Chapter 5: Uncertainty Modeling Process for Semantic Technologies (UMP-ST)

As explained in Chapter 1, probabilistic ontologies can be used to represent experts' knowledge in an automated system in order to overcome the information overload problem. However, one major problem is that probabilistic ontologies are complex and hard to model. It is challenging enough to design models that use only logic or only uncertainty; combining the two poses an even greater challenge. In fact, in the past few years I have received a number of e-mails from researchers all around the world asking for some information and/or literature on how to build probabilistic ontologies. The problem is that there is no methodology in the literature related to probabilistic ontology engineering.

Although there is now substantial literature about what PR-OWL is [27,29,31], how to implement it [23,20,19,26], and where it can be used [30,32,33,77,79,80], little has been written about how to model a probabilistic ontology.

This lack of methodology is not only associated with PR-OWL. Other languages that use probabilistic methods for representing uncertainty on the SW have been advancing in areas like inference [12,122], learning [36,86], and applications [14,120,87,13,41]. Examples of such languages include OntoBayes [136], BayesOWL [37], and probabilistic extensions of SHIF(**D**) and SHOIN(**D**) [85], and Markov Logic Networks (MLN). Despite this proliferation of languages and methods, little has been written about how to build such models.

Therefore, in this Chapter I will describe an approach for modeling a probabilistic ontology and using it for plausible reasoning in applications that use Semantic Technologies.

The Uncertainty Reasoning Process for Semantic Technologies (URP-ST)[1] presented

[1] In [22] I present this process as the modeling process. However, this is actually more than just modeling. This process represents the sequence of phases necessary in order to achieve the capability of plausible reasoning with semantic technologies. Therefore, I have changed the name of this process to Uncertainty Reasoning Process for Semantic Technologies (URP-ST).

in Figure 5.1 is divided into three steps: First we have to model the domain (T-Box[2]), then we need to populate the model with data (A-Box[3]), and finally we can use both the model (T-Box) and the data available (A-Box), *i.e.*, the KB, for reasoning. In other words, in order to be able to reason with uncertainty, first we need a model, which describes how the different concepts in our ontology interact under uncertainty by knowing which evidence supports which hypothesis, etc. Once there is a model available, it needs to be populated with the data available before it is able to do any reasoning. Finally, with the model and with the data available, it is possible to present the inference engine with queries for that domain, like `isA(person1, Terrorist)`. Notice that unlike standard ontology reasoning systems that return true only if that person is known to be a terrorist for sure, the probabilistic ontology reasoning system will return the likelihood that the person is a terrorist, for instance $P($`isA(person1, Terrorist) = true`$) = 75\%$.

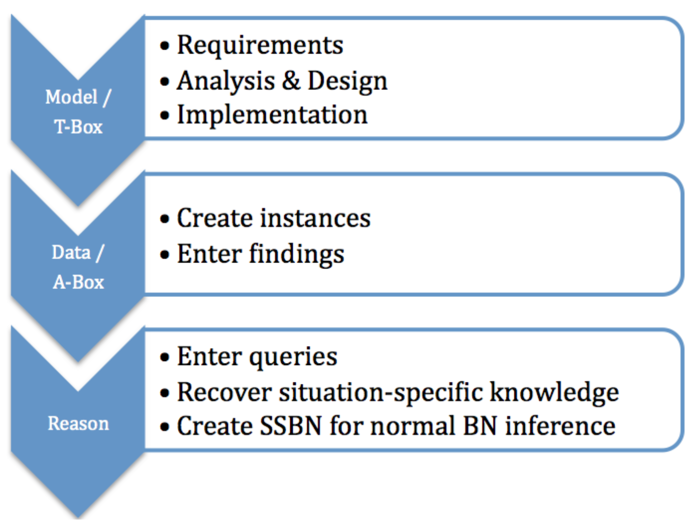


Figure 5.1: Uncertainty Reasoning Process for ST (URP-ST).

---

[2]T-Box statements describe the part of the KB that defines terms of a controlled vocabulary, for example, a set of classes and properties

[3]A-Box are statements about the vocabulary defined by the T-Box, for example, instances of classes. T-Box and A-Box together form the KB.

Now I focus in detail on the modeling phase of the URP-ST. I call this phase the Uncertainty Modeling Process for Semantic Technologies (UMP-ST). The UMP-ST consists of four major disciplines: Requirements, Analysis & Design, Implementation, and Test. These terms are borrowed from the Unified Process (UP)[4] [68] with some modifications to reflect our domain of ontology modeling instead of software development process. The methodology described here is also consistent with the Bayesian network modeling methodology described by [72] and [81].

Figure 5.2 depicts the intensity of each discipline during the UMP-ST[5]. Like the UP, UMP-ST is iterative and incremental. The basic idea behind iterative enhancement is to model our domain incrementally, allowing the modeler to take advantage of what was being learned during the modeling of earlier, incremental, deliverable versions of the model. Learning comes from discovering new rules, entities, and relations that were not obvious previously, which can give rise to new questions and evidence that might help us achieve our previously defined goal as well as give rise to new goals. Some times it is possible to test some of the rules defined during the Analysis & Design stage even before having implemented it. This is usually done by creating simple probabilistic models to evaluate whether the model will behave as expected before creating the more complex first-order logic probabilistic models. That is why in the first iteration (I1) of the Inception phase we have some testing happening before the implementation started.

[4] Although the most common instance of UP is the Rational Unified Process (RUP) [74], there are alternatives, like the Open Unified Process (OpenUP) [9].

[5] In [22] I present this methodology as UMP for the Semantic Web. However, this methodology is not restricted to the SW. Any application that uses semantic technologies can benefit from it, even if it is not designed to be used on the Web. Therefore, I decided to change the name to UMP for Semantic Technologies.

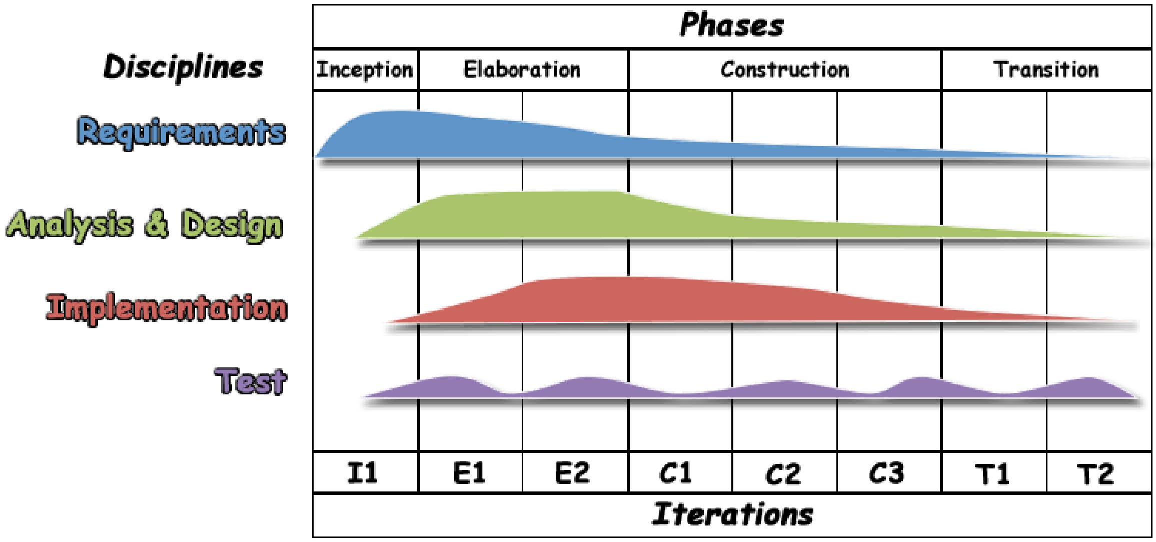


Figure 5.2: Uncertainty Modeling Process for Semantic Technologies (UMP-ST).

Figure 5.3 presents the Probabilistic Ontology Modeling Cycle (POMC). This cycle depicts the major activities or concepts in each discipline, how they usually interact, and the natural order in which they occur. However, as described previously, this is not the same as the waterfall model (see [114] for information about the waterfall model). *I.e.*, it is not necessary to go through implementation to be able to test the model. Besides that, the interactions between the disciplines are not restricted to the arrows presented. In fact, it is possible to have interactions between any pair of disciplines. For instance, it is not uncommon to discover a problem in the rules defined in the Analysis & Design discipline during the activities in the Test discipline. In other words, although, the arrow just shows interaction between the Test and Requirement disciplines, it is possible to go directly from Test to Analysis & Design.

Figure 5.3: Probabilistic Ontology Modeling Cycle (POMC) - Requirements (Goals), Analysis & Design (Entities, Rules, and Group), Implementation (Mapping and LPD), and Test (Evaluation).

In Figure 5.3 the Requirements discipline (Goals circle in blue) defines the goals that must be achieved by reasoning with the semantics provided by our model. The Analysis & Design discipline describes classes of entities, their attributes, how they relate, and what rules apply to them in our domain (Entities, Rules, and Group circles in green). This definition is independent of the language used to implement the model. The Implementation discipline maps our design to a specific language that allows uncertainty in ST, which in this case is PR-OWL (Mapping and LPD circles in red). Finally, the Test discipline is responsible for evaluating if the model developed during the Implementation discipline is behaving as expected from the rules defined during Analysis & Design and if they achieve the goals elicited during the Requirements discipline (Evaluation circle in purple). As explained before, it is possible to test some rules and assumptions even before the implementation. This is a crucial step to mitigate risk by identifying problems before wasting time in developing an inappropriate complex model.

The following sections illustrate the UMP-ST process and the POMC cycle through a case study in procurement fraud detection and prevention and a case study in maritime domain awareness. The URP-ST is also demonstrated by the use of UnBBayes to implement the model, to populate the KB, and to perform plausible reasoning.

On the one hand, Section 5.1 will focus on presenting in detail the activities that must be executed in each discipline in the POMC cycle. On the other hand, Section 5.2 will focus on presenting how the model evolves through time with every new iteration.

The objective of the first is to present as much detail as possible on the steps necessary to model a probabilistic ontology using the POMC cycle. The objective of the second is to show that the UMP-ST process provides a useful approach for allowing the natural evolution of the model through different iterations.

5.1 Probabilistic Ontology for Procurement Fraud Detection and Prevention in Brazil

A major source of corruption is the procurement process. Although laws attempt to ensure a competitive and fair process, perpetrators find ways to turn the process to their advantage while appearing to be legitimate. This is why a specialist has didactically structured the different kinds of procurement frauds the Brazilian Office of the Comptroller General (CGU) has dealt with in past years.

These different fraud types are characterized by criteria, such as business owners who work as a front for the company, use of accounting indices that are not common practice, etc. Indicators have been established to help identify cases of each of these fraud types. For instance, one principle that must be followed in public procurement is that of competition. Every public procurement should establish minimum requisites necessary to guarantee the execution of the contract in order to maximize the number of participating bidders. Nevertheless, it is common to have a fake competition when different bidders are, in fact, owned by the same person. This is usually done by having someone as a front for the enterprise, which is often someone with little or no education.

The ultimate goal of this case study is to structure the specialist knowledge in a way that an automated system can reason with the evidence in a manner similar to the specialist. Such an automated system is intended to support specialists and to help train new specialists, but not to replace them. Initially, a few simple criteria were selected as a proof of concept. Nevertheless, it is shown that the model can be incrementally updated to incorporate new criteria. In this process, it becomes clear that a number of different sources must be consulted to come up with the necessary indicators to create new and useful knowledge for decision makers about the procurements.

Figure 5.4 presents an overview of the procurement fraud detection process. The data for our case study represent several requests for proposal and auctions that are issued by the Federal, State and Municipal Offices (Public Notices - Data). The idea is that the analysts who work at CGU, already making audits and inspections, accomplish the collection of information through questionnaires that can specifically be created for the collecting of indicators for the selected criteria (Information Gathering). These questionnaires can be created using a system that is already in production at CGU. Once they are answered the necessary information is going to be available (DB - Information). Hence, UnBBayes, using the probabilistic ontology designed by experts (Design - UnBBayes), will be able to collect these millions of items of information and transform them into dozens or hundreds of items of knowledge. This will be achieved through logic and probabilistic inference. For instance, procurement announcements, contracts, reports, etc. - a huge amount of data - are analyzed allowing the gathering of relevant relations and properties - a large amount of information. Then, these relevant relations and properties are used to draw some conclusions about possible irregularities - a smaller number of items of knowledge (Inference - Knowledge). This knowledge can be filtered so that only the procurements that show a probability higher than a threshold, *e.g.* 20%, are automatically forwarded to the responsible department along with the inferences about potential fraud and the supporting evidence (Report for Decision Makers).

Figure 5.4: Procurement fraud detection overview.

5.1.1 Requirements

The objective of the requirements discipline is to define the objectives that must be achieved by creating a computable representation of domain semantics and reasoning with it. For this discipline, it is important to define the questions that the model is expected to answer (*i.e.*, the queries to be posed to the system being designed). For each question, a set of information that might help answer the question (evidence) must be defined.

There are basically two types of requirements: functional and non functional [134, 124]. The requirements just described above are called functional requirements. Functional requirements are statements related to what the system should provide, what features it should have, how it should behave, etc. In our case, functional requirements relate to the goals, queries, and evidence that pertain to our domain of reasoning. Non functional requirements on the other hand represent constraints on the system as a whole. For instance, in our use case a non functional requirement could be that the query has to be answered in less than a minute. Another example is that the posterior probability given as an answer to a given query has to be either exact or an approximation with an error bound of .5%.

Since it is easier and more straightforward to define non functional requirements, which define time constraints, error bounds, etc., we will focus on describing how to come up with the functional requirements in our use case.

In order to understand the requirements for the procurement fraud detection and prevention model, we first have to explain some of the problems encountered when dealing with public procurements.

One of the principles established by the Law N 8,666/93 is equality among the bidders. This principle prohibits the procurement agent from discriminating among potential suppliers. However, if the procurement agent is related to the bidder, he/she might feed information or define new requirements for the procurement in a way that favors the bidder.

Another principle that must be followed in public procurement is that of competition. Every public procurement should establish minimum requisites necessary to guarantee the execution of the contract in order to maximize the number of participating bidders. Nevertheless, it is common to have a fake competition when different bidders are, in fact, owned by the same person. This is usually done by having someone as a front for the enterprise, which is often someone with little or no education. Another common tactic is to set up front enterprises owned by relatives of the owner of the enterprise committing fraud.

According to [98] participating in a public procurement can be very expensive and time consuming. Thus, some firms are unwilling to take part in a process that is not guaranteed to achieve favorable results. Since this diminishes the number of enterprises participating in the procurement, collusion among the bidders is more likely to happen. What happens in Brazil is that a small group of firms regularly participate in procurements of certain goods and services. When this happens, the competitors in a public procurement take turns winning the contracts. They stipulate the winning bid, and all other firms bid above that price. There is no competition, and the government pays a higher price for the contract. Although collusion is not an easy thing to prove, it is reasonable to assume that collusion is enabled by some kind of relationship between the enterprises.

All firms in Brazil have a registration number, called CGC, which stands for General List of Contributors. When a firm is suspended from procuring with the public administration, its CGC number is used to inform all other public agencies that this firm should not participate in public procurements. However, the firm can simply close its business and open a new one using a different CGC. Thus the firm that should not be able to participate in public procurements is now allowed, since it now has a different number associated to it. Unfortunately, the Commercial Code permits this change of CGC number.

One other problem is that public procurement is quite complex and may involve large sums of money. Therefore, the members that form the committee of the procurement must not only be prepared, but also have a clean history (no criminal nor administrative conviction) in order to maximize morality, one of the ethical principles that federal, state, municipal and district government should all adopt.

Having explained that, in our fraud detection and prevention in the procurements domain we have the following set of goals/queries/evidences:

1. Identify whether a given procurement should be inspected and/or audited (*i.e.* evidence suggests further analysis is needed);

   (a) Is there any relation between the committee and the enterprises that participated in the procurement?

i. Look for member and responsible person of an enterprise who are related (mother, father, brother, or sister);

ii. Look for member and responsible person of an enterprise who live at the same address.

(b) Is the responsible person of the winner enterprise of the procurement a front?

i. Look for value of the contract related to this procurement;

ii. Look for his/her education degree;

iii. Look for his/her annual income.

(c) Was the responsible person of the winner enterprise of the procurement responsible for an enterprise that has been suspended from procuring with the public administration?

i. Look for this information in the General List of Contributors (CGC) database.

(d) Was competition compromised?

i. Look for bidders who are related (mother, father, brother, or sister).

2. Identify whether the committee of a given procurement should be changed.

(a) Is there any member of committee who does not have a clean history?

i. Look for criminal history;

ii. Look for administrative investigations.

(b) Is there any relation between members of the committee and the enterprises that participated in previous procurements?

i. Look for member and responsible person of an enterprise who are relatives (mother, father, brother, or sister);

ii. Look for member and responsible person of an enterprise who live at the same address.

Another important aspect of the Requirements discipline is defining traceability of requirements. Gotel and Finkelstein [51] define requirements traceability as:

> Requirements traceability refers to the ability to describe and follow the life of a requirement, in both forwards and backwards direction.

A common tool for defining requirements traceability is the specification tree, which is the arrangement of requirements in such a way that each requirement is linked to its "parent" requirement in the higher specification. This is exactly the way we have defined the requirements for our procurement model. Every evidence is linked to its higher level query, which is linked to its higher level goal. Here we are not only defining the requirements, but also defining their traceability.

However, requirements traceability (RT) is not only about defining links between requirements. In fact, RT also provides the link between work products of other disciplines, like the rules in the Analysis & Design and MFrags in the Implementation, and the goals, queries, and evidence elicited in the Requirements discipline. This kind of link makes RT specially useful for validation and management of change.

This kind of link between work products of different disciplines is typically done via a Requirements Traceability Matrix (RTM) [134, 124]. Table 5.1 presents a RTM with the traceability between the requirements defined in this Section for the fraud detection model. Notice that this matrix represents exactly the same thing as the specification tree defined previously. However, when mapping the work product of other disciplines to the requirements, in most cases, it will not be possible to use a specification tree, but it will always be possible to use RTM.

5.1.2 Analysis & Design

Once we have defined our goals and described how to achieve them, it is time to start modeling the entities, their attributes, relationships, and rules to make that happen. This is the purpose of the Analysis & Design discipline.

Table 5.1: Requirements Traceability Matrix for the requirements of the fraud detection model.

ID	1	1a	1ai	1aii	1b	1bi	1bii	1biii	1c	1ci	1d	1di	2	2a	2ai	2aii	2b	2bi	2bii
1	X																		
1a	X	X																	
1ai	X	X	X																
1aii	X	X		X															
1b	X				X														
1bi	X				X	X													
1bii	X				X		X												
1biii	X				X			X											
1c	X								X										
1ci	X								X	X									
1d	X										X								
1di	X										X	X							
2													X						
2a													X	X					
2ai													X	X	X				
2aii													X	X		X			
2b													X				X		
2bi													X				X	X	
2bii													X				X		X

The major objective of this discipline is to define the semantics of our model. In fact, most of our semantics can be defined in normal ontologies, including the deterministic rules that the concepts described in our model must obey. Since there are whole books describing how to design such ontologies, and our main concern is on the uncertain part of the ontology, we will not cover these methods in this Section. For more information see [7, 50, 101, 102].

