The goal is achieved by identifying the type of a ship (the queries) given information about the appearance of the ship (the evidence). For this situation, a ship entity is required. Also, type and appearance attributes for the ship entity are required. Suppose that the appearance attribute may depend on the type attribute. This is specified by a probabilistic rule. In the *Implementation* discipline, a probabilistic ontology is developed using results from the previous disciplines. A probabilistic ontology based on MEBN is used to reason about uncertainty. Therefore, a probabilistic ontology contains OWL classes based on elements from MEBN such as an MFrag, an MTheory, a node, a probability distribution, and an entity. In this step, these OWL classes are defined. For example, the ship entity defined in the previous discipline is mapped to an entity type indicating a ship in the probabilistic ontology. The attributes ship appearance and ship type are mapped to random variables ship appearance and ship type, respectively. The probabilistic rule for the attributes ship appearance and ship type is converted to the joint probability for the random variables ship appearance and ship type. The random variables ship appearance and ship type may belong to an MFrag representing attributes of a ship. The MFrag ship and other MFrags related with a maritime domain may integrate into an MTheory representing a maritime situation. The *Test* discipline is used to assess the probabilistic ontology developed in the *Implementation* discipline. More specific information for UMP-ST can be found in [13].

E. Human-aided MEBN learning in PSAW (HMLP)

HMLP is a framework which aims the development of a probabilistic ontology in PSAW. HMLP provides specific development steps and supporting methods (MEBN-RM, the reference MEBN model for PSAW, and MEBN learning). HMLP improves MEBN learning by providing expert knowledge which is used to limit the search space of parameters, variables, and structures for a probabilistic ontology in PSAW.

Similar to the four disciplines of UMP-ST, HMLP contains four steps (Fig. 1): (1) *Analyze Requirements*, (2) *Design World Model and Rules*, (3) *Construct Reasoning Model*, and (4) *Test Reasoning Model*. (See a full discussion of HMLP in [20]). A summary of HMLP is presented below.

Fig. 1. Process for Human-Aided MEBN Learning (This figure was taken from [20] and was modified)



Stakeholders who request the development of a reasoning model or a probabilistic ontology provide needs and/or missions as inputs of HMLP. An output from the end of HMLP is a reasoning model (in our case, a probabilistic ontology for PSAW). The followings describe the four steps in HMLP. (1) In the *Analyze Requirements* step, requirements which contain goals to be achieved, queries to answer, and evidence to be used in answering queries are defined. Also, the requirements include performance criteria, which are used in the *Test Reasoning Model* step, to evaluate the probabilistic ontology. (2) In the *Design World Model and Rules* step, a world model and rules are developed using the requirements in the previous step. This step contains two sub-steps (*Design World Model* step and *Design Rules* step). The *Design World Model* step defines the world model which may include entities, attributes, and relations (e.g., RM) using the requirements, domain knowledge and/or existing data schemas. The world model is used to identify rules. In the *Design Rules* step, the rules or influencing relationships between attributes in the world model are defined. (3) In the *Construct Reasoning Model* step, a probabilistic ontology is constructed using a training dataset, the world model, and the rules. This step includes two sub-steps (*Map to Reasoning Model* step and *Learn Reasoning Model* step). The *Map to Reasoning* step maps the world model and rules to a candidate probabilistic ontology. The *Learn Reasoning Model* uses a MEBN learning method to learn the probabilistic ontology from a training dataset. (4) The *Test Reasoning Model* step evaluates the learned probabilistic ontology in the previous step to determine whether to accept it. The accepted probabilistic ontology is a final result from HMLP.

III. PROGNOS PO VIA UMP-ST

To develop the PROGNOS PO, three iterations of the four steps in UMP-ST (*Requirement*, *Analysis & Design*, *Implementation*, and *Test*) were performed [14]. The following sub-sections summarize the four steps in UMP-ST to develop the PROGNOS PO.

A. Requirements

The *Requirement* step identifies requirements containing goals, queries, and evidence for a probabilistic ontology. The requirements for the PROGNOS PO were developed gradually over the three iterations. In the first iteration, a simple requirement regarding a ship of interest was identified [7]. In the second iteration, requirements for two types of terrorist-ships were defined. In the third iteration, requirements for crew members in a ship of interest were specified. The following list shows part of the resulting requirements [14].

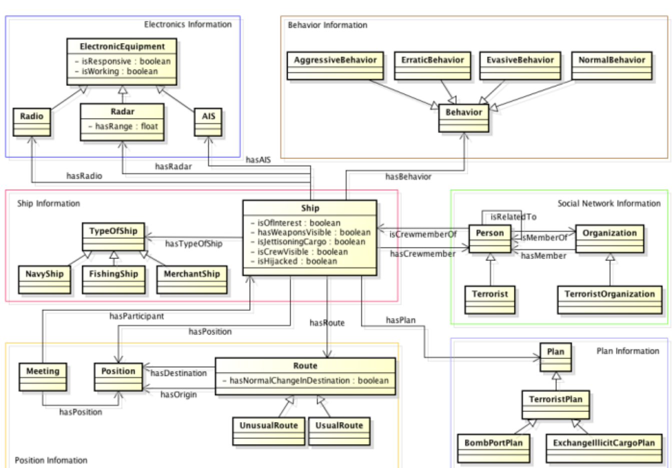
1. Identify if a ship is of interest,
1.1 Is the ship being used to exchange illicit cargo?
1.1.1 Was the ship hijacked?
1.1.2 Does the ship have a terrorist crew member?
1.1.2.1 Is the crew member associated with any terrorist organization?
…
1.2 Is the ship being used as a suicide ship to bomb a port?
…

The main goal was to identify a ship of interest (i.e., a terrorist-ship). In this requirement, we assumed the ship of interest may exchange illicit cargo and/or be used as a suicide ship to bomb a port. To support this goal, we needed to identify the type of a crew member of a ship. If the type of a crew member is a terrorist, the ship is highly likely to be a terrorist-ship. To identify whether a crew member is a terrorist, we can check whether the crew member is associated with any terrorist organization.