Nevertheless, we do need a starting point in order to design our probabilistic ontology. As a matter of fact, one good way to start modeling these properties is to use UML as described in Section 2.1. However, as we have seen, UML does not support complex rule definitions. So we will just document them separately to remind us of the rules that must be described when implementing our model in PR-OWL.

Figure 5.5 depicts a simplified design of our domain requirements. A `Person` has a `name`, a `mother` and a `father` (also `Person`). Every `Person` has a unique identification that in Brazil is called `CPF`. A `Person` also has an `Education` and `livesAt` a certain `Address`. In addition, everyone is obliged to file his/her `TaxInfo` every year, including his/her `annual-Income`. These entities can be grouped as `Personal Information`. A `PublicServant` is a `Person` who `worksFor` a `PublicAgency`, which is a Government Agency. Every public `Procurement` is owed by a `PublicAgency`, has a committee formed by a group of `Public-Servants`, and has a group of `participants`, which are `Enterprises`. One of these will be the `winner` of the `Procurement`. Eventually, the `winner` of the `Procurement` will receive a `Contract` of some `value` with the `PublicAgency` `owner` of the `Procurement`. The entities just described can be grouped as `Procurement Information`. Every `Enterprise` has at least one `Person` that is `responsible` for its legal acts.

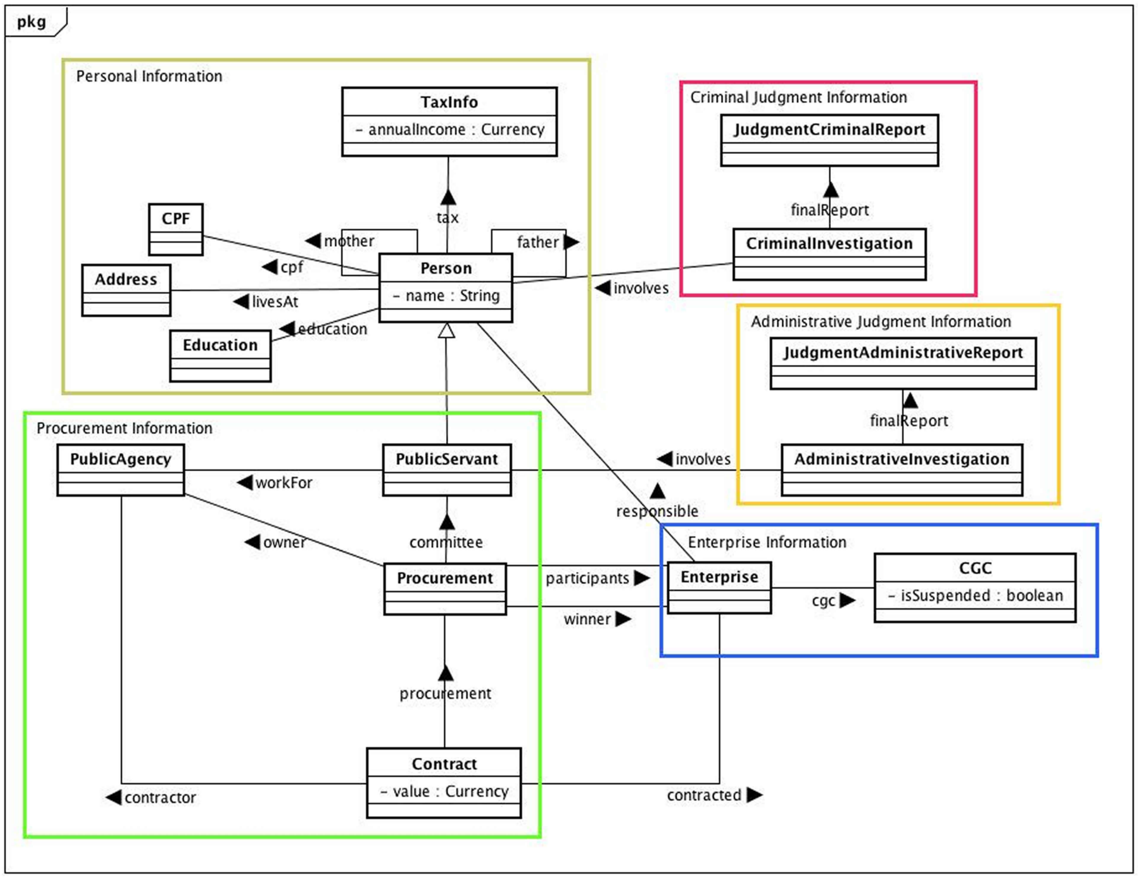


Figure 5.5: Entities, their attributes, and relations for the procurement model.

An `Enterprise` also has an identification number, the General List of Contributors `CGC`, which can be used to inform that this `Enterprise` is suspended from procuring with the public administration, `isSuspended`. These are grouped as the `Enterprise Information`. We also have `AdminstrativeInvestigation`, which has information about investigations that involves one or more `PublicServer`. Its `finalReport`, the `JudgmentAdministrativeReport`, contains information about the penalty applied, if any. These entities form the `Administrative Judgment Information`. Finally we have the `Criminal Judgment Information` group that describes the `CriminalInvestigation` that involves a `Person`, with its `finalReport`, the `JudgmentCriminalReport`, which has information about the verdict.

Table 5.2: Requirements Traceability Matrix for the rules of the fraud detection model.

ID	1	1a	1ai	1aii	1b	1bi	1bii	1biii	1c	1ci	1d	1di	2	2a	2ai	2aii	2b	2bi	2bii
1	X	X	X								X								
2	X	X		X							X								
3	X				X	X	X	X			X								
4	X								X	X									
5	X										X	X							
6	X	X	X	X	X	X	X	X	X	X	X	X							
7	X												X	X	X	X			
8	X																X	X	X
9	X												X	X	X	X	X	X	X

Besides the cardinality and uniqueness rules defined in the explanation above about the entities depicted in Figure 5.5, the probabilistic rules for our model include:

1. If a member of the committee has a relative (mother, father, brother, or sister) responsible for a bidder in the procurement, then it is more likely that a relationship exists between the committee and the enterprises, which inhibits competition.

2. If a member of the committee lives at the same address as a person responsible for a bidder in the procurement, then it is more likely that a relationship exists between the committee and the enterprises, which lowers competition.

3. If a contract of high value related to a procurement has a responsible person of the winner enterprise with low education or low annual income, then this person is likely to be a front for the firm, which lowers competition.

4. If the responsible person of the winner enterprise is also responsible for another enterprise that has its CGC suspended for procuring with the public administration, then this procurement is more likely to need further investigation.

5. If the responsible people for the bidders in the procurement are related to each other, then a competition is more likely to have been compromised.

6. If 1, 2, 3, or 5, then the procurement is more likely to require further investigation.

7. If a member of the committee has been convicted of a crime or has been penalized administratively, then he/she does not have a clean history. If he/she was recently investigated, then it is likely that he/she does not have a clean history.

8. If the relation defined in 1 and 2 is found in previous procurements, then it is more likely that there will be a relation between this committee and future bidders.

9. If 7 or 8, then it is more likely that the committee needs to be changed.

Once we have our rules defined, it is important to keep track of their traceability to the requirements. Although this is a step of the Requirements discipline, we will present it here.

In fact, when completing every discipline it is important to go back to the Requirements discipline to expand the RTM matrix.

Table 5.2 presents the traceability between the rules defined in the Analysis & Design stage and the goals, queries, and evidence defined in the Requirements stage. *I.e.*, this mapping defines which requirements the rules are realizing.

5.1.3 Implementation

Once we have finished our Analysis & Design, it is time to start implementing our model in a specific language. This Section describes how to model procurement fraud detection and prevention in PR-OWL using UnBBayes.

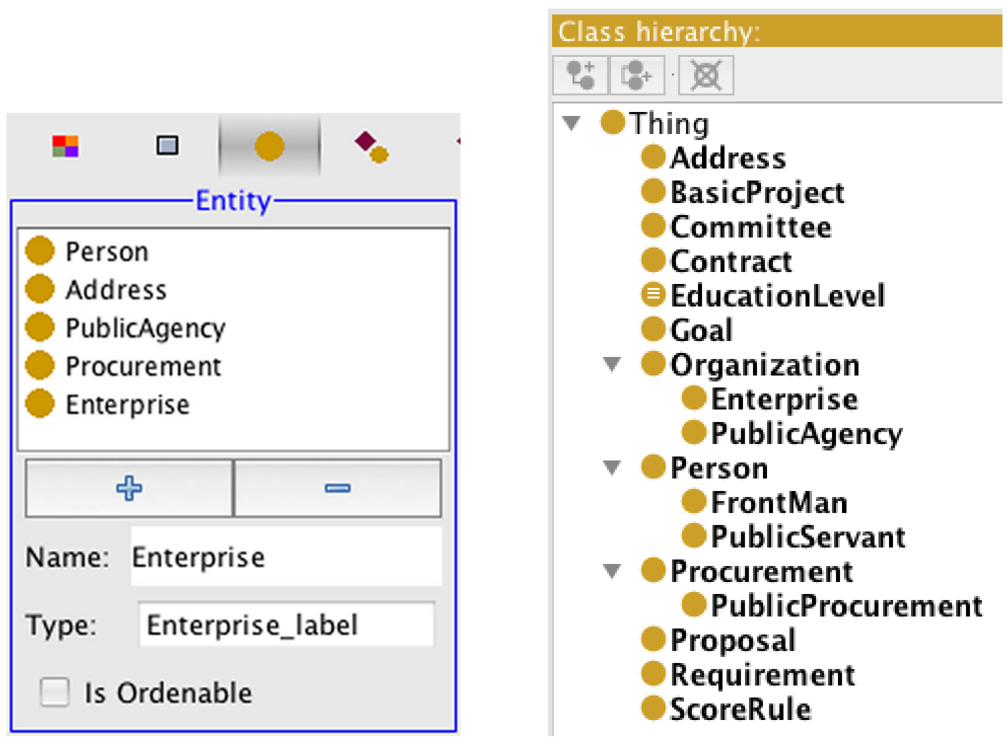
The first thing to do is to start mapping the entities, their attributes, and relations to PR-OWL, which uses essentially MEBN terms. This discipline is different from the previous ones, since it depends on the language/formalism being used. In this Section I will highlight the difference between implementing the fraud detection probabilistic ontology using PR-OWL 1 and PR-OWL 2.

PR-OWL 1, although with a few limitations, already has a mature implementation in UnBBayes (the first version was made publicly available in February 2008). PR-OWL 2 on the other hand is still under development [88] and the current working version has a lot of limitations and is just a proof-of-concept[6]. Therefore, the fraud detection probabilistic ontology will not be fully implemented in PR-OWL 2, but it will be implemented in PR-OWL 1. Nevertheless, once the final version of PR-OWL 2 is available it should be straightforward to migrate this PO to PR-OWL 2. Notice that the main objective of this Chapter is to describe the UMP-ST process and to highlight the differences between PR-OWL 1 and PR-OWL 2.

In PR-OWL 1, it is often a good idea to start mapping the entities. There is no need to map all entities in our model to an entity in PR-OWL. In fact, in our model we will make many simplifications. One of them is due to a limitation in UnBBayes current version, which

[6]PR-OWL 2 is being developed by the Group of Artificial Intelligence (GIA) at the University of Brasília, Brazil.

is the lack of support for a type hierarchy. Therefore, we will not have the `PublicServant` entity and we will assume that a `Person` might work for a `PublicAgency`. We will also assume that every `Person` and `Enterprise` in our KB is uniquely identified by its name, so we will not consider, in this simplified example, the `CPF` and `CGC` entities. Figure 5.6(a) presents the entities implemented in our PR-OWL ontology using UnBBayes. For more details about defining entities in UnBBayes see [20].



(a) Entities implemented in PR-OWL 1 using UnBBayes.

(b) Entities implemented in OWL for use in PR-OWl 2 using Protégé.

Figure 5.6: Entities for the procurement domain.

In PR-OWL 2, on the other hand, it is not necessary to map these entities. In fact, the entities are defined as classes in a regular ontology using OWL. Then PR-OWL 2 simply makes use of them. As previously explained, it is not the objective of the UMP-ST process to explain how to design standard deterministic ontologies. However, the Analysis & Design discipline helps with a starting point for defining this ontology. The class hierarchy presented in Figure 5.6(b) was derived from the UML diagram created during the Analysis & Design stage presented in Figure 5.5.

Once we have our entities defined, we consider characteristics that may be uncertain. Uncertainty is represented in MEBN by defining random variables (RVs). On the one hand, to define a RV in PR-OWL 1 using UnBBayes, we first define its home MFrag. Grouping the RVs into MFrags is done by examining the grouping created during Analysis & Design. On the other hand, in PR-OWL 2 RVs are independent of MFrag and are defined globally by defining its arguments, mapping to OWL, and default distributions.

Typically, a RV represents an attribute or a relation from our designed model in Analysis & Design. For instance, the RV `livesAt(person)` maps the relation `livesAt` in our designed model. As it is a functional relation, `livesAt` relates a `Person` to an `Address`. Hence, the possible values (or states) of this RV are instances of `Address`.

It is important to notice that although we followed the best practice of having the same domain and range on both OWL terms (*e.g.* `livesAt`) and PR-OWL 1 random variables (*e.g.* `livesAt(person)`), there is nothing in the language that guarantees these manual mappings will be kept the same throughout the life cycle of the model. Moreover, since there is no formal link between these terms, it is impossible for reasoners to identify that these terms are even linked. At best, it could only "guess" they are the same, since they have similar syntax (*e.g.* predicate `livesAt` has a similar name to the random variable `livesAt(person)`), which is, at best, contradictory for a language that is designed to convey semantics of terms and relations.

Chapter 4 described how PR-OWL 2 formalizes the mapping between RVs and OWL properties. In the proof-of-concept PR-OWL 2 plugin for UnBBayes, from now on called PR-OWL 2 plugin [88], a RV is automatically created and and its mapping automatically defined by dragging the OWL property and dropping it in the MFrag where it will be used as a resident node, as shown in Figure 5.7.

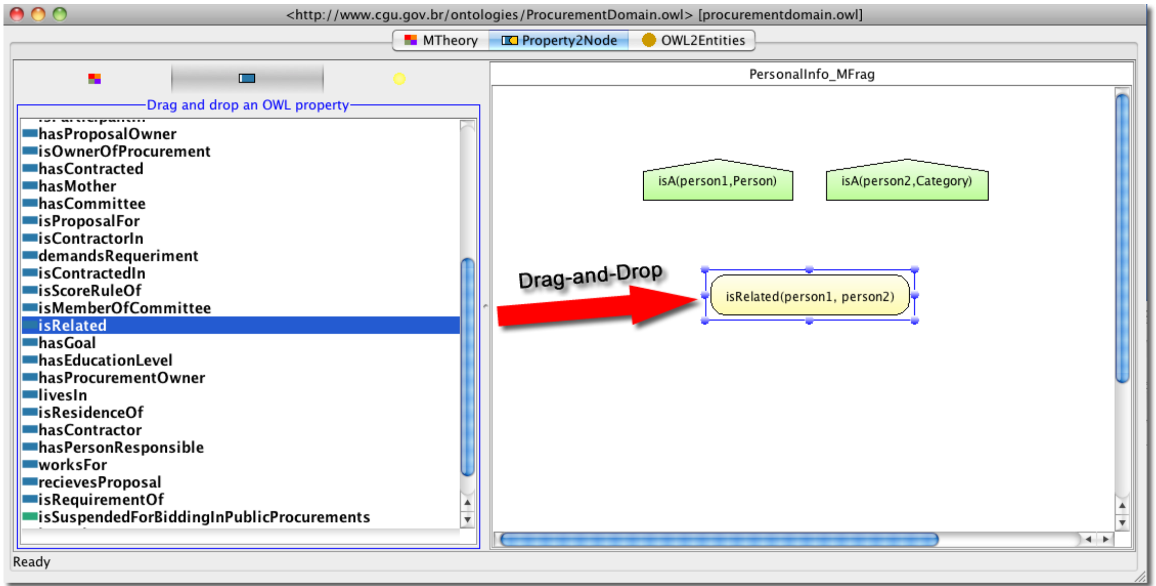


Figure 5.7: Creating a RV in PR-OWL 2 plugin from its OWL property by drag-and-drop.

We can also avoid explicitly representing some entities, by simply defining discrete outputs. In our implementation, we only need to know the education level of a `Person`, which is either `noEducation`, `middleSchool`, `highSchool`, `undergraduate`, or `graduate`. These are the states of the RV `hasEducationLevel(person)`, therefore, in PR-OWL 1, there is no need to define the entity `EducationLevel`, since no actual mapping will exist between the categorical RV and the OWL property `hasEducationLevel`. However, in PR-OWL 2, in order to represent categorical values, we would create a class `EducationLevel` with the `oneOf` construct from OWL. This construct allows us to define a set of predefined possible values for that class, which is exactly what we need.

Because the current version of UnBBayes-MEBN does not support continuous RVs, we must define a discretization for numerical attributes. For example, the attribute `value` of the `Contract` entity from our designed model is continuous, since it represents some `float` value in a specific `Currency`. However, we can discretize it by defining common intervals, as `lower than 10,000.00`, `between 10,000.01 and 100,000.00`, `between 100,000.01 and 500,000.00`, `between 500,000.01 and 1,000,000.00`, and `greater` `than 1,000,000.01`, which will be the states of the resident node `valueOf(procurement)`. This is the case for both implementations of PR-OWL 1 and PR-OWL 2 in UnBBayes. The difference is that in future versions of UnBBayes, which will support continuous RVs, PR-OWL 1 will not be able to use data types such as `float`, while PR-OWL 2 will, since the latter uses OWL's types instead of defining its own types as the former does.

Once all resident RVs are created, their relations can be defined by analyzing dependence between nodes. One good way to look for dependence is by looking at the rules defined in our model. For instance, rule 3 indicates that there is a dependence between `valueOf(procurement)`, `hasEducationLevel(person)`, and `isFront(person, enterprise)`.