B. Analysis & Design

This step defines the types of entities, their properties and relationships, and the rules that apply to them, i.e., the semantics of the domain model. The Unified Modeling Language (UML) diagrams can provide a convenient and understandable visualization of the classes and relationships for the model semantics. The requirements defined in the previous step are used to develop the model semantics. Thus, entities, attributes for the entities, and relationships between the entities were identified. For example, from Requirement 1, an entity was derived (i.e., a ship) and an attribute of the entity was derived (i.e., the type of a ship). From Requirement 1.1.2, a new entity was derived (i.e., a (terrorist) person) and a relationship between the entities was derived (i.e., a ship has a crew (terrorist) member). In the second iteration, Carvalho [14] developed the model represented by UML as shown in Fig. 2.

Fig. 2. Entities, their attributes, and relations for the MDA model after the second iteration (This figure provided by permission of Carvalho [14])



The classes and relationships form six natural groups (i.e., *Electronics*, *Behavior*, *Ship*, *Position*, *Plan*, and *Social Network*). The ship types are *NavyShip*, *FishingShip*, and *MerchantShip*. Ship routes are *UnusualRoute* and *UsualRoute*. Two ships can meet each other at a position. A ship can use electronic devices such as *Radio*, *Radar*, and *AIS* (Automatic Identification System). A ship can show *behavior* such as *Aggressive*, *Erratic*, *Evasive*, and *Normal*. A ship can have a (terrorist) crewmember who may belong to a (terrorist) organization. A ship can have a terrorist plan such as *BombPort* and *ExchangeIllicitCargo*.

After developing the model semantics, conditional rules were identified. There were three iterations of this process. The following list shows a few of the conditional rules from [14].

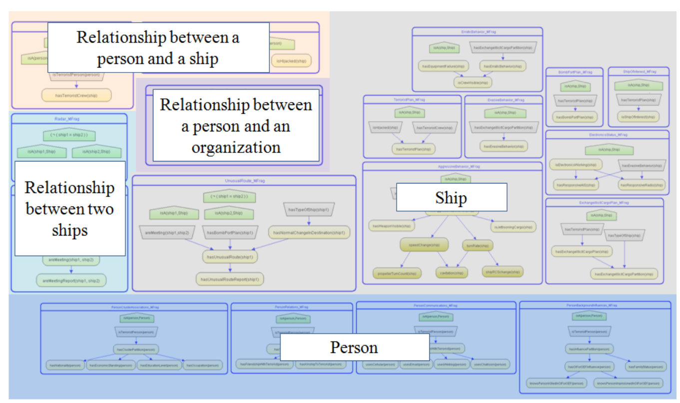
1.(a) *If a crew member is a member of a terrorist organization, then it is more likely that he is a terrorist.*
1.(b) *If an organization has a terrorist member, it is more likely that it is a terrorist organization.*
…
4.(a) *Research shows that if a crew member has a relationship with terrorists, there is a 68% chance that he has a friend who is a terrorist.*
…

These conditional rules were derived from extensive research about terrorism [16] and from the knowledge provided by a domain expert. These rules were used to develop the PROGNOS PO.

C. Implementation

In the *Implementation* step, the PROGNOS PO was designed. The PROGNOS PO can be found in [14][15]. Fig. 3 shows the PROGNOS PO containing five groups of MFrags.

Fig. 3. Original PROGNOS probabilistic ontology



The first set of MFrags is for a ship of interest. It includes nine MFrags *Aggressive Behavior*, *Terrorist Plan*, *Evasive Behavior*, *Erratic Behavior*, *Unusual Route*, *Bomb Port Plan*, *Ship Of Interest*, *Electronics Status*, and *Exchange Illicit Cargo Plan*. These MFrags are used to reason about properties of a ship (e.g., unusual behavior and an illegal plan). The second set of MFrags is for a person of interest. It includes four MFrags *Person Communications*, *Person Background Influences*, *Person Cluster Associations*, and *Person Relations*. These MFrags are used to identify a person who may communicate with a terrorist, has a suspicious background and history, and has a relationship with a terrorist. The third set of MFrags is for information of relationships between two ships. It includes two MFrags, *Radar* and *Meeting*. These MFrags are used to identify whether one ship is within radar range of another ship and whether two ships are meeting. The fourth set of MFrags is for information about the relationship between a person and an organization. It includes one MFrag *Terrorist Person* in which a person who belongs to an organization is identified. The last set of MFrags is for information about a relationship between a person and a ship. It includes two MFrags *Has Terrorist Crew* and *Ship Characteristics*. These MFrags are used to link a person and a ship, and to identify whether a ship has a terrorist crew member.

The following list shows part of a partial PROGNOS PO containing information about MFrags (*F*), context nodes (*C*), resident nodes (*R*), resident parent nodes (*RP*), and input parent nodes (*IP*). Note that a partial probabilistic ontology doesn't contain a class local distribution and domain information for a random variable.