The MFrags implemented in order to address all the rules defined in the Analysis & Design are:

1. Personal Information
2. Procurement Information
3. Enterprise Information
4. Front of Enterprise
5. Exists Front in Enterprise
6. Related Participant Enterprises
7. Member Related to Participant
8. Competition Compromised
9. Owns Suspended Enterprise
10. Judgement History
11. Related to Previous Participants
12. Suspicious Committee

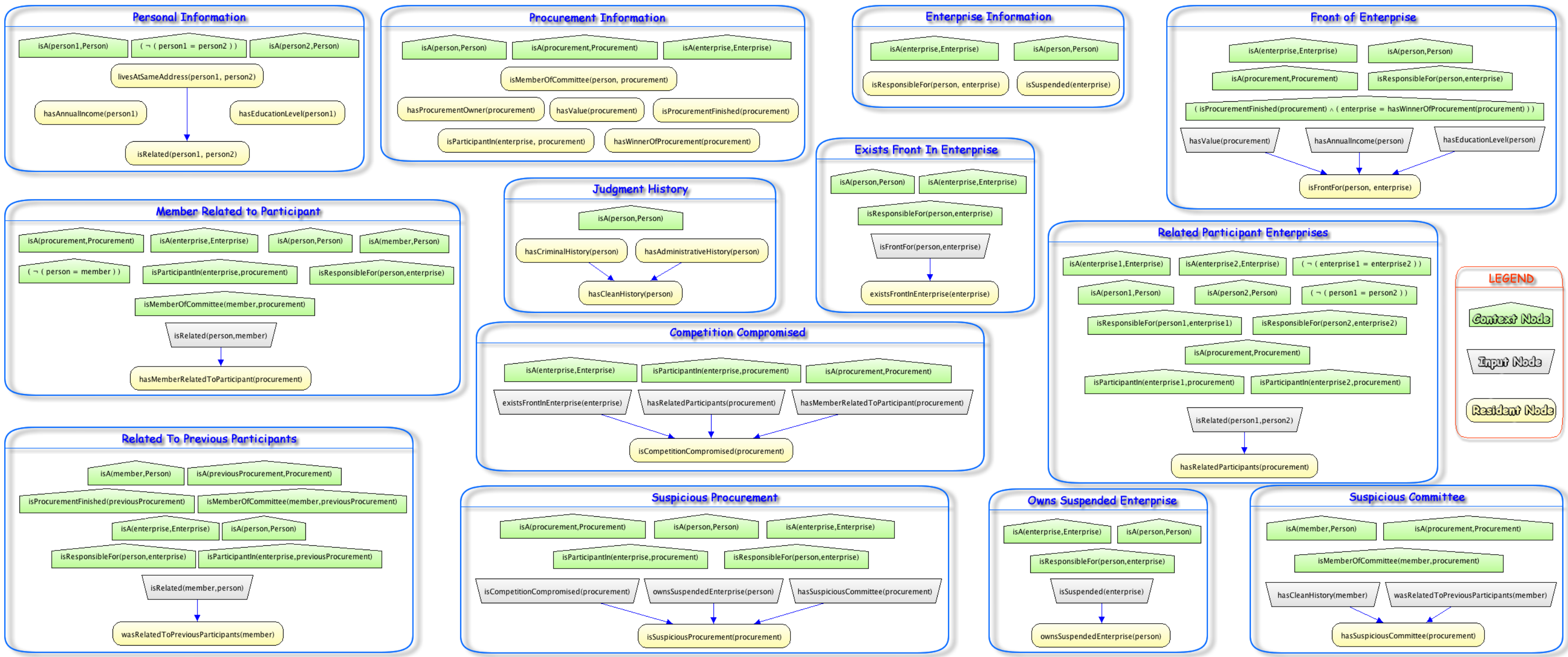


Figure 5.8: Probabilistic ontology for fraud detection and prevention in public procurements.

13. Suspicious Procurement

Table 5.3: Requirements Traceability Matrix for the MFrags of the fraud detection model.

ID	1	2	3	4	5	6	7	8	9
1	X	X	X		X	X		X	X
2	X	X	X	X	X	X	X	X	X
3	X	X	X	X	X	X		X	X
4			X			X			
5			X			X			
6					X	X			
7	X	X				X		X	X
8	X	X	X		X	X		X	X
9				X					
10							X		X
11								X	
12							X	X	
13	X	X	X	X	X	X	X	X	X

Table 5.3 presents the traceability between the MFrags defined in the Implementation stage and the rules defined in the Analysis & Design stage. This mapping, together with the mapping of the rules to the requirements presented in Table 5.2 provides the mapping that defines which requirements the MFrags are realizing.

Figure 5.8 presents an MTheory, in PR-OWL 1, that represents the final probabilistic ontology for the procurement fraud detection and prevention model. This MTheory is composed of nine MFrags. In each MFrag, the resident RVs are shown as yellow rounded rectangles; the input RVs are shown as gray trapezoids; the context RVs are shown as green pentagons. The two main goals described in our requirements are defined in the `Suspicious Procurement` and `Suspicious Committee` MFrags. A more sophisticated design to model whether to do further investigation or whether to change the committee would define a utility function and use expected utility to make the decision. Future versions of UnBBayes will support Multi-Entity Influence Diagrams [27].

The final step in constructing a probabilistic ontology in UnBBayes is to define the local probability distribution (LPD) for all resident nodes (in PR-OWL 2 the default distribution is defined only once on the RV itself). Figure 5.9 presents a LPD for the resident node `isSuspiciousProcurement(procurement)`, which is the main question we need to answer in order to achieve one of the main goals in our model. This distribution follows UnBBayes-MEBN expressive grammar for defining LPDs. For more information see [16, 19].

Appendix B Section B.1 presents the details and explanations of all MFrags and all resident nodes and their respective LPDs of the probabilistic ontology discussed in this Section.

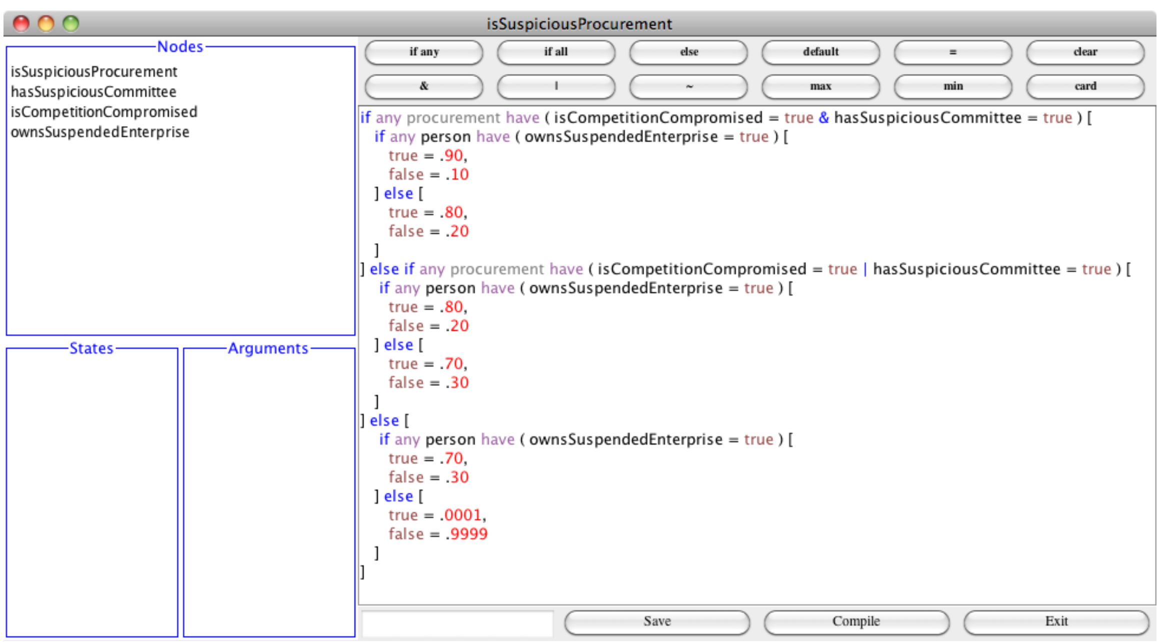


Figure 5.9: LPD for node `isSuspiciousProcurement(procurement)`.

5.1.4 Test

In most modeling methodologies, test plays an essential role. This is no different in the UMP-ST methodology. As Laskey and Mahoney [81] point out, test should not just be for showcase and to demonstrate that the model works. The Test discipline goal is to find flaws and areas for improvement in the model.

Before we start describing the activities in the Test discipline, it is important to understand the different types of evaluation that need to be done. The literature distinguishes two types of evaluation, verification and validation [6]. On the one hand, verification is concerned with delivering all the functionality promised to the customer. This usually involves reviewing requirements, documentation, design, and code. Verification is often done through inspections and by following checklists. On the other hand, validation is concerned with the correct behavior of the system. Validation is the actual testing of the system and it is done after verification.

A common slogan that summarizes the main difference between verification and validation is that verification tests whether the system was built right; validation tests whether we built the right system.

For instance, in the model we have been describing in this Section we would like to verify that all queries covered by the requirement are indeed being answered in less than a minute and that the posterior probability given as an answer to a given query is either exact or has an approximation with an error bound of .5% or less. These are non-functional requirements described during our Requirements stage in Subsection 5.1.1.

Although verification is an important and necessary evaluation, I will focus on describing how to validate our model. Laskey and Mahoney [81] present three types of validation: elicitation review, importance analysis, and case-based evaluation.

Elicitation review is related to reviewing the model documentation, analysing if all the requirements were addressed on the final model, making sure all the rules defined during the Analysis & Design stage were implemented, validating the semantics of the concepts described by the model, etc. This is an important step towards achieving consistency in our model, especially if it was designed by more than one expert.

A good way to verify if all the requirements were addressed in the final implementation of the model is to look at the RTM matrices. By looking at the RTM matrix for the MFrags implemented in our model we can verify that all the rules defined during Analysis & Design were covered. Since the RTM matrix of the rules defined during Analysis & Design covered all the requirements, then we can infer that all the requirements were implemented in our model.

Importance analysis measures the strength of a link between nodes using some kind of sensitivity analysis method [75,96]. According to [81], "importance analysis for a given variable (called *focus* variable) measures the impact on the focus variable's belief of obtaining evidence about each of a set of other variables (the *evidence* variables)."

In this section I will focus on case-based evaluation, which is defining different scenarios to test our model. One type of case-based evaluation is case-based unit testing. In case-based unit testing we want to test the behavior of part of the model, more specifically, verifying how the focus variable behaves with different set of evidence. In the case of PR-OWL, we can analyze the behavior of the random variables of interest given evidence per MFrag. This MFrag testing is important to capture local consistency of the model.

As an example of unit testing, I demonstrate how to define different scenarios to test the `JudgmentHistory_MFrag`. Essentially, we want to verify how the query `hasCleanHistory(person)` will behave in light of different set of evidence for a person's criminal and administrative history.

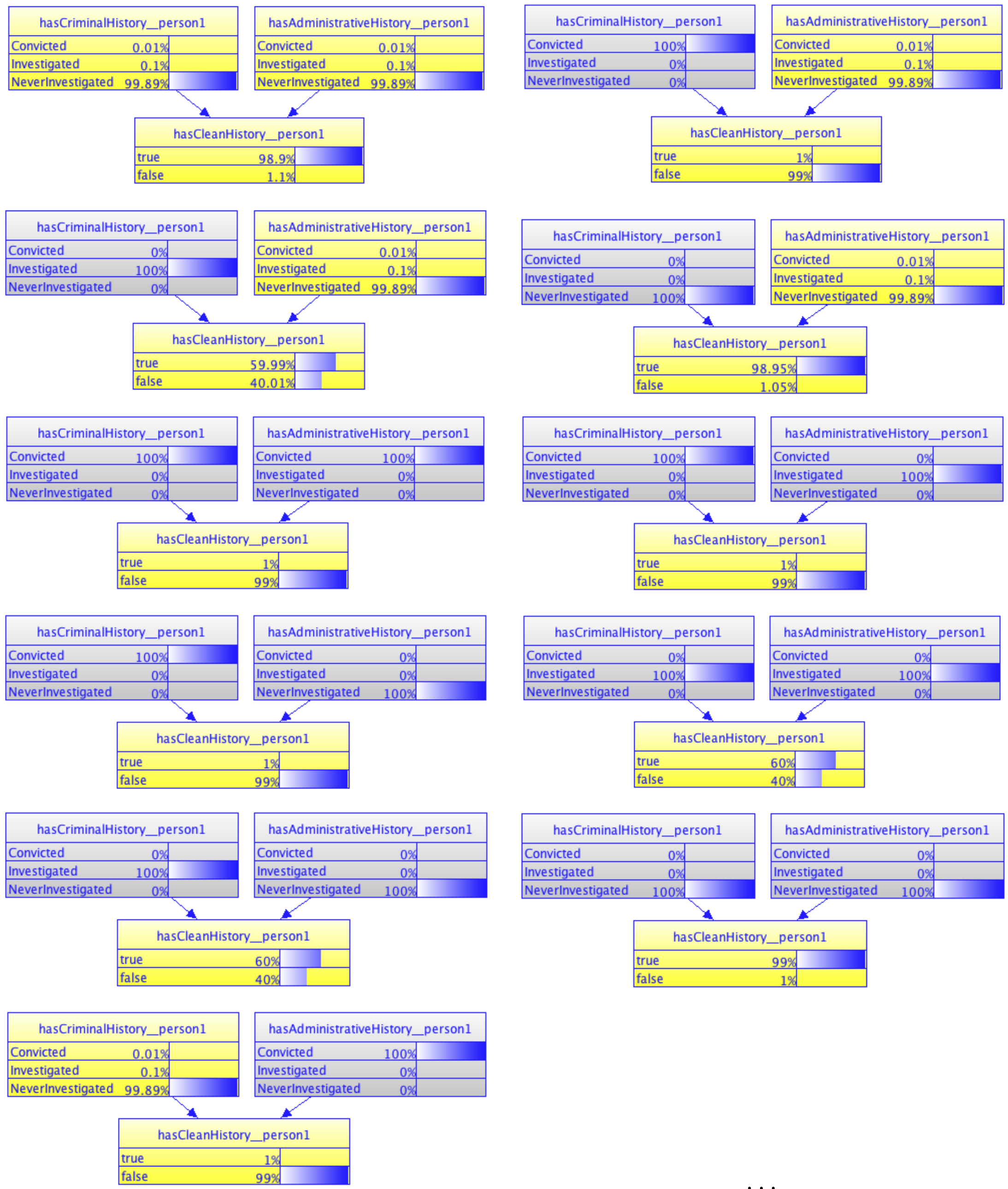


Figure 5.10: Results of unit testing for the `JudgmentHistory_MFrag`.

Notice that we do not show all possible combinations of the states for each node in Figure 5.10, since their behavior is similar in the sense that stating that `hasCriminalHistory(person1) = Convicted` and `hasAdministrativeHistory(person1) = Investigated` is the same thing as stating that `hasCriminalHistory(person1) = Investigated` and `hasAdministrativeHistory(person1) = Convicted`, and so on. The important thing to do is to try to cover as much as possible and to analyze the results by verifying if the posterior probabilities behave as expected. In our case, the posterior probabilities are consistent with the expected result as defined by the expert. In this MFrag the focus variable is the child, however, in other MFrags the focus variable might the parent and thus we would want to evaluate the behavior of a parent node given evidence on the children, which is the opposite of what was done here.

The other type of case-based evaluation is concerned with the behavior of the model as a whole. As such, I use it as an important type of integration testing. In the case of PR-OWL, we can define scenarios with evidence that are represented in different MFrags. So, when we ask a query, the SSBN construction will instantiate different parts of the model, which helps us validate how the model works as a whole, and not just each part independently. This validation is important to capture global consistency of the model.

It is important to try out different scenarios in order to capture the nuances of the model. In fact, it is a good practice to design the scenarios in order to cover the range of requirements the model must satisfy [134, 124]. Although it is impossible to cover every scenario we might encounter, we should aim for good coverage, and especially look for important "edge cases". In order to illustrate this approach, let's define three different scenarios. The first one concerns a regular procurement with no evidence to support the hypothesis of a suspicious procurement or committee. The second one has conflicting evidence in the sense that some supports the hypothesis of having a suspicious procurement or committee but some does not. Finally, on the third scenario there is overwhelming evidence supporting the hypothesis of a suspicious procurement or committee. Nevertheless, a serious and more comprehensive evaluation of the model would have more than just three scenarios.

When defining a scenario, it is important to define the hypothesis being tested and what is the expected result, besides providing the evidence which will be used. In this use case I was the subject matter expert, since I work for the Brazilian Office of the Comptroller General (CGU), which is the Government Agency responsible for supervising and auditing projects which involve federal money.

In the first scenario we have the following:

1. Hypothesis being tested
   (a) `isSuspiciousProcurement(procurement)`
   (b) `isSuspiciousCommittee(procurement)`
2. Expected result
   (a) Low probability that `isSuspiciousProcurement(procurement1) = true`
   (b) Low probability that `isSuspiciousCommittee(procurement1) = true`
3. Evidence
   (a) `hasAdministrativeHistory(member1) = NeverInvestigated`
   (b) `hasCriminalHistory(member2) = NeverInvestigated`
   (c) `hasProcurementOwner(procurement1) = agency1`
   (d) `isMemberOfCommittee(member1, procurement1) = true`
   (e) `isMemberOfCommittee(member2, procurement1) = true`
   (f) `isMemberOfCommittee(member3, procurement1) = true`
   (g) `isParticipantIn(enterprise1, procurement1) = true`
   (h) `isParticipantIn(enterprise2, procurement1) = true`
   (i) `isParticipantIn(enterprise3, procurement1) = true`
   (j) `isProcurementFinished(procurement1) = false`
   (k) `isResponsibleFor(person1, enterprise1) = true`

(l) `isResponsibleFor(person2, enterprise2) = true`

(m) `isResponsibleFor(person3, enterprise3) = true`

Figure 5.11 presents part of the SSBN network generated from scenario 1 and as expected the probability of both `isSuspiciousProcurement(procurement1) = true` and `isSuspiciousCommittee(procurement1) = true` are low, 2.35% and 2.33%, respectively.

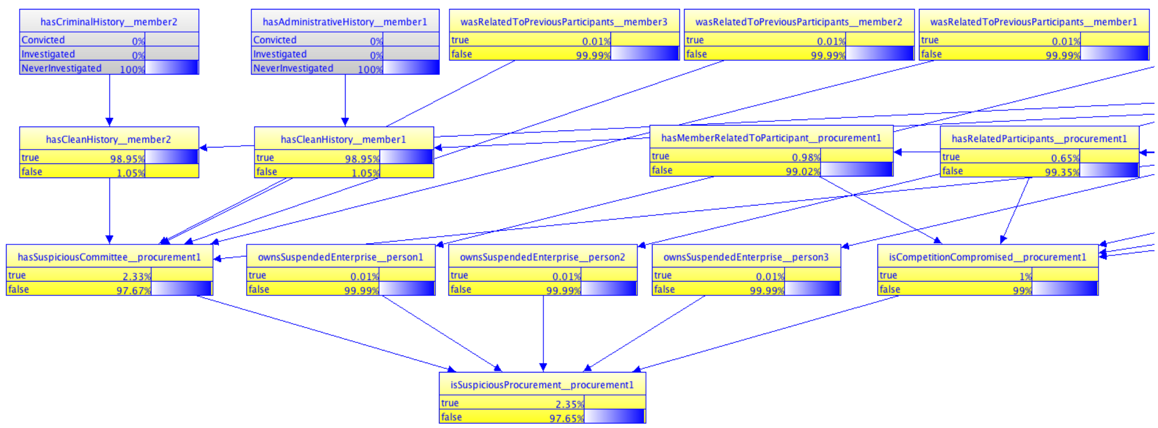


Figure 5.11: Part of the SSBN generated for the first scenario.

In the second scenario we have the following:

1. Hypothesis being tested

   (a) `isSuspiciousProcurement(procurement)`

   (b) `isSuspiciousCommittee(procurement)`

2. Expected result

   (a) Probability that `isSuspiciousProcurement(procurement1) = true` between 10% and 50%

   (b) Probability that `isSuspiciousCommittee(procurement1) = true` between 10% and 50%

3. Evidence (in italic we have the new evidence compared to scenario 1)

   (a) *`hasAdministrativeHistory(member1) = Investigated`*

   (b) `hasAdministrativeHistory(member1) = NeverInvestigated`

   (c) `hasCriminalHistory(member2) = NeverInvestigated`

   (d) `hasProcurementOwner(procurement1) = agency1`

   (e) `isMemberOfCommittee(member1, procurement1) = true`

   (f) `isMemberOfCommittee(member2, procurement1) = true`

   (g) `isMemberOfCommittee(member3, procurement1) = true`

   (h) `isParticipantIn(enterprise1, procurement1) = true`

   (i) `isParticipantIn(enterprise2, procurement1) = true`

   (j) `isParticipantIn(enterprise3, procurement1) = true`

   (k) `isProcurementFinished(procurement1) = false`

   (l) `isResponsibleFor(person1, enterprise1) = true`

   (m) `isResponsibleFor(person2, enterprise2) = true`

   (n) `isResponsibleFor(person3, enterprise3) = true`

Figure 5.12 presents part of the SSBN network generated from scenario 2 and as expected the probability of both `isSuspiciousProcurement(procurement1) = true` and `isSuspiciousCommittee(procurement1) = true` are 20.82% and 28.95%, respectively.

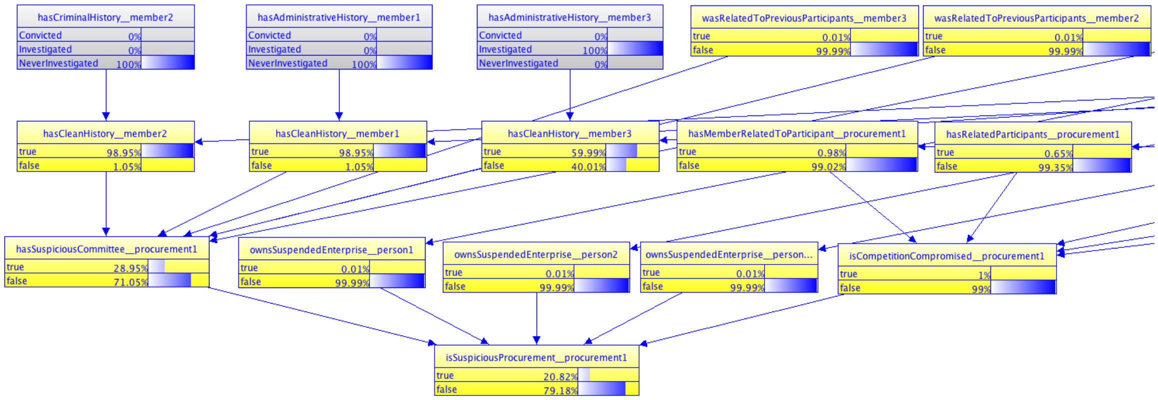


Figure 5.12: Part of the SSBN generated for the second scenario.

In the third scenario we have the following:

1. Hypothesis being tested

   (a) `isSuspiciousProcurement(procurement)`

   (b) `isSuspiciousCommittee(procurement)`

2. Expected result

   (a) Probability that `isSuspiciousProcurement(procurement1) = true` greater than 50%

   (b) Probability that `isSuspiciousCommittee(procurement1) = true` between 10% and 50%

3. Evidence (in italic we have the new evidence compared to scenario 2)

   (a) *`livesAtSameAddress(person1, person3)`*

   (b) *`livesAtSameAddress(person2, member3)`*

   (c) `hasAdministrativeHistory(member1) = Investigated`

   (d) `hasAdministrativeHistory(member1) = NeverInvestigated`

(e) `hasCriminalHistory(member2) = NeverInvestigated`

(f) `hasProcurementOwner(procurement1) = agency1`

(g) `isMemberOfCommittee(member1, procurement1) = true`

(h) `isMemberOfCommittee(member2, procurement1) = true`

(i) `isMemberOfCommittee(member3, procurement1) = true`

(j) `isParticipantIn(enterprise1, procurement1) = true`

(k) `isParticipantIn(enterprise2, procurement1) = true`

(l) `isParticipantIn(enterprise3, procurement1) = true`

(m) `isProcurementFinished(procurement1) = false`

(n) `isResponsibleFor(person1, enterprise1) = true`

(o) `isResponsibleFor(person2, enterprise2) = true`

(p) `isResponsibleFor(person3, enterprise3) = true`

Figure 5.13 presents part of the SSBN network generated from scenario 3 and as expected the probability of both `isSuspiciousProcurement(procurement1) = true` and `isSuspiciousCommittee(procurement1) = true` are 60.08% and 28.95%, respectively.

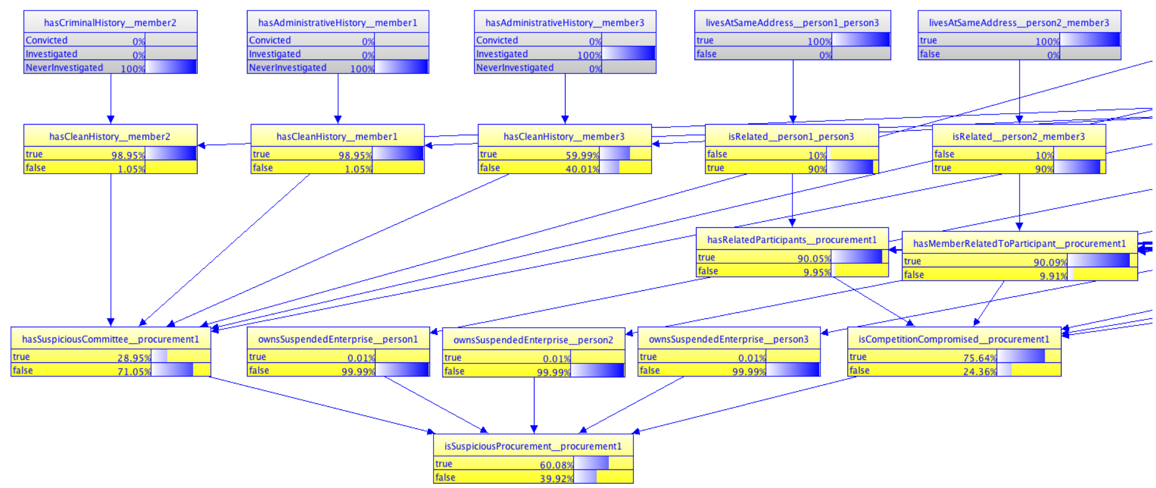


Figure 5.13: Part of the SSBN generated for the third scenario.

5.2 Probabilistic Ontology for Maritime Domain Awareness

Maritime Domain Awareness (MDA) involves the ability to automatically integrate information from multiple sources in a complex and evolving scenario to produce a dynamic, comprehensive, and accurate picture of the naval operations environment. The emphasis on net-centric operations and the shift to asymmetric warfare have added an additional level of complexity and technical challenge to automated information integration and predictive situation assessment. A probabilistic ontology (PO) is a promising tool to address this challenge. The PO for Maritime Domain Awareness (MDA) described in this Section was presented in [17, 18] and is part of the PROGNOS project [32, 33].

PROGNOS (PRobabilistic OntoloGies for Net-centric Operation Systems) is a naval predictive situational awareness system devised to work within the context of U.S. Navy's FORCENet. The system uses the UnBBayes-MEBN framework, which implements a MEBN reasoner capable of saving MTheories in PR-OWL format.

The focus of this Section is to highlight the key role iterations play in incrementally expanding the model during its lifecycle. In this Section I will not present as much detail in each discipline as I did in Section 5.1. Instead I will highlight how we can leverage the UMP-ST process and PR-OWL's modularity in order to minimize change in the existing model as we add new requirements in new iterations.

The PROGNOS MDA PO was created using the Uncertainty Model for Semantic Technologies (UMP-ST) and the Probabilistic Ontology Modeling Cycle (POMC) with the support of the stakeholders (MEBN and PR-OWL experts and subject matter experts, who are retired officers from US Navy and US Coast Guard, Richard Haberlin and Michael Lehocky, respectively). The probabilistic ontology developed so far has passed through three iterations. The first iteration consists of a simple model to identify whether a ship is of interest. The second iteration expanded the model to provide clarification of the reasons behind declaring a ship of interest. The third iteration focused on detecting an individual crew member's terrorist affiliation given his close relations, group associations, communications, and background influences.

5.2.1 First Iteration

Requirements

The original model consists of the following set of goal/query/evidence:

1. Identify whether a ship is of interest, *i.e.*, it seems to be suspicious in any way.
    (a) Does the ship have a terrorist crew member?
        i. Verify if a crew member is related to any terrorist;
        ii. Verify if a crew member is associated with any terrorist organization.
    (b) Is the ship using an unusual route?
        i. Verify if there is a report that the ship is using an unusual route;
        ii. Verify if there is a report that the ship is meeting some other ship for no apparent reason.
    (c) Does the ship seem to exhibit evasive behavior?

i. Verify if an electronic countermeasure (ECM) was identified by a navy ship;

ii. Verify if the ship has a responsive radio and automatic identification system (AIS).

Analysis & Design

Once we have defined our goals and described how to achieve them, it is time to start modeling the entities, their attributes, relationships, and rules to make that happen. This is the purpose of the Analysis & Design discipline.

Figure 5.14 depicts a simplified design of our domain requirements. A `Ship` is a ship of interest, `isOfInterest`, if it represents some kind of threat. A `Ship` has a crew, which is represented by `hasCrewmember` and the inverse relation `isCrewmemberOf`. It is assumed that a ship represents some kind of threat if and only if one of its crew members is a `Terrorist` (subclass of `Person`).

The social network information available determines that a `Person` might be related to another `Person` by `isRelatedTo`. Moreover, a `Person` might be a member of an `Organization`, represented by the `isMemberOf` and the inverse `hasMember` relations. An `Organization` might be a `TerroristOrganization` (subclass of `Organization`). It is also assumed that a `Person` related to a `Terrorist` is more likely to be a `Terrorist` and an `Organization` that has a `Terrorist` member is more likely to be a `TerroristOrganization`.

This model is simplified in the sense that it represents a screenshot in time of the domain. In other words, there is only one possible crew for a given `Ship` and a `Person` can only be a crew member of a unique `Ship`. Following the same rationale, a `Ship` can only have one possible `Position`, represented by `hasPosition`.

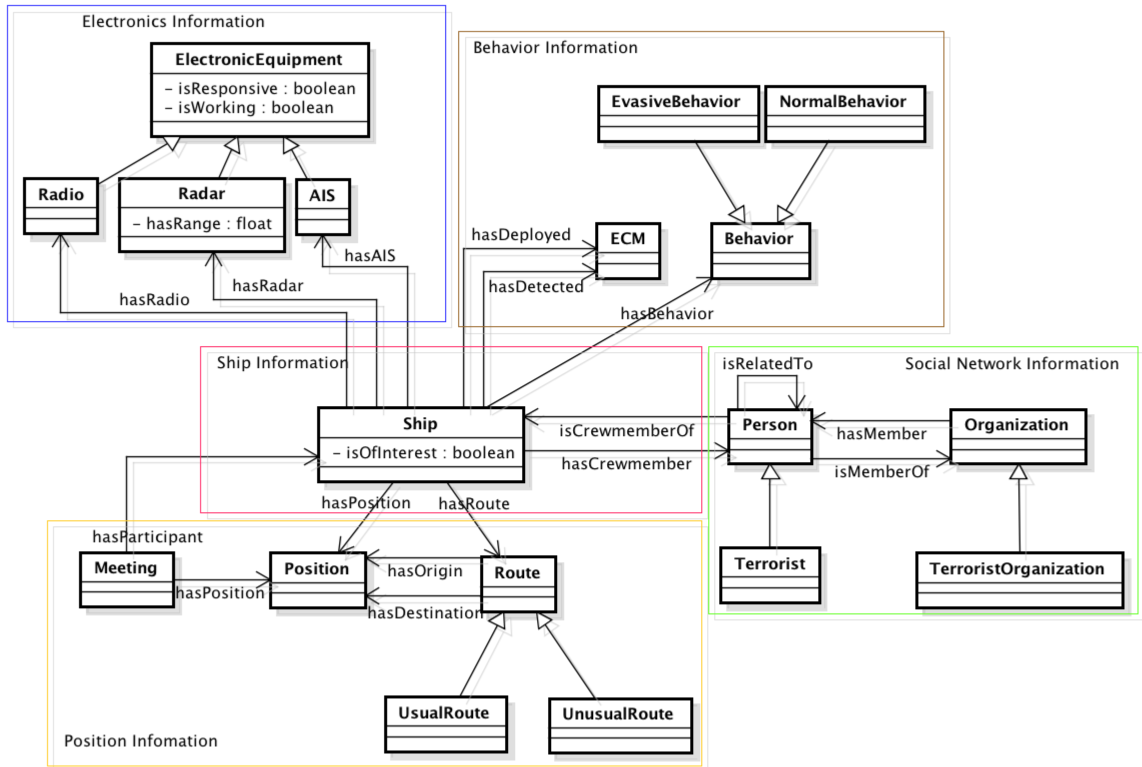


Figure 5.14: Entities, their attributes, and relations for the MDA model after the first iteration.

The `Position` of a `Ship` is usually consistent with its `Route`. A `Route` has a specific origin and destination `Position`, represented by `hasOrigin` and `hasDestination`, respectively. If the `Ship` is following the usual route from its origin to its destination, then its `Route` is said to be a `UsualRoute`, otherwise, if the `Ship` is going to places that are not consistent with the expected route (safest/shortest distance from origin to destination), then its `Route` is said to be an `UnusualRoute`. Furthermore, usually ships try to avoid getting too close to each other, therefore, if two of more ships get too close together, it is said that they are `Meeting` in a certain `Position`, represented by `hasPosition`. The ships participating in this `Meeting` are represented by `hasParticipant`, which maps a `Meeting` to two or more ships (`Ship`). If two or more ships are meeting, then it is more likely that they doing some illicit transaction on the ocean, therefore, they will probably meet at an unusual `Position`, which means that they are on an `UnusualRoute`. One example illustrating this idea is that a ship carrying Weapons of Mass Destruction (WMD) might want to pass its dangerous cargo to one or more smaller ships in order to increase the chances of infiltrating the coast with the WMD.

As for the electronic equipment described in this model, `ElectronicEquipment`, a `Ship` can have an Automatic Identification System (AIS), represented by `hasAIS`, which is used for identifying and locating vessels by electronically exchanging data with other nearby ships and Vessel Traffic Services (VTS) stations. Moreover, a `Ship` usually has at least one `Radar`, represented by `hasRadar`, with a specific range, defined by `hasRange`. The range is defined in this model by a `float` number, however, in a more realistic and detailed model this should be a measure of distance, *i.e.*, a class by itself with value and unit of measure. `AIS` and `Radar` are subclasses of `EletronicEquipment` and as such, they can be responsive, represented by `isReponsive`, which entails that they are working, represented by `isWorking`, and turned on.

A `Ship` might have different behaviors (`Behavior`). A `Ship` might deploy an Electronic Countermeasure (ECM), represented by `hasDeployed`. Besides that, a different `Ship` might detect an ECM, represented by `hasDetected`, although it does not necessarily know which `Ship` deployed it. To be able to detect an ECM, the ship that deployed the ECM has to be in the `Radar` range of the `Ship` that detects it. An ECM is a subsection of electronic warfare, which includes any sort of electrical or electronic device designed to trick or deceive radar, sonar, or other detection systems. It may be used both offensively and defensively in any method to deny targeting information to an enemy. A `Ship` that has deployed an ECM is said to have exhibited an `EvasiveBehavior`. Furthermore, if an `ElectronicEquipment` is working but is not responsive, then the `Ship` is also said to have exhibited an `EvasiveBehavior`. In all other cases, the `Ship` is said to have `NormalBehavior`. As shown, `EvasiveBehavior` and `NormalBehavior` are subclasses of `Behavior`.

Besides the cardinality and uniqueness rules defined in the explanation above about the entities depicted in Figure 5.14, the probabilistic rules for our model include:

1. A ship is of interest if and only if it has a terrorist crew member;
2. If a crew member is related to a terrorist, then it is more likely that he is also a terrorist;
3. If a crew member is a member of a terrorist organization, then it is more likely that he is a terrorist;
4. If an organization has a terrorist member, it is more likely that it is a terrorist organization;
5. A ship of interest is more likely to have an unusual route;
6. A ship of interest is more likely to meet other ships for trading illicit cargo;
7. A ship that meets other ships to trade illicit cargo is more likely to have an unusual route;
8. A ship of interest is more likely to have an evasive behavior;
9. A ship with evasive behavior is more likely to have non responsive electronic equipment;
10. A ship with evasive behavior is more likely to deploy an ECM;
11. A ship might have non responsive electronic equipment due to working problems;
12. A ship that is within radar range of a ship that deployed an ECM might be able to detect the ECM, but not who deployed it.

Implementation

Once we have finished our Analysis & Design, it is time to start implementing our model in a specific language. In this project we implement our model in PR-OWL using UnBBayes.