PO 1: Original PROGNOS probabilistic ontology

```
1     [F: ErraticBehavior_MFrag
2         [C: isA(ship,Ship)]
3         [R: hasErraticBehavior(ship) [IP: hasExchangeIllicitCargoPartition(ship)]]
4         [R: hasEquipmentFailure(ship)]
5         [R: isCrewVisible(ship)[RP: hasErraticBehavior(ship)][RP: hasEquipmentFailure(ship)]]
6     ]
7     [F: TerroristPerson_MFrag
8         [C: isA(person,Person), isA(org,Organization)]
9         [R: isTerroristOrganization(org)[RP: isTerroristPerson(person), isMemberOfOrganization(person, org)]]
10        [R: isTerroristPerson(person)][R: isMemberOfOrganization(person, org)]
11    ]
12    [F: ShipCharacteristics_MFrag
13        [C: isA(ship,Ship), isA(person,Person)]
14        [R: hasCrewMember(ship, person)] [R: hasTypeOfShip(ship)][R: isHijacked(ship)]
15    ]
16    [F: EvasiveBehavior_MFrag
17        [C: isA(ship,Ship)]
18        [R: hasEvasiveBehavior(ship)[IP: hasExchangeIllicitCargoPartition(ship)]]
19    ]
20    [F: PersonCommunications_MFrag
21        [C: isA(person,Person)]
22        [R: communicatesWithTerrorist(person)[IP: isTerroristPerson(person)] ]
23        [R: usesChatroom(person) [RP: communicatesWithTerrorist(person)]]
24        [R: usesEmail(person) [RP: communicatesWithTerrorist(person)]]
25        [R: usesCellular(person) [RP: communicatesWithTerrorist(person)]]
26        [R: usesWeblog(person) [RP: communicatesWithTerrorist(person)]]
27    ]
28    [F: PersonBackgroundInfluences_MFrag
29        [C: isA(person,Person)]
30        [R: hasInfluencePartition(person) [IP: isTerroristPerson(person)]]
31        [R: knowsPersonImprisionedInOIForOEF(person) RP: hasOIForOEFInfluence(person)]]
32        [R: hasFamilyStatus(person) [RP: hasInfluencePartition(person)]]
33        [R: hasOIForOEFInfluence(person) [RP: hasInfluencePartition(person)]]
34        [R: knowsPersonKilledInOIForOEF(person) [RP: hasOIForOEFInfluence(person)]]
35    ]
36    [F: AggressiveBehavior_MFrag
37        [C: isA(ship,Ship)]
38        [R: hasAggressiveBehavior(ship) [IP: hasBombPortPlan(ship), hasExchangeIllicitCargoPartition(ship)]]
39        [R: hasWeaponVisible(ship) [RP: hasAggressiveBehavior(ship)]]
40        [R: isJettisoningCargo(ship) [RP: hasAggressiveBehavior(ship)]]
41        [R: speedChange(ship) [RP: hasAggressiveBehavior(ship)]]
42        [R: turnRate(ship) [RP: hasAggressiveBehavior(ship)]]
43        [R: propellerTurnCount(ship) [RP: speedChange(ship)]]
44        [R: cavitation(ship) [RP: speedChange(ship)][RP: turnRate(ship)]]
45        [R: shipRCSchange(ship)  [RP: turnRate(ship)]]
46    ]
47    [F: ShipOfInterest_MFrag
48        [C: isA(ship,Ship)] [R: isShipOfInterest(ship) [IP: hasTerroristPlan(ship)]]
49    ]
50    [F: ExchangeIllicitCargoPlan_MFrag
51        [C: isA(ship,Ship)]
52        [R: hasExchangeIllicitCargoPlan(ship) [IP: hasTerroristPlan(ship)]]
53        [R: hasExchangeIllicitCargoPartition(ship)
54                [IP: hasTypeOfShip(ship)][RP: hasExchangeIllicitCargoPlan(ship)]]
55    ]
56    [F: PersonRelations_MFrag
57        [C: isA(person,Person)]
58        [R: hasKinshipToTerrorist(person) [RP: hasTerroristBeliefs(person)]]
59        [R: hasFriendshipWithTerrorist(person) [RP: hasTerroristBeliefs(person)]]
60        [R: hasTerroristBeliefs(person) [IP: isTerroristPerson(person)]]
61    ]
62    [F: Meeting_MFrag
63        [C: isA(ship1,Ship), isA(ship2,Ship)]
64        [C: ( ¬ ( ship1 = ship2 ) )]
65        [R: areMeeting(ship1, ship2) [IP: hasExchangeIllicitCargoPartition(ship1)]]
66        [R: areMeetingReport(ship1, ship2) [RP: areMeeting(ship1, ship2)]]
67    ]
68    [F: BombPortPlan_MFrag
69        [C: isA(ship,Ship)] [R: hasBombPortPlan(ship) [IP: hasTerroristPlan(ship)]]
70    ]
71    [F: HasTerroristCrew_MFrag
72        [C: isA(ship,Ship), isA(person,Person)]
73        [C: hasCrewMember(ship,person)]
74        [R: hasTerroristCrew(ship) [IP: isTerroristPerson(person)]]
75    ]
76    [F: UnusualRoute_MFrag
77        [C: isA(ship2,Ship), isA(ship1,Ship)]
78        [C: ( ¬ ( ship1 = ship2 ) )]
79        [R: hasUnusualRoute(ship1)
80                [RP: hasNormalChangeInDestination(ship1)]
81                [IP: hasBombPortPlan(ship1)][IP: areMeeting(ship1,ship2)]]
82        [R: hasUnusualRouteReport(ship1) [RP: hasUnusualRoute(ship1)]]
83        [R: hasNormalChangeInDestination(ship1) [IP: hasTypeOfShip(ship1)] ]
84    ]
85    [F: TerroristPlan_MFrag
86        [C: isA(ship,Ship)]
87        [R: hasTerroristPlan(ship) [IP: hasTerroristCrew(ship)][IP: isHijacked(ship)]]
88    ]
89    [F: ElectronicsStatus_MFrag
90        [C: isA(ship,Ship)]
91        [R: isElectronicsWorking(ship)]
92        [R: hasResponsiveRadio(ship)
93                [IP: hasEvasiveBehavior(ship)][RP: isElectronicsWorking(ship)]]
94        [R: hasResponsiveAIS(ship)
95                [IP: hasEvasiveBehavior(ship)][RP: isElectronicsWorking(ship)]]
96    ]
97    [F: Radar_MFrag
98        [C: isA(ship1,Ship), isA(ship2,Ship)] [C: ( ¬ ( ship1 = ship2 ) )]
99        [R: isWithinRadarRange(ship1, ship2)]
100   ]
101   [F: PersonClusterAssociations_MFrag
102       [C: isA(person,Person)]
103       [R: hasOccupation(person) [RP: hasClusterPartition(person)]]
104       [R: hasEducationLevel(person) [RP: hasClusterPartition(person)]]
105       [R: hasClusterPartition(person) [IP: isTerroristPerson(person)]]
106       [R: hasEconomicStanding(person) [RP: hasClusterPartition(person)]]
107       [R: hasNationality(person) [RP: hasClusterPartition(person)]]
108   ]
```

PO 1 shows the context nodes and the resident nodes in the MFrags, and the relationship between the resident nodes. For example, the MFrag *ErraticBehavior_MFrag* (Line 1~6) contains an *isA* context node and three resident nodes *hasErraticBehavior*, *hasEquipmentFailure*, and *isCrewVisible*. The resident node *hasErraticBehavior* is influenced by an input node *hasExchangeIllicitCargoPartition*. The resident node *isCrewVisible* is influenced by the resident nodes *hasErraticBehavior* and *hasEquipmentFailure*. This PROGNOS PO was tested in the next step.

D. Test

In this step, the PROGNOS PO was evaluated to determine whether to accept it. To do this, the case-based evaluation, in which various scenarios were defined and used to examine the reasoning implications of the probabilistic ontology, was used. For example, given a scenario which was developed by a subject matter expert (SME), some information (e.g., history of a target) from the scenario for a target was used as evidence for inference of the PROGNOS PO to identify some properties (e.g., whether the target is a terrorist) of the target. If the result of inference coincided exactly with the scenario from SME, we could think that the probabilistic ontology was reasonable. For this test, three qualitatively different scenarios were used [14].

After three iterations for UMP-ST, an overall test for the PROGNOS PO was performed using a simulation. In the real world situation, it is very difficult to acquire a real dataset to develop such a probabilistic ontology which contains rare events. For this reason, the simulation was used to produce a test dataset given different scenarios generated randomly. Carvalho [14] and Costa et al [15] introduced some results for this test. In such a test, it is important that knowledge used to develop a probabilistic ontology and knowledge used to develop a simulation for testing the probabilistic ontology should not be same. If they are same, the test is meaningless, because the probabilistic ontology and the simulation are same models, but just in different forms.

IV. PROGNOS PO VIA HMLP

In this section, we introduce an extended PROGNOS PO derived from the HMLP process. The following shows how the development operates.

A. Analyze Requirements

This step is not much different from the requirement step in UMP-ST. Therefore, we can reuse requirements developed from the PROGNOS project. The full requirements can be found in [14]. However, the reference MEBN model for PSAW can provide more items by which a PSAW modeler can consider predefined entities, RVs, and MFrags for PSAW. Recall the four MFrag groups from the reference model: *Reported Object*, *Observing Conditions*, *Target Object*, and *Situation*. The last of these, *Situation*, is of special note. In PSAW, understanding a situation in which targets operate for their own purposes is one of the important issues. Identifying just the type of a target is an insufficient task for PSAW. The meaning of *awareness* is not to perceive and estimate actual properties of a target but is to understand, interpret, and explain the relationships between targets. Kokar et al [17] stated: "*The main part of being aware is to be able to answer the question of "what's going on?"*". Awareness of a situation is subjective according to an observer, who is aware of the situation. The modeler, who is developing a probabilistic ontology to support PSAW, should define what situation will be considered and explained through all observation from the world. For the awareness of the PROGNOS situation, we add the following new requirement.

New Goal 1: Recognize emergency situation at sea

Query 1.1: How high is the potential terrorist threat?

Evidence 1.1.1: Ship(s) of interest

Evidence 1.1.2: Crew member(s) of interest

The new goal aims to alert a response team when the threat reaches a certain level. This will be accomplished by estimating potential terrorist attacks in the field given estimation of terrorist ships and terrorist crew members.

In HMLP, a requirement can contain a performance criterion specifying a measure of accuracy (e.g., the mean squared error or the Brier score [26]). For example, we might require that the mean squared error between ground truth and estimated results from the probabilistic ontology shall be less than a given threshold (e.g., a mean squared error < 0.1).