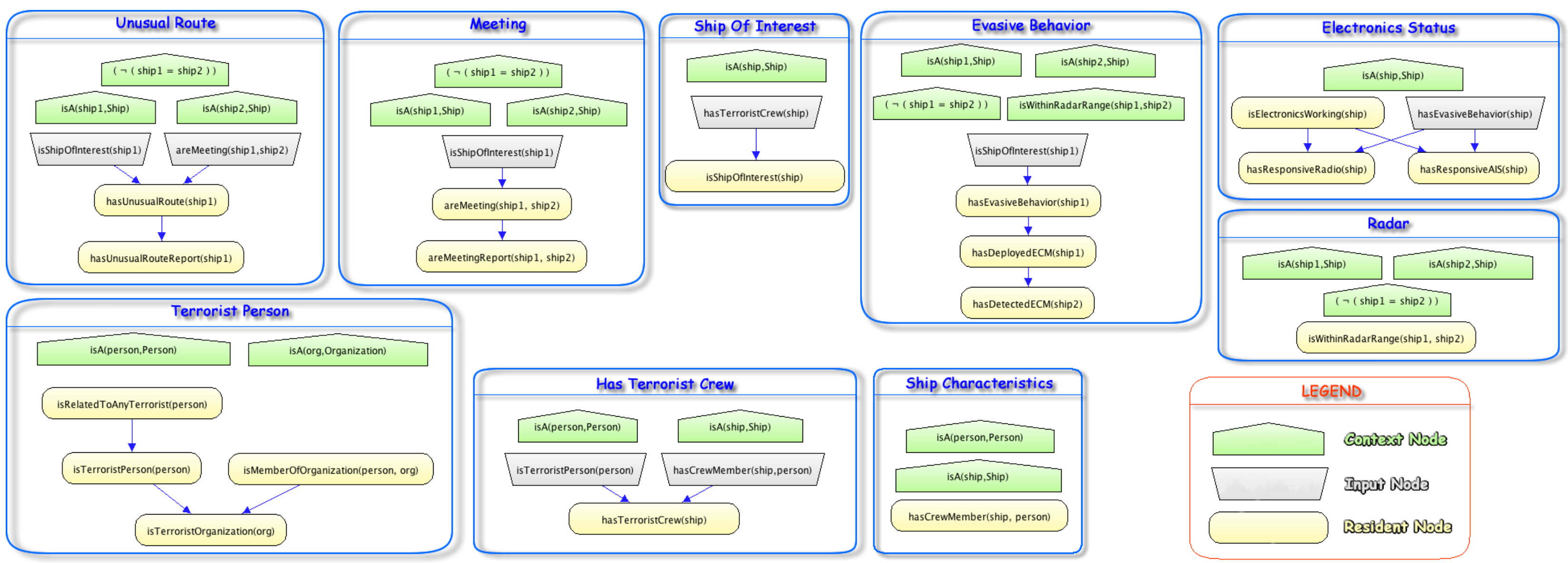


Figure 5.15: MTheory created in first iteration.

The final result of this initial iteration is the PO depicted in Figure 5.15. There, the hypotheses related to the identification of a terrorist crew member are presented in the `Has Terrorist Crew`, `Terrorist Person`, and `Ship Characteristics` MFrags. The hypotheses related to the identification of unusual routes are presented on the `Unusual Route` and `Meeting` MFrags. Finally, the hypotheses related to identification of evasive behavior are shown in the `Evasive Behavior`, `Electronics Status`, and `Radar` MFrags.

Appendix B Subsection B.2.1 presents the details and explanations of all MFrags and all resident nodes and their respective LPDs of the probabilistic ontology discussed in this Subsection.

Test

Although I have described many different types of evaluation and tests we can perform in our model in Subsection 5.1.4, this iteration will focus on performing integration test based on case-based evaluation.

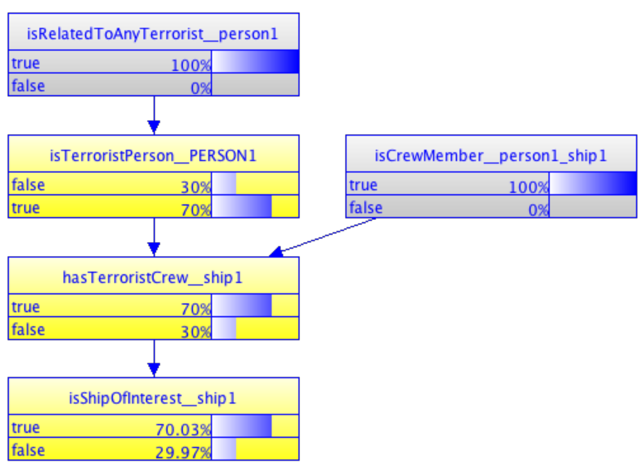


Figure 5.16: SSBN generated for scenario 1.

I will illustrate 5 different scenarios by increasing not only the complexity of the generated model, but also the probability that `ship1` is of interest. These increases are due to new evidence that is available in every new scenario, which supports the hypothesis that `ship1` is of interest.

In scenario 1, the only information available is that `person1` is a crew member of `ship1` and that `person1` is related to at least one terrorist. Figure 5.16 shows that there is a 70.03% probability of `ship1` being of interest, which is consistent with the fact that one of its crew members might be a terrorist.

In scenario 2, besides having the information available from scenario 1, it is also known that `ship1` met `ship2`. Figure 5.17 shows the probability of `ship1` being of interest has increased to 89.41%, which is consistent with the new supporting evidence that `ship1` met `ship2`.

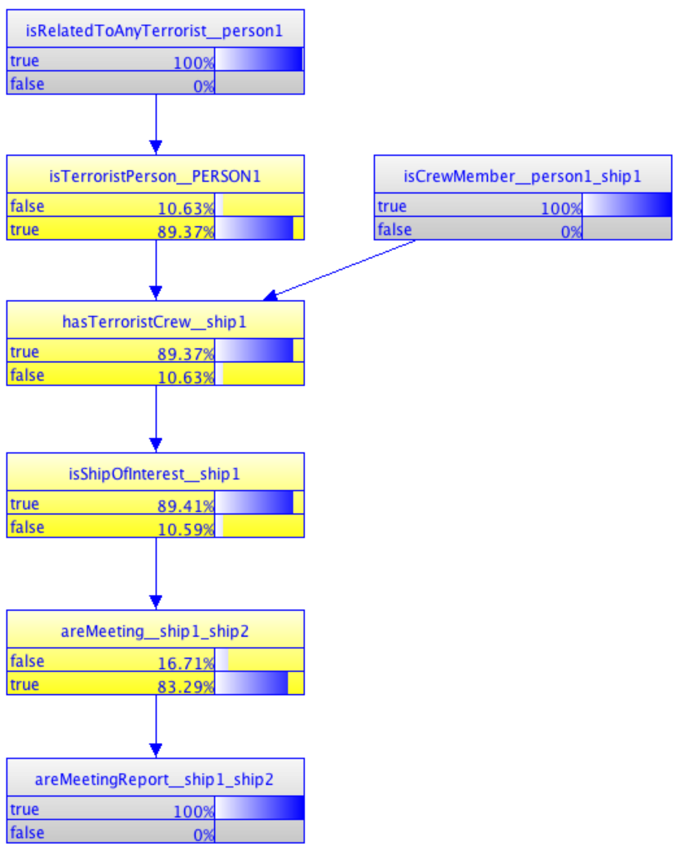


Figure 5.17: SSBN generated for scenario 2.

In scenario 3, besides having the information available from scenario 2, it is also known that **ship1** has an unusual route. Figure 5.18 shows the probability of **ship1** being of interest has increased to 97.19%, which is consistent with the new supporting evidence that **ship1** is not going to its destination using a normal route.

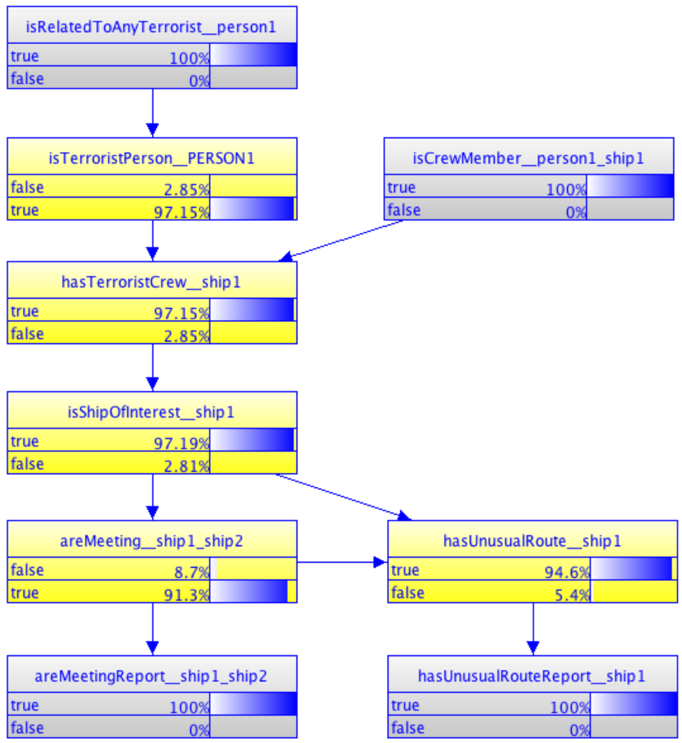


Figure 5.18: SSBN generated for scenario 3.

In scenario 4, besides having the information available from scenario 3, it is also known that **navyShip** has detected an ECM. Figure 5.19 shows the probability of **ship1** being of interest has increased to 99.97%, which is consistent with the new supporting evidence that **ship1** is probably the ship that deployed the ECM. It is important to notice that there are only two ships that could deploy the ECM in this scenario, which are the ships within range of **navyShips** radar (**ship1** and **ship2**). From the other evidence that supports the fact that **ship1** is most likely a ship of interest, it becomes more likely that **ship1** is the one that deployed the ECM. That is why the probability that **ship2** having deployed the ECM is so low (due to explaining away).

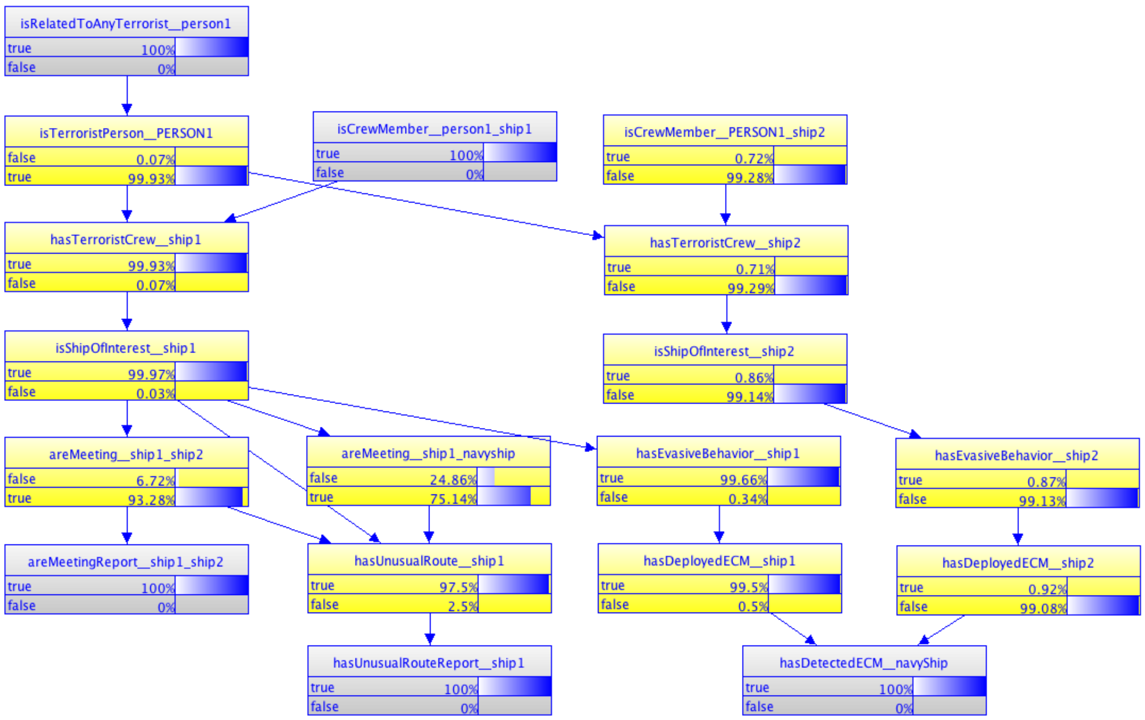


Figure 5.19: SSBN generated for scenario 4.

In scenario 5, besides having the information available from scenario 4, it is also known that **ship1** does not have a responsive radio nor a responsive AIS. Figure 5.20 shows that the probability of **ship1** being of interest is 100.00%.

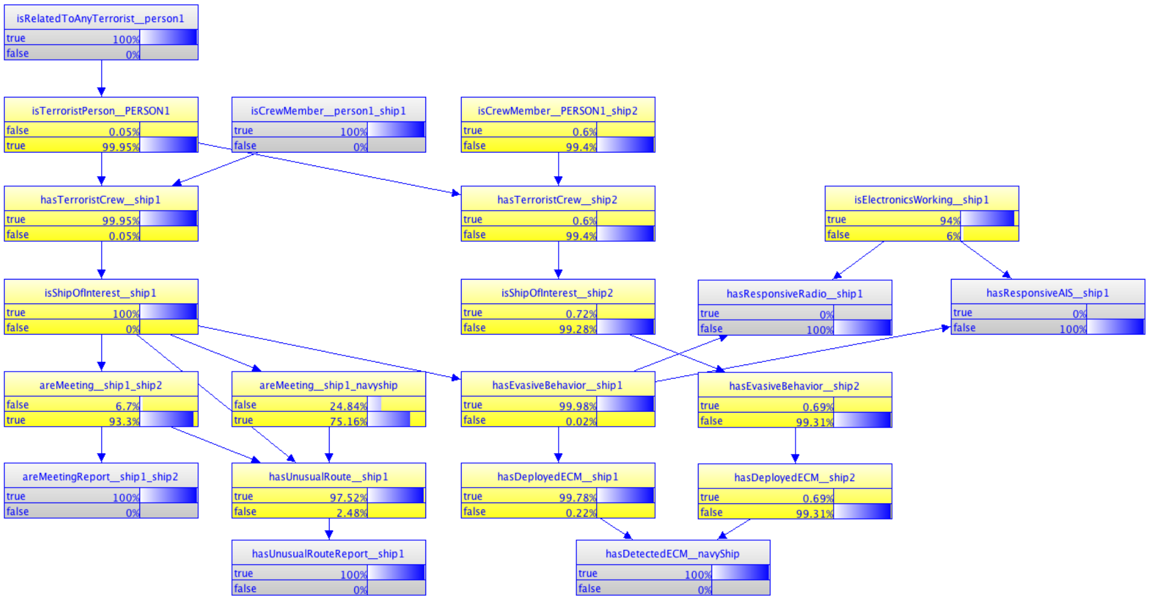
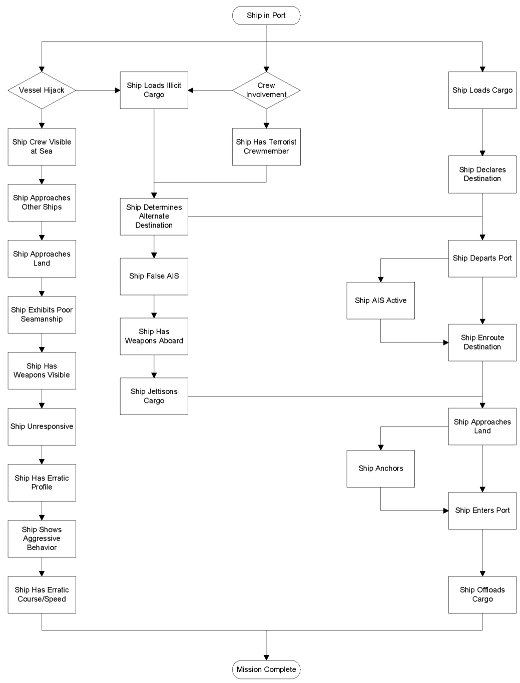


Figure 5.20: SSBN generated for scenario 5.

5.2.2 Second Iteration

Once the initial model was built and tested, the second iteration shifted focus to understanding the reasons for classifying a ship's behavior as suspicious. The approach was to define possible terrorist plans that might result in specific behaviors. At this stage, two terrorist plans were taken into consideration: exchange illicit cargo (*e.g.*, explosives) and bomb a port using a suicide ship. Another distinction from the original model is that the behavior depends not only on the plan being executed, but also on the type of the ship. In addition, there are now two reasons why a ship might be executing a terrorist plan: it either has a terrorist crew member (the only option in the original model) or the ship was hijacked.



(a) Merchant ship with exchange illicit cargo plan on the left, and normal behavior on the right.

(b) Fishing ship with bomb a port plan on the left, and normal behavior on the right.

Figure 5.21: Normal and suspicious behavior of merchant and fishing ships.

Figure 5.21 provides an activity diagram with the expected behaviors of ships involved in illicit activities on the left, and what would be the normal behavior from ships with no terrorist plan on the right.

Requirements

With the new task of identifying the terrorist plans associated to a suspicious ship (*i.e.*, exchanging illicit cargo, bombing a port, or no terrorist plan), the second iteration's set of goal/query/evidence was also expanded:

Identify whether a ship is a ship of interest, *i.e.*, if the ship has some terrorist plan associated with it.

1. Is the ship being used to exchange illicit cargo?

(a) Was the ship hijacked?

(b) *Does the ship have a terrorist crew member?*

i. *Verify if a crew member is related to any terrorist;*

ii. *Verify if a crew member is associated with any terrorist organization.*

(c) *Is the ship using an unusual route?*

i. *Verify if there is a report that the ship is using an unusual route;*

ii. *Verify if there is a report that the ship is meeting some other ship for no apparent reason.*

iii. Verify if the ship had a normal change in destination (*e.g.*, to sell the fish, which was just caught.)

(d) *Does the ship seem to exhibit evasive behavior?*

i. ~~*Verify if an electronic countermeasure (ECM) was identified by a navy ship;*~~

ii. *Verify if the ship has a responsive radio and automatic identification system (AIS).*

(e) Does the ship seem to exhibit erratic behavior?

i. Verify if the crew of the ship is visible.

(f) Does the ship seem to exhibit aggressive behavior?

i. Verify if the ship has weapons visible;

ii. Verify if the ship is jettisoning cargo.

2. Is the ship being used as a suicide ship to bomb a port?

(a) Was the ship hijacked?

(b) *Does the ship have a terrorist crew member?**

(c) *Is the ship using an unusual route?**

(d) Does the ship seem to exhibit aggressive behavior?*

Requirements inherited from the first iteration are in italic. Items crossed out refer to evidence considered by the SMEs, but that pertain only to war ships. Since these are not included in the scenarios they were excluded from the model. Queries marked with '*' are also used for another subgoal. For instance, an unusual route is expected both from ships with plan to bomb a port and from ships planning to exchange illicit cargo. The associated evidence is shown only for the first subgoal using the query.

Analysis & Design

As the original requirements were expanded, the UML model was also expanded to identify new concepts needed for achieving the new goals. Figure 5.22 displays the resulting model, with some classes added (*e.g.*, `Plan`, `TerroristPlan`, `TypeOfShip`, etc) and others removed (*e.g.*, `ECM`). Major changes are the new types of behavior (`AggressiveBehavior` and `ErraticBehavior`), the classification of ships (`TypeOfShip` and its subclasses), and planning information (`Plan`, `TerroristPlan`, and its subclasses). In addition, class `Ship` was expanded to allow for situational awareness of its behavior and to predict future actions based on it.

The next step is to define rules associated with the new requirements. The probabilistic rules below complement the cardinality and uniqueness rules in Figure 5.22 (same typing convention for rules inherited or not used in the model apply).

1. *A ship is of interest if and only if it has a terrorist ~~crew member~~* plan;
2. A ship has a terrorist plan if and only if it has terrorist crew member or if it was hijacked;
3. *If a crew member is related to a terrorist, then it is more likely that he is also a terrorist*;
4. *If a crew member is a member of a terrorist organization, then it is more likely that he is a terrorist*;

5. *If an organization has a terrorist member, it is more likely that it is a terrorist organization*;
6. *A ship of interest is more likely to have an unusual route*, independent of its intention;
7. *A ship of interest*, with plans of exchanging illicit cargo, *is more likely to meet other ships*;
8. *A ship that meets other ships to trade illicit cargo is more likely to have an unusual route*;
9. A fishing ship is more likely to have a normal change in its destination (*e.g.*, to sell the fish caught) than merchant ships;
10. A normal change in destination will probably change the usual route of the ship;
11. *A ship of interest*, with plans of exchanging illicit cargo, *is more likely to have an evasive behavior*;
12. A ship with evasive behavior is more likely to have non responsive electronic equipment;
13. A ship might have non responsive electronic equipment due to maintenance problems;
14. ~~A ship with evasive behavior is more likely to deploy an ECM~~;
15. ~~A ship that is within radar range of a ship that deployed an ECM might be able to detect the ECM, but not who deployed it~~;
16. A ship of interest, with plans of exchanging illicit cargo, is more likely to have an erratic behavior;
17. A ship with normal behavior usually does not have the crew visible on the deck;
18. A ship with erratic behavior usually has the crew visible on the deck;

19. If the ship has some equipment failure, it is more likely to see the crew on the deck in order to fix the problem;

20. A ship of interest, independent of its intention, is more likely to have an aggressive behavior;

21. A ship with aggressive behavior is more likely to have weapons visible and to jettison cargo;

22. A ship with normal behavior is not likely to have weapons visible nor to jettison cargo.

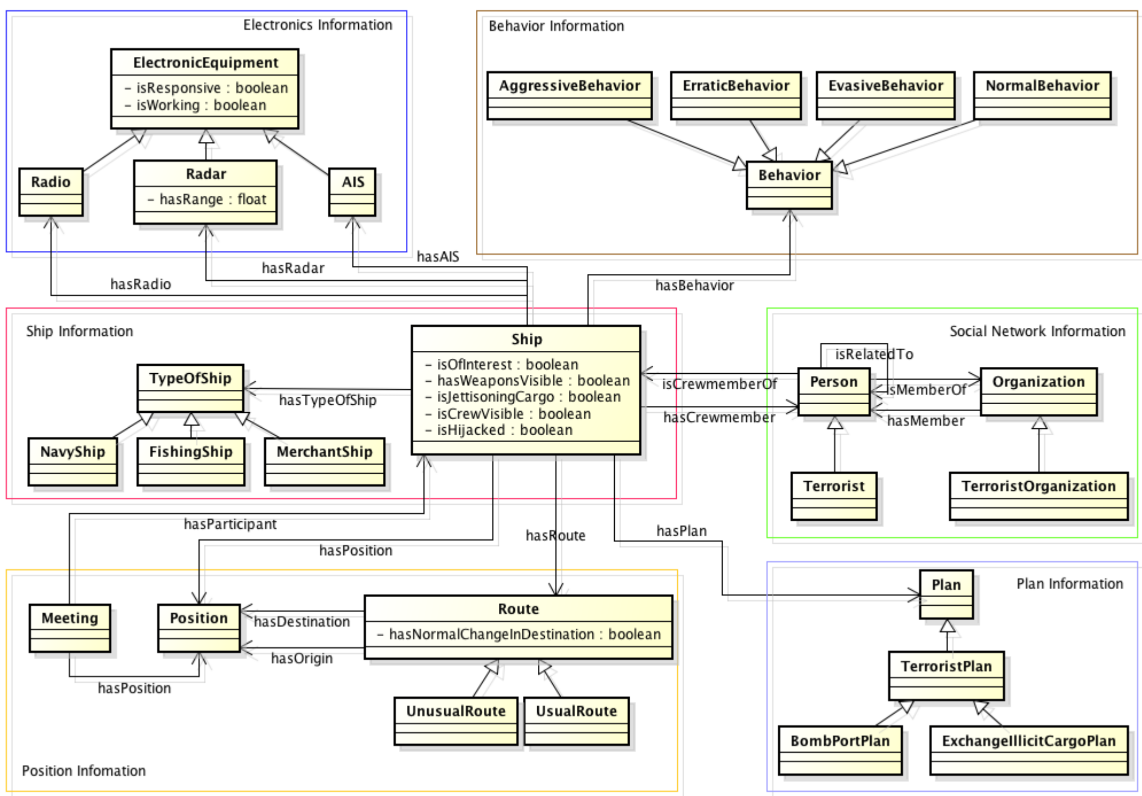


Figure 5.22: Entities, their attributes, and relations for the MDA model after the second iteration.

Implementation

Once the Analysis and Design stage is finished, implementation in a specific language (PR-OWL in this case) begins. The initial step is to map entities, attributes, and relations to PR-OWL. There is no need to map all entities in the model to entities in PR-OWL 1. In fact, the MDA model contains many simplifications. One is to define the random variable `hasTypeOfShip` mapping to values `Fishing` or `Merchant`, instead of creating them as subclasses. This can be done by creating a class in OWL using `oneOf` to specify the individuals that represent the class `ShipType`. Also, the original assumption of every entity being uniquely identified by its name still holds. The entities implemented in the MDA PO were `Person`, `Organization`, and `Ship`. All other entities were simplified in a similar manner as `ShipType`. For details on defining entities in UnBBayes see [20].

As explained in Subsection 5.1.3, in PR-OWL 2, it is not necessary to map these entities. In fact, the entities are defined as classes in a regular ontology using OWL. Then PR-OWL 2 simply makes use of them. As previously explained, our focus in this Section is to show how the model evolves when using the UMP-ST process, not on describing details on how to create a deterministic ontology.

After defining entities, the uncertain characteristics are identified. Uncertainty is represented in MEBN as random variables (RVs). On the one hand, to define a RV in PR-OWL 1 using UnBBayes, we first define its home MFrag. Grouping the RVs into MFrags is done by examining the grouping created during Analysis & Design. On the other hand, in PR-OWL 2 RVs are independent of the MFrags containing them and are defined globally by defining their arguments, mapping to OWL, and default distributions.

Typically, a RV represents an attribute or a relation in the designed model. For instance, the RV `isHijacked(Ship)` maps to the attribute `isHijacked` of the class `Ship` and the RV `hasCrewMember(Ship, Person)` maps to the relation `hasCrewMember` (refer to Figure 5.22). As a predicate relation, `hasCrewMember` relates a `Ship` to one `Person` or more, the same way class `Ship` might have one `Person` or more as its crew members. Hence, the possible values (or states) of this RV are `True` or `False`. Subclasses were avoided by using Boolean

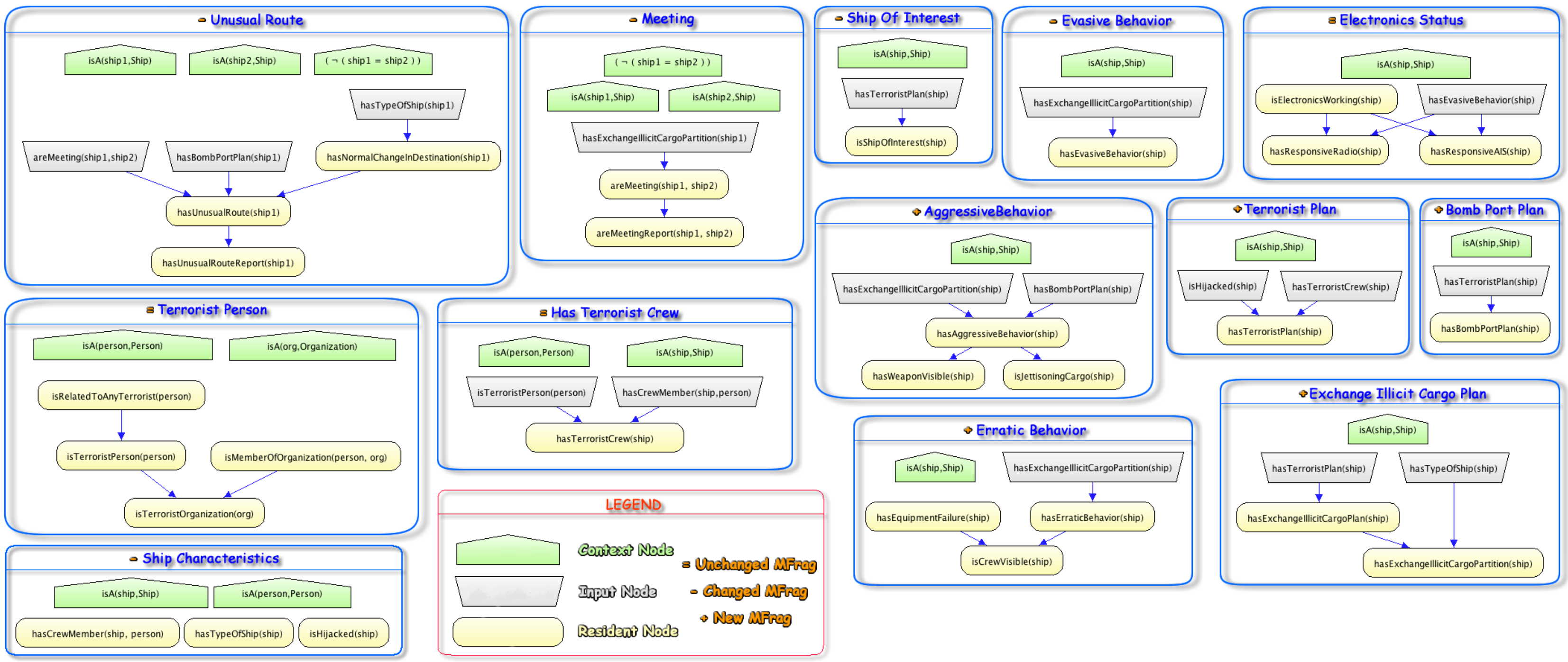


Figure 5.23: MTheory created in second iteration.

RV like `isTerrorist(Person)`, which represents the subclass `Terrorist`.

Once all resident RVs are created, their relations are defined by analyzing dependencies. This is achieved by looking at the rules defined in the model. For instance, the first rule indicates a dependence between `hasTerroristPlan(Ship)` and `isShipOfInterest(Ship)`. The structure of the relations added to the MDA PO can be seen in Figure 5.23.

After defining the relations, the local probability distributions are inserted for each resident node. For conciseness, these are not presented here but they must be consistent with the probabilistic rules defined in the Analysis & Design stage.

Appendix B Subsection B.2.2 presents the details and explanations of all MFrags and all resident nodes and their respective LPDs of the probabilistic ontology discussed in this Subsection.

Test

Although I have described many different types of evaluation and tests we can perform in our model in Subsection 5.1.4, this iteration will focus on performing integration test based on case-based evaluation, as was the case in the first iteration.

As explained in Subsection 5.1.4 it is important to try out different scenarios in order to capture the nuances of the model. In a serious test of the model, we would have to model a lot scenarios in order to cover at least the most important aspects of our requirements. However, I define only three qualitatively different scenarios in order to illustrate the mechanics of defining and testing a scenario. The first one has a regular ship with no evidence that supports the hypothesis of having a terrorist plan. The second one has conflicting evidence in the sense that some supports the hypothesis of having a terrorist plan but some does not. Finally, on the third scenario there is overwhelming evidence that supports the hypothesis of having a terrorist plan.

When defining a scenario, it is important to define the hypothesis being tested and what is the expected result, besides providing the evidence which will be used.

In the first scenario we have the following:

1. Hypothesis being tested
   (a) `isShipOfInterest(ship)`
   (b) `hasTerroristPlan(ship)`
2. Expected result
   (a) Low probability that `isShipOfInterest(ship1) = true`
   (b) High probability that `hasTerroristPlan(ship1) = NoPlan`
3. Evidence
   (a) `hasCrewMember(ship1, person1) = true`
   (b) `hasCrewMember(ship1, person2) = true`
   (c) `hasResponsiveRadio(ship1) = true`
   (d) `hasResponsiveAIS(ship1) = true`
   (e) `hasTypeOfShip(ship1) = Merchant`

Figure 5.24 presents the SSBN network generated from scenario 1 and as expected the probability of `isShipOfInterest(ship1) = true` is low and `hasTerroristPlan(ship1) = NoPlan` is high, 1.65% and 99.96%, respectively.

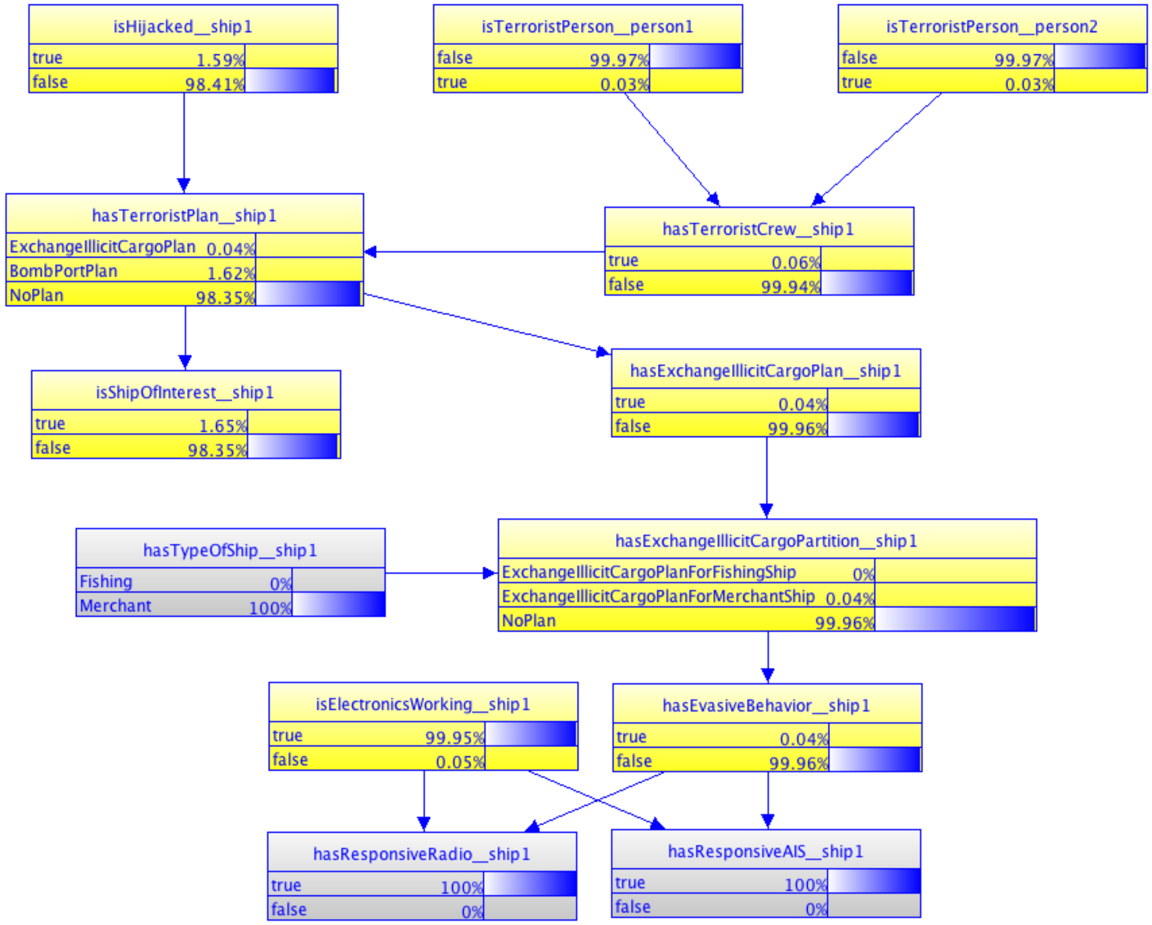


Figure 5.24: SSBN generated for the first scenario.

In the second scenario we have the following:

1. Hypothesis being tested

   (a) `isShipOfInterest(ship)`

   (b) `hasTerroristPlan(ship)`

2. Expected result

   (a) Probability that `isShipOfInterest(ship1) = true` between 33% and 67%

   (b) Probability that `hasTerroristPlan(ship1) = NoPlan` between 33% and 67%

3. Evidence (in italic we have the different evidence compared to scenario 1)

   (a) `hasCrewMember(ship1, person1) = true`

   (b) `hasCrewMember(ship1, person2) = true`

   (c) *`hasResponsiveRadio(ship1) = false`*

   (d) `hasResponsiveAIS(ship1) = true`

   (e) `hasTypeOfShip(ship1) = Merchant`

   (f) *`hasUnusualRouteReport(ship1) = true`*

Figure 5.25 presents part of the SSBN network generated from scenario 2 and as expected the probability of both `isShipOfInterest(ship1) = true` and `hasTerroristPlan(ship1) = NoPlan` are 44.27% and 58.60%, respectively.

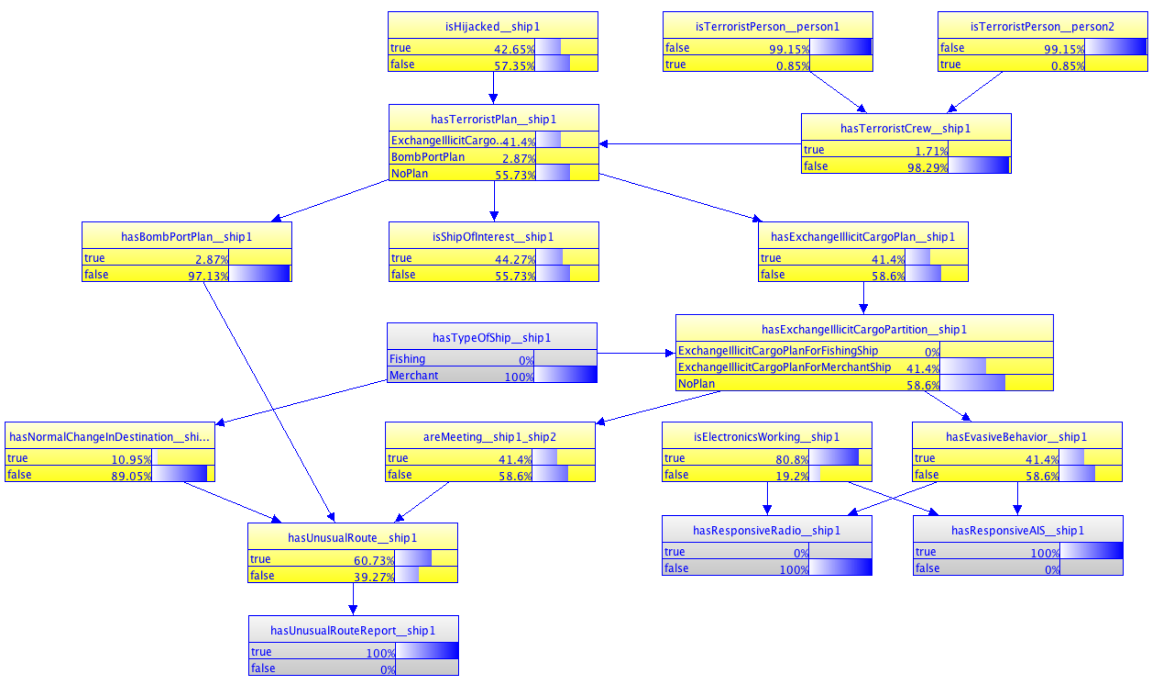


Figure 5.25: SSBN generated for the second scenario.