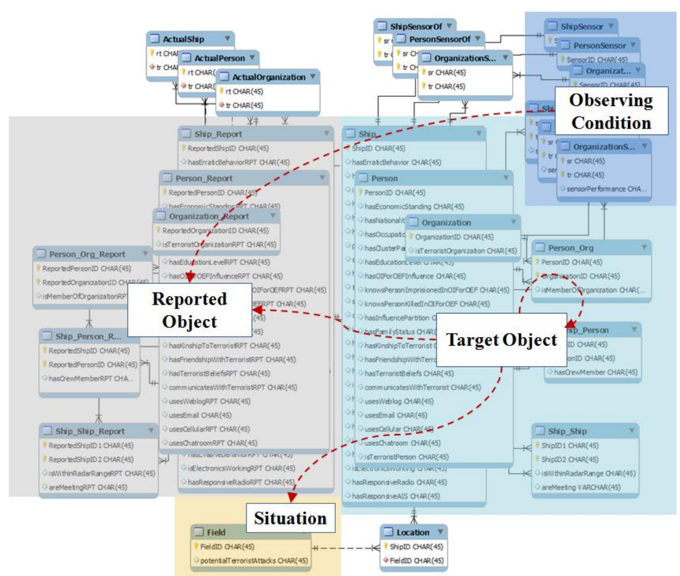
B. Design World Model and Rules

This step performs two sub-steps (*Design World Model* and *Design Rules*). The *Design World Model* step is to define a world model for PROGNOS from the requirements defined in the previous step.

In this step, the reference MEBN model for PSAW can be used to identify possible entities, random variables, and relationships between the random variables. Fig. 4 shows a PROGNOS world model represented in an EER (enhanced entity–relationship) model. We develop the PROGNOS world model using the requirements and the reference model.

The reference model suggests four groups: (1) *Reported Object*, (2) *Observing Condition*, (3) *Target Object*, and (4) *Situation*. A world model for the original PROGNOS PO included the seven relations (e.g., *Target*, *Ship*, *Person*, *Organization*, *Person_Org*, *Ship_Person*, and *Ship_Ship*). The original PROGNOS PO treated only the target object group. In other words, it did not emphasize sensing. We would expect evidence (e.g., reported objects) to be reported to estimate actual targets (e.g., target objects), so relations (i.e., *Ship_Report*, *Person_Report*, *Organization_Report*, *Ship_Ship_Report*, *Person_Org_Report*, *Ship_Person_Report*, and *ReportedTarget*) for the reported object group are added in the world model for the extended PROGNOS PO. Observations may contain observation errors influenced by observing conditions (e.g., weather). The observing condition group contains two relations *Sensor* and *SensorProperty*. In the previous step, a requirement for the awareness for a situation was added. Therefore, we added a relation *Field* for the situation group in Fig. 4. Relations (i.e., *Location*, *SensorOf*, and *ActualTarget*) which are not classified in these groups are supporting relations used to join the relations in the four groups.

Fig. 4. Part of EER Model for a PROGNOS world model



The reference model provides some rules or relationships between these groups as shown in the arrows (Fig. 4). The observing conditions group and the target object group can influence the reported object group. For example, the attribute *sensorPerformance* in the relation *SensorProperty* influenced the report attributes in the report relations *Ship_Report*, *Person_Report*, *Organization_Report*, *Ship_Ship_Report*, *Person_Org_Report*, and *Ship_Person_Report*. The arrows in Fig. 4 indicate these relationships. The following shows a few of these rules.

Rule 1: *causal* ({*hasErraticBehavior*, *sensorPerformance*}, *hasErraticBehaviorRPT*)

Rule 2: *causal* ({*isShipOfInterest*, *isTerroristPerson*}, *PotentialTerroristAttacks*)

…

Rule 1 means that two attributes *hasErraticBehavior* and *sensorPerformance* cause the attribute *hasErraticBehaviorRPT*. Rule 2 means that two attributes *isShipOfInterest* and *isTerroristPerson* cause the attribute *PotentialTerroristAttacks*.

C. Construct Reasoning Model

This step performs two sub-steps (*Map to Reasoning Model* and *Learn Reasoning Model*) to construct the PROGNOS PO. MEBN-RM provides a converting rule from RM to a probabilistic ontology. Entity relations which contain only one attribute for the primary key of the relation (e.g., *ship* and *person*) can be defined as entity types in the probabilistic ontology. Each of the attributes in the relations could be mapped to a resident node in the probabilistic ontology using MEBN-RM. For example, the attribute *hasErraticBehavior* of the relation *Ship* became the resident node *hasErraticBehavior*(*ship*).

Rules which are defined in the previous step are used to develop relationships between resident nodes in the probabilistic ontology. For example, from Rule 1, we had a conditional dependence P(*hasErraticBehaviorRPT*(*ship_report*) | *hasErraticBehavior*(*ship*), *sensorPerformance*(*shipSensor*, *ship*)). From Rule 2, we had a conditional dependence P(*PotentialTerroristAttacks*(*field*) | *isShipOfInterest*(*ship*), *isTerroristPerson*(*person*)).

We could model the extended PROGNOS PO as shown in Fig. 5 using the resident nodes, the relationships between the resident nodes, and the MFrag groups.

Fig. 5. Extended PROGNOS probabilistic ontology

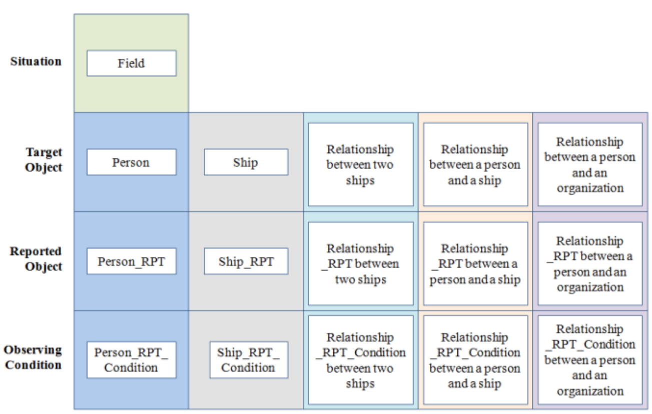


Fig. 5 shows a set of MFrags in the extended PROGNOS PO. The list on the left indicates the four MFrag groups. Each group is decomposed into sub-groups. For example, the target object group contains five sets of MFrags (Person MFrags, Ship MFrags, MFrags for the relationship between two ships, MFrags for the relationship between a person and a ship, and MFrags for the relationship between a person and an organization). The following list (PO 2) shows part of new MFrags added into the extended PROGNOS PO.

PO 2: Part of New MFrags added into the original PROGNOS probabilistic ontology

```
[F: Orgainzation_Report_MFrag
        [C: isA(sr,SENSOR), isA(tr ,ORGANIZATION), isA(rt,REPORTEDTARGET)]
        [C: SensorOf(sr, tr), tr = ReportedTarget(rt)]
        [R: isTerroristOrganizationRPT(rt)
                [IP: isTerroristOrganization(tr)]
                [IP: performance(sr, tr)]
        ]
]
[F: Situation_MFrag
        [C: isA(ship,SHIP), isA(person,PERSON), isA(field,FIELD)]
        [C: field = Location(ship)]
        [C: hasCrewMember(ship, person)]
        [R: PotentialTerroristAttacks(field) [IP: isShipOfInterest(ship), isTerroristPerson(person)]]
]
...
```

In PO 2, we added the ship report MFrag which can be used to reason about Rule 1. Also, we added the situation MFrag which can be used to reason about Rule 2.

In the *Learn Reasoning Model* step, the extended PROGNOS PO can be refined using a MEBN learning algorithm. The goal of MEBN learning is to learn an MTheory from a training dataset. A basic MEBN learning method for relational datasets was suggested [9][10]. This approach assumes that the training dataset is stored in a relational database based on RM. MEBN learning searches parameters, variables, and structures to find an MTheory that provides a good fit to the training dataset. In our case, the structures are given by the above steps as suggested in the PSAW reference model. Therefore, only parameter learning is required. The goal of parameter learning is to estimate the parameters θ^* of a class local distribution L given a training dataset $\boldsymbol{D}$ and the type of distribution being learned, which fit well the training dataset $\boldsymbol{D}$.

For a discrete random variable case, Dirichlet distribution is commonly used because it is conjugate to the multinomial distribution. With a Dirichlet prior distribution, the posterior predictive distribution has a simple form [21][22]. For continuous random variables, multiple regression can be used. Park et al [9] introduced a basic MEBN parameter learning and structure learning for a conditional Gaussian hybrid model in which no discrete random variable may have a continuous parent random variable.

For example, parameters for a conditional Gaussian distribution can be estimated using multiple regression. The following class local distribution (CLD) is an illustrative example of a conditional linear Gaussian CLD for the node *Speed_RPT*(*rt*, *tr*), which means a speed report *rt* for a target *tr*. The CLD of the node is a continuous CLD with hybrid parents (*Sensor_Condition* and *Speed*). In this case, we assume that the discrete parent node *Sensor_Condition*(*sr*, *tr*), which means a condition of a sensor *sr* for a target *tr*, has two states (*Good* and *Bad*) and the node *Speed* (*tr*), which means an actual speed of a target *tr*, is continuous.

CLD 1 [Conditional Linear Gaussian]: *Speed_RPT*(*rt*, *tr*)

1 **if** some sr.tr **have** (*Sensor_Condition* = Good) [
2 $\Theta_{1.0} + \Theta_{1.1}*Speed$ + NormalDist(0, $\Theta_{1.2}$)
3] **else** [
4 $\Theta_{2.0} + \Theta_{2.1}*Speed$ + NormalDist(0, $\Theta_{2.2}$)
5]

Parameter learning for this CLD estimates the parameters ($\Theta_{1.0}$, $\Theta_{1.1}$, and $\Theta_{1.2}$) in Line 2 and the parameters ($\Theta_{2.0}$, $\Theta_{2.1}$, and $\Theta_{2.2}$) in Line 4 using multiple regression.

D. Test Reasoning Model

This step performs two sub-steps (*Experiment Reasoning Model* and *Evaluate Experimental Results*) to evaluate the extended PROGNOS PO from the test dataset. In the *Experiment Reasoning Model* step, the performance of estimation and prediction for the extended PROGNOS PO can be assessed using a performance measure (e.g., the mean squared error or the Brier score). Each experiment consists of the following five steps. (1) The test dataset provides entity information (e.g., ship1, person1, and field1) and ground truth information (e.g., isShipOfInterest_ship1 = true, isTerroristPerson_person1 = true) to the extended PROGNOS PO. (2) Given these, the extended PROGNOS PO is used to compute a marginal probability distribution (e.g., P(PotentialTerroristAttacks_field1 | isShipOfInterest_ship1 = true, isTerroristPerson_person1 = true) in response to a query. (3) The test dataset provides ground truth data (e.g., PotentialTerroristAttacks_field1 = High). (4) Steps 1-3 are repeated for all test cases. (5) Finally, for results for all cases, the measures are calculated.