In the third scenario we have the following:

1. Hypothesis being tested

    (a) `isShipOfInterest(ship)`

    (b) `hasTerroristPlan(ship)`

2. Expected result

    (a) Probability that `isShipOfInterest(ship1) = true` greater than 50%

    (b) Probability that `hasTerroristPlan(ship1) = ExchangeIllicitCargoPlan` greater than 50%

3. Evidence (in italic we have the different evidence compared to scenario 2)

    (a) `hasCrewMember(ship1, person1) = true`

    (b) `hasCrewMember(ship1, person2) = true`

    (c) `hasResponsiveRadio(ship1) = false`

    (d) `hasResponsiveAIS(ship1) = true`

    (e) `hasTypeOfShip(ship1) = Merchant`

    (f) `hasUnusualRouteReport(ship1) = true`

    (g) *`areMeeting(ship1, ship2) = true`*

    (h) *`isJettisoningCargo(ship1) = true`*

Figure 5.26 presents the SSBN network generated from scenario 3 and as expected the probability of both `isShipOfInterest(ship1) = true` and `hasTerroristPlan(ship1) = ExchangeIllicitCargoPlan` are 94.44% and 93.00%, respectively.

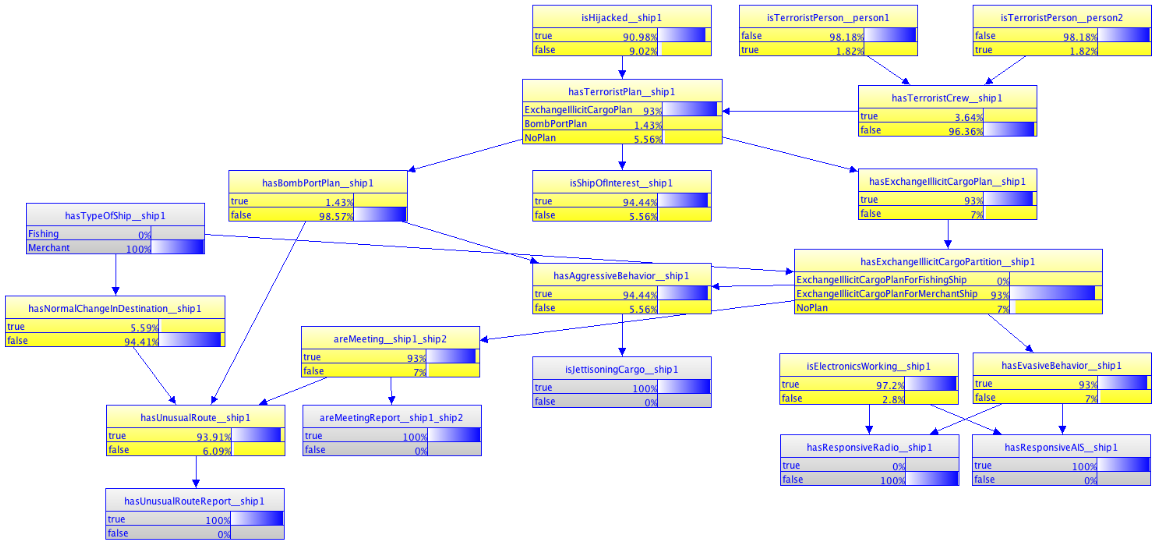


Figure 5.26: SSBN generated for the third scenario.

5.2.3 Third Iteration

While the original model considered whether a person is related to a terrorist or is part of a terrorist organization, this iteration focuses on determining whether a person *is* a terrorist. Ethical aspects excluded, creating a profile of a terrorist from the available merchant population reduces the volume of individuals requiring further investigation by limited analytic resources. The idea is to infer an individual crew member's terrorist affiliation given his close relations, group associations, communications, and background influences. The work presented in this Subsection is based on the work of Haberlin and Costa [55], which depicts a BN for this domain, and Carvalho *et al.* [18], which difines a probabilistic ontology based on the BN model from [55].

Literature on the subject reveals several models that sought to map the terrorist social network using social network analysis and some method of probabilistic inference. Using automation to identify interconnections between terrorists can reduce operator workload. Yang and Ng constructed a social network from weblog data gathered through topic-specific exploration [135]. Similarly, Coffman and Marcus performed social network analysis through pattern analysis to classify the roles of actors within a network using communication data [25]. Dombroski and Carley propose a hierarchical Bayesian inference model to produce a representation of a network's structure and the accuracy of informants [38]. Krebs has mapped a terrorist network topology from open-sources following the 9/11/2001 attacks and introduced a model representing the degrees of separation in Al Quaida leadership [73]. In a few cases, these network analyses were taken a step further and used to evaluate effects of friendly force courses of action, effects of removing particular individuals, and predicting attacks based on patterns of activity. Wagenhals and Levis used a timed influence net to add a temporal component to a model with terrorists embedded in a society that is supporting them to describe desired and undesired effects to both the adversary and local population caused by friendly forces [132]. Moon and Carley linked social and spatial relations to predict the evolution of a terrorist network over time, and posit the effect of "isolating" particular individuals within the network [95].

These models all concern groups, their members, and linkages. Our third iteration has the goal of applying high-level fusion by combining information about relations, group affiliations, communications, and ethno-religious or political background into a model describing the likelihood that a particular individual becomes a terrorist. This extends the overall high-level fusion MDA PO developed so far.

Requirements

The main goal is to identify the likelihood of a particular crew member being a terrorist. Specific statistics were not available in open-source material so the model assumes 0.001 percent of the target demographic to be involved in terrorism, and expands the query "Does the ship have a terrorist crew member?" as follows (same typing convention applies):

1. *Does the ship have a terrorist crew member?*
   (a) *Is the crew member associated with any terrorist organization.*
   (b) Has the crew member been negatively influenced in some way by his/her personal

history?

   i. Verify if the crew member has direct knowledge of someone either detained or killed by coalition forces during the conflict;
   ii. Verify if the crew member is married.

(c) Has the crew member been using communications media frequently used by terrorists?

   i. Verify if the crew members uses cellular communication;
   ii. Verify if the crew members uses chat room communication;
   iii. Verify if the crew members uses email communication;
   iv. Verify if the crew members uses weblog communication;

(d) Is the crew member a potential terrorist recruit?

   i. *Verify if the crew member is related to any terrorist;*
   ii. Verify if the crew member has friendship with any terrorist.

(e) Is the crew member associated with any of the four primary terrorist cliques introduced by Sageman who are operating in the Middle East, North Africa and Southeastern Asia [116]?

   i. Verify if the crew member is a professional, semiskilled, or unskilled laborer;
   ii. Verify the education level of the crew member;
   iii. Verify if the crew member is from the upper, middle, or lower class;
   iv. Verify the nationality of the crew member.

Analysis and Design

This stage formally defines the model semantics captured in the UML model. Table 5.4 presents a two step approach to identifying the major entities, their attributes, and relationships. Initially, the requirements are the main source for keywords representing concepts to be defined in the ontology (*e.g.*, highlighted text in Table 5.4). Then, the chosen keywords are grouped in a logical manner, *e.g.*, grouping attributes with the entities possessing them

Table 5.4: A simple method for identifying entities, attributes, and relationships.

...Does the ship have a terrorist crew member? ...Is the crew member associated with any terrorist organization. ...Verify if the crew member is married. ...Verify if the crew members uses cellular communication; ...	`Ship`
	`-hasCrewMember`
	
	`Person`
	`-isMemberOfOrganization`
	`-isMarried`
	`-usesCellularCommunication`

(see simple grouping on the second column). Although not shown here for brevity, this method was used for the analysis and design of all the requirements in this iteration. The resulting attributes, relationships and their grouping for the entities `Person` and `Organization` is shown in Table 5.5.

These three iterations are meant to illustrate the probabilistic definitions of the ontology, and thus reflect just the initial steps in building a full model. Further analysis of the terms listed in Table 5.5 will show that other entities are necessary to encode the MDA complete semantics. For instance, the `Country` entity is needed to express the relationship that `Person hasNationaly some Country`. The next step is to understand the domain rules, making use of the concepts identified so far to achieve the goals elicited during the requirements stage. The following rules, already grouped in fragments, were identified after a review of the open source literature available (same typing convention applies):

1. Terrorist organization grouping;
   (a) *If a crew member is a member of a terrorist organization, then it is more likely that he is a terrorist;*
   (b) *If an organization has a terrorist member, it is more likely that it is a terrorist organization.*
2. Background influence grouping;

(a) For those who are terrorists, 100% of them chose to do so because of something in their past. That is, no one was born a terrorist, or just woke up one day and decided to be a terrorist. That is the easy case. For those who are not, 20% chose not to become terrorists despite having some possible factor in their background and 80% chose not to become a terrorist possibly because they have never been exposed[7].

(b) An individual is usually negatively affected (leads him/her in becoming a terrorist) by having direct knowledge of someone either detained or killed by coalition forces during the conflict;

(c) In the geographic area of interest, an estimated 2% of the population knows someone who was killed as a result of OEF/OIF [94];

(d) In the geographic area of interest, approximately 2% of the population knows someone detained as a result of coalition operations [94];

(e) Contrary to common perception, terrorists are predominantly married in keeping with the teachings of the Quran [116]. And about half of the general population in the target demographic is married.

3. Communication grouping;

(a) It is possible that a crew member may communicate with a terrorist without being involved in terrorism due to non-terrorist affiliations or other relationships that have some normal expectation of interaction;

(b) For each of the internet communications paths there is also the background usage rate of 28.8% in the Middle East [5]. Because the data is not broken down for the three internet transmission paths, this probability was applied equally to chat room, email, and weblog methods of communication;

(c) Similarly, cellular telephone usage among the general population is assumed to be 31.6% based on Egyptian subscriber rates [4];

[7]This rule and explanation was given by the SME.

(d) Given the availability of cellular technology and subscribing to the prioritization, a probability of 90% is assigned to terrorists communicating using cellular telephones;

(e) The transient nature and unfettered availability of chat room communications makes it appealing to individuals who desire to remain nameless. A probability of 85% is assigned to terrorists communicating through chat rooms;

(f) Email is the least desirable form of communication because it requires some form of subscriber account. Even in the event that fictitious information is used in creating such an account, an auditable trail may lead determined forces to the originator. Still, it is a versatile means of communication and is assigned a probability of 65% for terrorists;

(g) The anonymity associated with weblog interaction is very appealing to terrorists. This path is similar to chat room communications, but is less transient in content and can reach more subscribers simultaneously. For these reasons, a probability of 80% is assigned to weblog communications.

4. Relationship grouping;

   (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;

   (b) Research shows that if a crew member has a relationship with terrorists, there is a 14% chance that he is related to a terrorist.

5. Cluster grouping;

   (a) It is assumed that all active terrorists fall into one of the terrorist cliques or their subsidiaries described by Sageman [116];

   (b) Contrary to popular thought, terrorists tend to not be unskilled drifters with no options other than martyrdom;

Table 5.5: Grouping for entities, attributes, and relations in third iteration.

Terrorist grouping	–hasFriendshipWithTerrorist
-Person	
–isTerrorist	Background influence grouping
-Organization	-Person
–isMemberOfOrganization	–hasInfluencePartition
–isTerroristOrganization	–hasFamilyStatus
	–hasOIForOEFInfluence
Communication grouping	–knowsPersonKilledInOIForOEF
-Person	–knowsPersonImprisionedInOIForOEF
–usesWeblog	
–usesEmail	Cluster grouping
–usesCellular	-Person
–usesChatroom	–hasClusterPartition
	–hasNationality
Relationship grouping	–hasEconomicStanding
-Person	–hasEducationLevel
–hasTerroristBeliefs	–hasOccupation
–hasKinshipToTerrorist	

(c) Many believe terrorist recruits to be uneducated simpletons who are easily persuaded by eloquent muftis who appeal to their sense of honor and perception of persecution. In fact, the data indicate that the typical terrorist is more educated than the average global citizen and is by far more educated than the average citizen in the Middle East, North Africa, and Southeastern Asia regions [116];

(d) Terrorist from the clusters described by Sageman [116] are less likely to be of lower class than other people from that demographic area.

Given the extensive research previously done, it was possible to assert some probability values when elaborating these rules during the Analysis & Design stage, whereas in previous iterations probabilities were defined only during the Implementation stage. Usually, only imprecise statements are used in these conditional rules (*e.g.*, more likely, less likely, rare, etc).

Implementation

Appropriate assumptions are needed to accommodate available data without compromising the utility of the model. First, a terrorist will communicate with other terrorists with certainty, but there is variability on the communication path used. Also, an individual might communicate with terrorists inadvertently. Next, there is 0.1% chance that any random person in the target demographic is a terrorist, which drives the coincidental interaction between a honest crew member and someone who may happen to be a terrorist without his knowledge. Further, the target area (Middle East, North Africa and Southeast Asia) enables using the cluster organizations introduced by Sageman [116] as basis for the groups in the association partition. Other attributes within this partition are compiled given the individual's participation in one of those groups. Additionally, a crew member could be involved with a terrorist organization other than the four identified, and that would negatively affect the outcome since he would be grouped with non-terrorists. However, it is likely that smaller groups are splinters from one of these major clusters and could therefore be included in the analysis under their super-group. Finally, in its current form, the model only captures the influence of Operation Enduring Freedom (OEF) and Operation Iraqi Freedom (OIF) and marital status in the crew member's background. Figure 5.27 presents the last MFrags changed/added to the MTheory for Domain Maritime Awareness (see Figure 5.23 for previous MFrags).

Appendix B Subsection B.2.3 presents the details and explanations of all MFrags and all resident nodes and their respective LPDs of the probabilistic ontology discussed in this Subsection.

Test

Again, although I have described many different types of evaluation and tests we can perform in our model in Subsection 5.1.4, this iteration will focus on performing integration test based on case-based evaluation, as was the case in the first and second iterations.

As explained in Subsection 5.1.4 it is important to try out different scenarios in order to

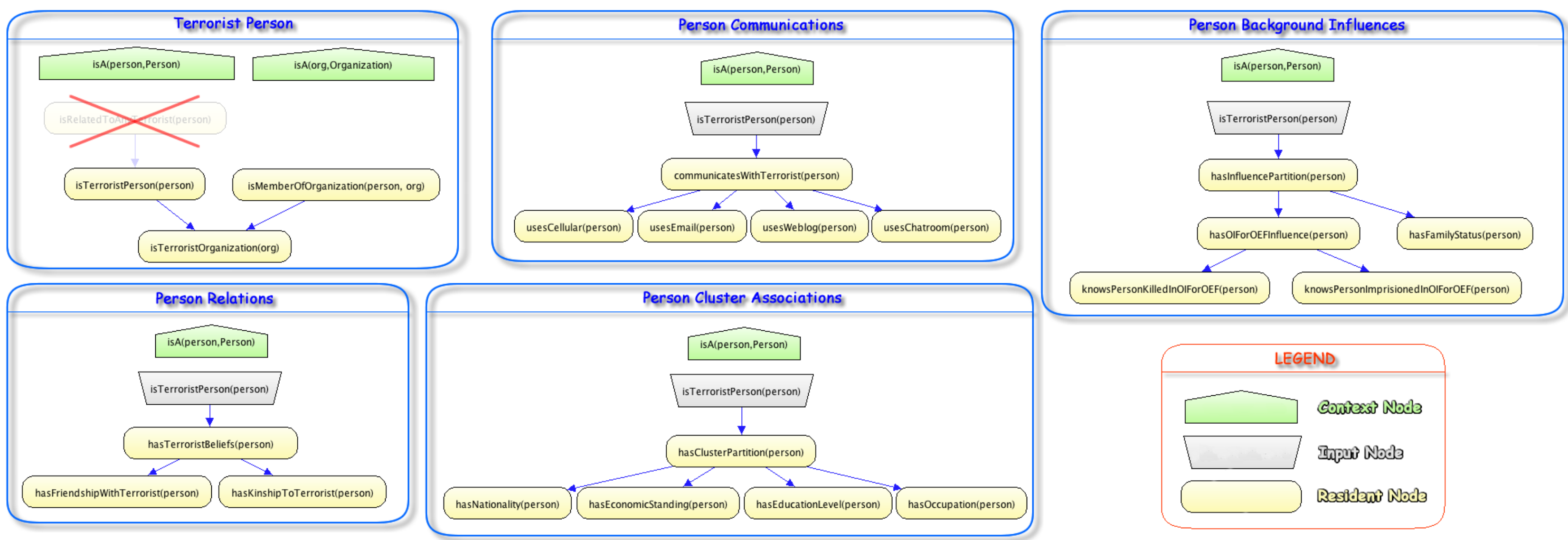


Figure 5.27: MFrags changed/added in third iteration.

capture the nuances of the model. In a serious test of the model, we would have to model a lot scenarios in order to cover at least the most important aspects of our requirements. However, I define only three qualitatively different scenarios in order to illustrate the mechanics of defining and testing a scenario. The first is a general case in which an individual fits a profile and can therefore be "correctly" identified. In the second and third cases, situations are introduced in which individuals could be incorrectly profiled using these techniques.

In the first scenario ("obvious" guilty), Bakari, a student at Misr University in Cairo and member of a terrorist organization, has been tasked with smuggling explosive materials into the United States for use in making improvised explosive devices (IED). He is from a middle-class Egyptian family with a large extended family, including one uncle who is a member of the Mojahedin-e Khalq Organization. Because he is a full-time student, he has not had the opportunity to earn enough money for a suitable dowry and is still single. Recently, postings on a terrorist-related weblog have been attributed to Bakari's school account, in which he laments his colleagues he watched being taken prisoner by the coalition.

Figure 5.28 shows that with all the information above the probability that Bakari is involved in terrorism is 72.59%, primarily due to the weblog communications and affiliation of his uncle. Removing the communications link drops him all the way down to 48.85%. Including communications activity and removing the uncle affiliation drops his percentage to 3.07%. It is clear that being related to and communicating with terrorists will flag an individual very significantly as a terrorist candidate.

These results are taken into consideration in new iterations. Usually, LPDs are adjusted in order to make probabilities that were too high go down, and vice-versa. Although it is not described here, this was done a few times during this modeling process. In fact, before this model can be deployed, it should go through a few more iterations of adjusting the distributions.