In the *Evaluate Experimental Results* step, we evaluate the measures using the performance criteria in the requirements defined in the *Analyze Requirement* step (e.g., a mean squared error < 0.1). If the evaluation is not satisfied (e.g., a mean squared error >= 0.1), we can return to the previous steps to improve the performance of the extended PROGNOS PO. We can investigate the extended PROGNOS PO in the *Construct Reasoning Model* step. Unsatisfactory performance can be caused by a training database of insufficient size. In this case, we may find more datasets and apply them to the learning process. Also, it is possible that the MEBN learning algorithm which we use is ineffective. In this case, the application of a more effective MEBN learning algorithm is required. The world model in the *Construct Reasoning Model* step can be incorrect. For this, we may need to conduct a further field investigation and research to develop a more accurate world model. The requirements in the *Analyze Requirements* step can be impracticable or requires a too high standard to address it. In this case, readjustments for the requirements can be performed by the stakeholders.

V. COMPARING UMP-ST AND HMLP

HMLP is a modification of UMP-ST that specifies some detailed sub-steps and uses two reference models (the reference MEBN model for PSAW and MEBN-RM). These reference models can support efficient modeling for a probabilistic ontology in PSAW. The first steps (*Requirement*) for both processes are same. In the case of HMLP, the reference MEBN model for PSAW provides some guidance on groups of entities to be defined (i.e., Reported Object, Observing Condition, Target Object, and Situation). In the second step of HMLP, the reference model also supports developing a world model in terms of PSAW by providing candidate entities (i.e., *T*, *OR*, *SR*, *TR*, and *RT*), attributes, and relationships. In the third step of HMLP, MEBN-RM supports the development of entities, random variables, and MFrags from a relational schema. HMLP also makes use of MEBN learning algorithms, so given a training dataset, a probabilistic ontology can be efficiently constructed. The second and third steps are mainly different with UMP-ST. These steps in HMLP can accelerate the modeling for probabilistic ontologies in PSAW and produce more comprehensive models.

Table 1 shows feature comparison between the original PROGNOS PO and the extended PROGNOS PO. Each number in the table means the number of the features (entities, random variables, relationships between random variables, and MFrags). For example, the number of entities in the original model is three (*Ship*, *Person*, and *Organization*), while the number of entities in the extended model is ten (*Field*, *Ship*, *Person*, *Organization*, *ShipSensor*, *PersonSensor*, *OrganizationSensor*, *ReportedShip*, *ReportedPerson*, and *ReportedOrganization*). Table 1 shows that the feature of the extended PROGNOS PO is more comprehensive than the feature of the original PROGNOS PO. The original PROGNOS PO contains 51 RVs, while the extended PROGNOS PO contains 115 RVs. This means that the extended PROGNOS PO can answer more various questions. For example, a reasoning about potential terrorist attacks in a field can be performed using the extended PROGNOS PO, but the original PROGNOS PO can't. Also, the extended PROGNOS PO contains observing conditions for sensors, so this may enable us to perform more accurate reasoning.

TABLE 1. Comparison between the original PROGNOS probabilistic ontology and the extended PROGNOS probabilistic ontology

	Entities	Random Variables	Relationships	MFrags
Original	3	51	53	18
Extended	10	116	147	36

If we assume that there is a training dataset for MEBN learning, the development period for the PROGNOS PO can be reduced. Usually, to develop an RV and its parameter, we study literature related to the RV and find possible parameters for the RV. Another way for the development of such an RV is to use domain expert knowledge. A subject matter expert (SME) may provide values and parameters for the RV, and relationships between RVs. In the PROGNOS project, to develop one RV, we used the following steps: (1) an SME in the maritime domain explained domain knowledge to an RV developer, (2) the RV developer developed the RV using the MEBN/PR-OWL software [27], and (3) the RV in the MEBN/PR-OWL software was evaluated by the SME. These steps were implemented with at least one day per RV. If we assume that for each RV, one day may be required to develop it by one RV developer and one SME, then the original PROGNOS PO requires around 51 days. On the contrary, if we assume that all datasets are available, the development with MEBN learning may require around one day for setting the datasets and learning a PO using a MEBN learning algorithm.

VI. CONCLUSION

UMP-ST was applied for construction of a probabilistic ontology to support PROGNOS including the PROGNOS PO. The PROGNOS PO played an important role in the operation of PROGNOS. However, manually developing and maintaining a probabilistic ontology is a labor-intensive and insufficiently agile process. Therefore, HMLP containing the reference models and machine learning methods was introduced. In the previous work for PROGNOS, UMP-ST was applied to develop the PROGNOS PO. This paper applied HMLP to develop the extended PROGNOS PO which was more comprehensive than the original model and was developed more quickly.

The following summarizes future research issues. (1) HMLP in this research was not fully applied with MEBN learning from a training dataset. Evaluation of effectiveness (i.e., reasoning accuracy) of reasoning models learned from

MEBN learning is required. (2) A probabilistic ontology can contain MFrags, context nodes, resident (or inputs) nodes, graphs, FOL formula for nodes, and class local distributions for nodes. These elements can be subject to MEBN learning. Especially, FOL formula learning in a probabilistic ontology is a difficult topic relative to the others. In our approach, a dataset for learning is given from a relational database. Because we rely on MEBN-RM, we do not need to perform the complicated task of FOL formula learning from text data. FOL formula learning in a probabilistic ontology can be supported by Inductive Logic Programming [23][24] and Statistical Natural Language Processing [25]. (3) Also, future steps for the extended PROGNOS PO are to apply it to a realistic reasoning system for Maritime Domain Awareness.

ACKNOWLEDGMENTS

We appreciate Dr. K. C. Chang, Dr. W. Sun, Dr. R. Carvalho, Dr. R. Haberlin, Mr. S. Matsumoto, and Mr. A. Mugali for their efforts in the previous PROGNOS research. The research was partially supported by the Office of Naval Research (ONR), under Contract#: N00173-09-C-4008.

REFERENCES

[1] Laskey, K. B. (2008). MEBN: A Language for First-Order Bayesian Knowledge Bases. Artificial Intelligence, 172(2-3).

[2] Laskey, K. B., D'Ambrosio, B., Levitt, T. S., & Mahoney, S. M. (2000). Limited Rationality in Action: Decision Support for Military Situation Assessment. Minds and Machines, 10(1), 53-77.

[3] Wright, E., Mahoney, S. M., Laskey, K. B., Takikawa, M. & Levitt, T. (2002). Multi-Entity Bayesian NetworksforSituation Assessment. Proceedings of the Fifth International Conference on Information Fusion.

[4] Costa, P. C. G., Laskey, K. B., Takikawa, M., Pool, M., Fung, F., & Wright, E. J. (2005). MEBN Logic: A Key Enabler for Network Centric Warfare. Proceedings of the 10th ICCRTS.

[5] Suzic, R. (2005). A generic model of tactical plan recognition for threat assessment. In Defense and Security (pp. 105-116). International Society for Optics and Photonics.

[6] Costa, P. C. G., Laskey, K. B., & Chang, K. C. (2009). PROGNOS: Applying Probabilistic Ontologies To Distributed Predictive Situation Assessment In Naval Operations. Proceedings of the 14th ICCRTS.

[7] Carvalho, R. N., Costa, P. C. G., Laskey, K. B., & Chang, K. C. (2010). PROGNOS: predictive situational awareness with probabilistic ontologies. In Proceedings of the 13th International Conference on Information Fusion.

[8] Park, C. Y., Laskey, K. B., Costa, P. C. G., & Matsumoto, S. (2014). Predictive Situation Awareness Reference Model using Multi-Entity Bayesian Networks. Proceedings of the 17th International Conference on Information Fusion.

[9] Park, C. Y., Laskey, K. B., Costa, P. C. G., & Matsumoto, S. (2013). Multi-Entity Bayesian Networks Learning For Hybrid Variables In Situation Awareness. Proceedings of the 16th International Conference on Information Fusion (Fusion 2013).

[10] Park, C. Y., Laskey, K. B., Costa, P. C. G., & Matsumoto, S. (2013). Multi-Entity Bayesian Networks Learning In Predictive Situation Awareness. Proceedings of the 18th International Command and Control Technology and Research Symposium (ICCRTS 2013).

[11] Codd, E. F. (1970). A Relational Model of Data for Large Shared Data Banks. Communications of the ACM.

[12] Codd, E. F. (1969). Derivability, Redundancy, and Consistency of Relations Stored in Large Data Banks. IBM Research Report.

[13] Carvalho, R. N., Laskey, K. B., & Da Costa, P. C. (2016). Uncertainty modeling process for semantic technology (No. e2045v1). PeerJ Preprints.

[14] Carvalho, R. N. (2011). Probabilistic Ontology: Representation and Modeling Methodology. PhD Dissertation. George Mason University.

[15] Costa, Paulo. C. G., Laskey, Kathryn B., Chang, Kuo-Chu, Sun, Wei, Park, Cheol Y., & Matsumoto, Shou. (2012). High-Level Information Fusion with Bayesian Semantics. Proceedings of the Nineth Bayesian Modelling Applications Workshop.

[16] Sageman, M. (2004). Understanding terror networks. University of Pennsylvania Press.

[17] Kokar, M. M., Matheus, C. J., & Baclawski, K. (2009). Ontology-based situation awareness. Information fusion, 10(1), 83-98.

[18] Smith, B. (2003). Ontology. Retrieved September 2, 2016from http://ontology.buffalo.edu/smith/articles/ontology_pic.pdf.

[19] Paulo C. G Costa, Bayesian Semantics for the Semantic Web. PhD Dissertation, George Mason University, July 2005. Brazilian Air Force.

[20] Park, C. Y., Laskey, K. B., Costa, P. C., & Matsumoto, S. (2016). A process for human-aided Multi-Entity Bayesian Networks learning in Predictive Situation Awareness. In 19th International Conference on Information Fusion (FUSION).

[21] Heckerman, D., Geiger, D., & Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. Machine Learning, 20:197–243.

[22] Koller, D., & Friedman, N. (2009). Probabilistic Graphical Models: Principles and Techniques. The MIT Press, 1 edition.

[23] Lavrac, N. & Dzeroski, S. (1994). Inductive Logic Programming: Techniques and Applications. Ellis Horwood, New York.

[24] Muggleton, S., & De Raedt, L. (1994). Inductive logic programming: Theory and methods. The Journal of Logic Programming, 19, 629-679.

[25] Manning, C. D., & Schütze, H. (1999). Foundations of statistical natural language processing (Vol. 999). Cambridge: MIT press.

[26] Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. Monthly weather review, 78(1), 1-3.

[27] Costa, P., Ladeira, M., Carvalho, R. N., Laskey, K., Santos, L., & Matsumoto,S. (2008). A first-order Bayesian tool for probabilistic ontologies. in Proceedings of the Twenty-First International Florida Artificial Intelligence Research Society Conference (FLAIRS 2008), (Coconut Grove, FL,USA), pp. 631–636, AAAI Press.