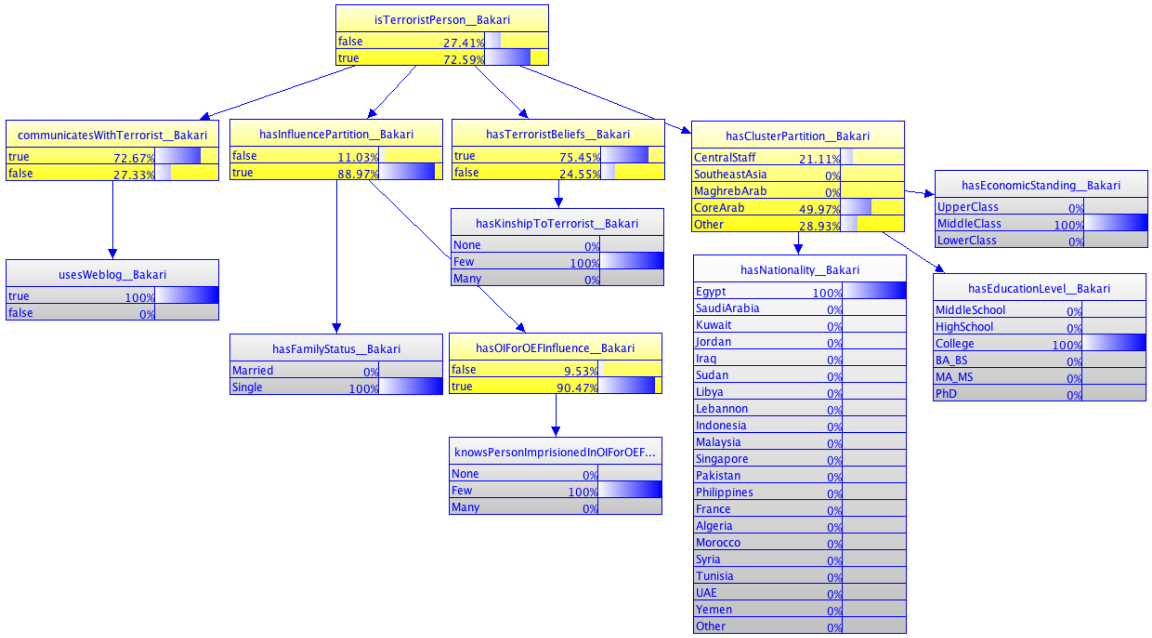


Figure 5.28: SSBN generated for scenario 1.

Also of note is effect of the Influence Partition on the outcome of this case. The scenario introduced information about Bakari's marital status, and this has very little effect. Removing the marital status results in a probability of being a terrorist of 76.27%. This value is higher, because the "standard" terrorist profiled requires an individual to be married, not single. Knowing someone imprisoned has a greater effect and removal of this information reduces the overall terrorism likelihood to 35.33%.

It is clear from this case study that family and friend relationships weigh heavily on the determination of terrorist activity. In the case where an individual has a casual or coincidental relationship with someone involved, or there is a case of name-based mistaken identity, this would likely lead to an incorrect determination. Ranking the partitions from most influential to least gives an ordering of Relationship, Communication, Influence, and Cluster.

In the second scenario (guilty who looks innocent), Arif leaves his Indonesian village at age 17 to provide for his family through life as a merchant sailor. He is an unmarried, unskilled worker who did not complete high school. While looking for work as a mariner in Jakarta, he shared a room with 5 others, at least one of whom has become involved with the Jemaah Islamiyah organization. Arif joins his friend at a Jemaah Islamiyah meeting where he is given a cell phone and contact information.

In this case, Arif is involved in the beginning stages of the terrorist recruitment process. While his background has none of the profile indicators, his growing affiliation and recruitment will eventually lead him to a positive assessment. It is nearly impossible to force a positive likelihood onto the crew member being a terrorist by switching features in the Influence, Communications, and Relationship partitions. This is due to the fact that the cluster partition has driven the model to an unlikely terrorist character in the "Other" category (see Figure 5.29). Since he does not fall into one of the terrorist cliques, it will be difficult to identify him as a terrorist. His background does not fit with the classic profile.

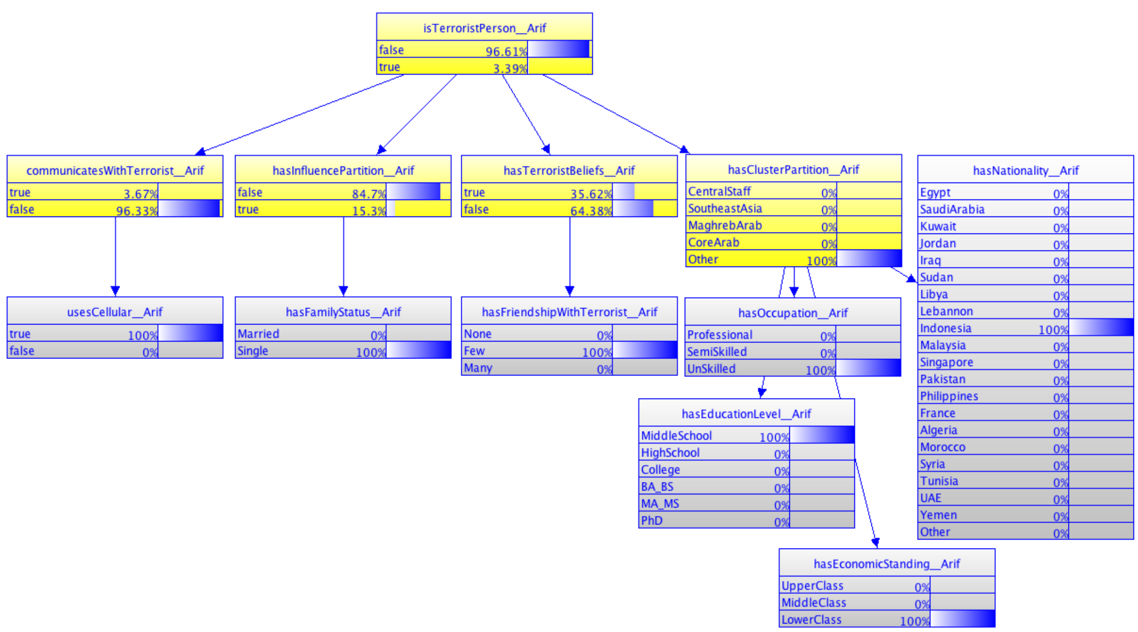


Figure 5.29: SSBN generated for scenario 2.

This scenario demonstrates a weakness of the model and intelligence collection in general.

Profiles are built on history, but cannot account for rapid transition from one social group to another. Arif arrived in Jakarta as a farmer looking for work and through rapid social affiliation became a terrorist suspect. The unknown question is whether he will continue to grow his relationship with Jemaah Islamiyah, or turn toward life as a commercial seaman.

In the third scenario (innocent who looks guilty), Irasto leaves Amman, Jordan to earn a living as a merchant sailor. He comes from a middle class family and began studies at the University of Jordan before local violence frightened him into leaving. While in school, several of his friends were detained by coalition forces under suspicion of terrorist affiliation and have not been seen since. He frequently communicated with them by email and cell phone prior to their disappearance.

The unknown status of Irasto's friends muddies the waters for this scenario. They were detained as part of OEF/OIF, and therefore affect the Influence Partition, but we have no information as to whether these friends were actually confirmed to be terrorists. If the safe route is taken (from an intelligence perspective) they will be considered terrorists and Irasto will also be pronounced a terrorist with a likelihood of 89.92%; without this assumption the probability drops to 3.44%. These are the worst and best case, respectively. However, if we considered the likelihood that his friends are terrorists, then we would obtain a probability between those two numbers (the extreme cases).

In this iteration we are simplifying this friendship relationship. In fact, `hasFriendshipWithTerrorist` is logically equivalent to the existentially quantified RV saying there exists x such that x is a terrorist and x is friends with Irasto. This RV is a built-in RV in MEBN. However, due to UnBBayes limitation, we are considering this existential operation is done outside the model and we just receive the result (`None`, `Few`, or `Many`). The problem with this approach is that if we want to infer the likelihood, for instance, that Irasto's friends are terrorists using our model, their probability will not influence Irasto's probability of being a terrorist, since there is no connection between the node `hasFriendshipWithTerrorist` and Irasto's actual friends. In future iterations this has to be dealt with.

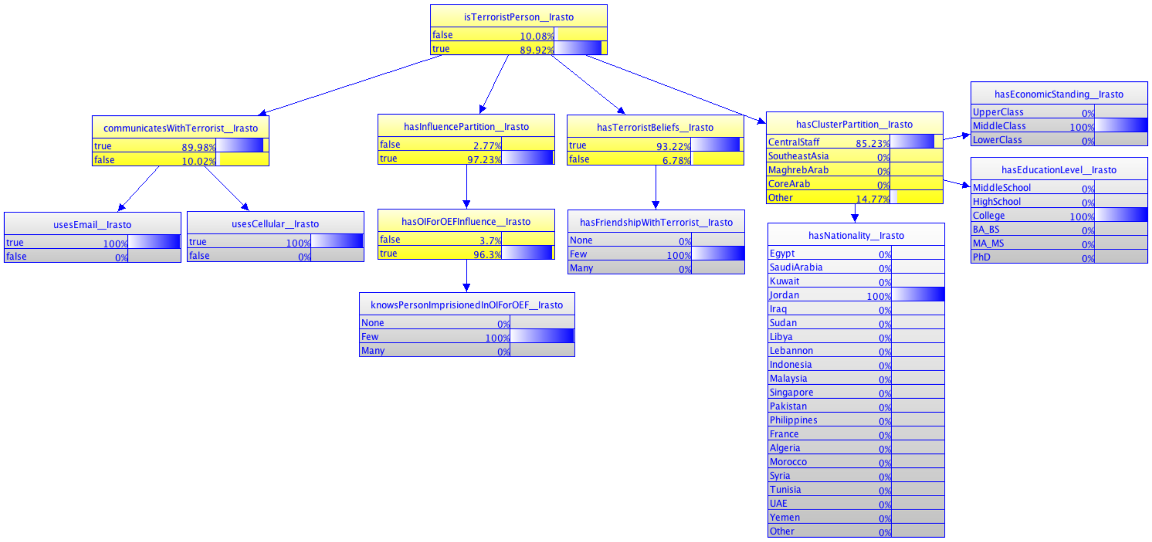


Figure 5.30: SSBN generated for scenario 3.

The model indicates that Irasto appears to be involved with either the Cental Staff or Maghreb Arab clique. This drives the Relationship partition into strongly affecting the overall likelihood. The same dilemma exists for communications. Irasto communicated with his friends using two of the profiled communication paths. If those friends are determined to be terrorists, then his likelihood jump significantly over what it would be if they are not. The model recognizes guilt by association. These two particulars illustrate some of the problems introduced when intelligence is not shared between organizations. If the analysts have access to the final determination of his friends, Irasto will be more likely to have a correct determination of guilt or innocence.

5.2.4 Testing the Final MDA PO

One of the major challenges in systems like PROGNOS is evaluating the situational awareness and prediction generated by its probabilistic model.

The literature in Machine Learning, Artificial Intelligence, and Data Mining usually work with real data by separating it into training and testing sets. However, in systems that try to predict rare events, such as terrorist attacks, either there is not enough data or the data available is classified. Therefore, in these cases, there is not sufficient data to be used for testing.

To overcome this limitation, a common approach is to create different use cases or scenarios manually. This use case generation process is discussed in Subsection 5.2.4. However, this is a tedious process and usually not statistically sufficient to confidently assess how good these probabilistic models are. In Subsection 5.2.4 we present a framework that can be used for automatically creating different and random, yet consistent, scenarios to provide sufficient statistical data for testing. It is to be stressed, however, that this testing is only as good as the use cases incorporated into the testing process, and there is no substitute for real-world evaluation. It is important to notice that although this Subsection focuses on case-based evaluation, this should not be the only test done in the final model. In fact, Subsection 5.1.4 presents all the tests that should be performed in the Test discipline.

Creating scenarios manually

In the first iteration the main goal is to identify if a ship is of interest, *i.e.*, if the ship seems to be suspicious in any way. The assumption in this model is that a ship is of interest if and only if there is at least one terrorist crew member.

The following iterations provide clarification on the reasons behind declaring a ship as being of interest and detects an individual crew member's terrorist affiliation given his close relations, group associations, communications, and background influences.

To test this final probabilistic model, let's define 4 major scenarios:

1. a possible bomb plan using fishing ship;
2. a possible bomb plan using merchant ship;
3. a possible exchange illicit cargo using fishing ship;
4. a possible exchange illicit cargo using merchant ship.

For each of these major scenarios let's create 5 variations:

1. “sure” positive;

2. “looks” positive;

3. unsure;

4. “looks” negative;

5. “sure” negative.

All 20 different scenarios were analysed by the SMEs and were evaluated as reasonable results (what was expected). Figure 5.31 presents part of the SSBN generated for a scenario where a merchant ship is exchanging illicit cargo and the evidence makes it obvious to detect that this is the case. In order to be concise, and since the focus is on the automatic testing presented in Subsection 5.2.4, this will be the only scenario presented.

Figure 5.31: Part of the SSBN generated for “sure” positive of a possible exchange illicit cargo using merchant ship.

Creating scenarios automatically

Besides being a tedious process, there are a few problems with the manual creation of scenarios as presented in Subsection 5.2.4 and in the scenarios tested for each individual iteration. In the first set of scenarios created for the first iteration, it is clear that the test designers just tested how well the model behaves when all the evidence is in favor of the hypotheses being tested. However, how will the model behave if we receive evidence both in favor of and against the hypotheses being tested? Is it still a good model in these cases?

In fact, this is a problem that the last set of scenarios presented in Subsection 5.2.4 addresses. This, the fact that some evidence favors the hypotheses and some does not, is why there are scenarios where the expected result is "looks" positive, "looks" negative, and unsure. However, even twenty different scenarios is not enough considering the amount of information that is used as evidence in the final model. Let's clarify by presenting the numbers. In the final model there are more than 20 evidence nodes with at least 2 states each (some have more than 10 states). This gives more than $2^{20} = 1,048,576$ different configurations of evidence. In other words, while we tried to cover different types of scenarios, 20 is still an extremely small number compared with the possible configurations. However, it is unreasonable to think a human being will be able to generate and analyze more than one million different scenarios. For this reason, we created a framework for simulating different scenarios automatically for the PROGNOS project [28].

There are three basic steps in our simulation framework:

1. Create entities and generate some basic static ground truth for them (*e.g.*, create ships, define their type, and their plan);

2. Generate dynamic data for entities based on their static ground truth data (*e.g.*, if the ship is a fishing ship and it has a normal behavior, it will go from its origin port to its fishing area and after some time it will go to its destination port);

3. Simulate reports for different agencies. Each agency has a probability of generating a correct report (*e.g.*, saying a person is from Egypt when he is actually from Egypt),

an incorrect report (*e.g.*, saying a person is not from Egypt when he is in fact from Egypt), and no report at all (*e.g.*, not being able to say where a person is from). The idea is that different agencies are expected to be more accurate in certain types of information than others (*e.g.*, the Navy is expected to have more accurate data on a ship's position than the FBI).

The information generated in the first two steps are considered the ground truth, while the reports generated in the third step is given as input to the probabilistic model, like the MDA PO described in this Section. The probabilistic model can then use these information as evidence to provide situational awareness and prediction after the reasoning process through its posterior probabilities. Once we know what the model "thinks" is more reasonable (*e.g.*, if a ship is of interest), we can ask the simulation for the correct information, *i.e.*, the ground truth with respect to the hypotheses being tested (*e.g.*, if the ship is indeed of interest). We can then evaluate if the model provided a correct result. Since this process is automatic, we can run this evaluation process as many times as we need to and finally compute some metrics (*e.g.*, confusion matrix) to evaluate how well our model performs. Furthermore, a subset of these generated scenarios should be selected in order to be presented to the SMEs to determine whether the results are reasonable.

Table 5.6: Number of TP, FP, TN, and FN.

Real/Inferred	**T**	**F**
T	24	3
F	11	577

To test the final MDA PO, I ran the simulation with 615 ships, where 27 of them were ship of interest and 588 were regular ships with no terrorist plan. Tables 5.6 and 5.7 present the confusion matrix for the node isShipOfInterest(ship). Table 5.6 presents the number of ships while Table 5.7 presents the probability of true positive (TP), false positive (FP), true negative (TN), and false negative (FN). As It can be seen, the percentage of misclassifications of ships of interest was small, only 3 in 27, *i.e.*, only 11.11%.

Table 5.7: Percentage of TP, FP, TN, and FN.

Real/Inferred	**T**	**F**
T	88.89%	11.11%
F	1.87%	98.13%

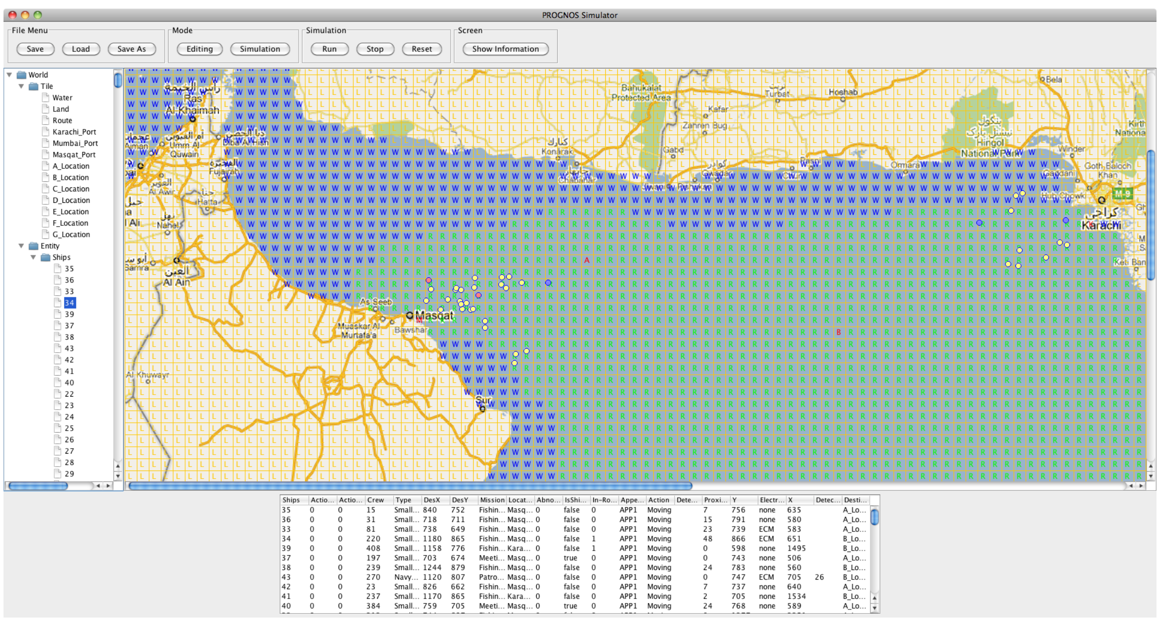


Figure 5.32: Simulation editor.

In the case of the PROGNOS evaluation, the subject matter experts who evaluated the use cases also supported the domain knowledge engineering effort. A more rigorous evaluation process would make use of independent subject matter experts who had not been involved in the system design process. These independent evaluators would develop use cases for evaluation and rate the quality of the system's output.

However, to be able to compute the three steps described above, we need to define some basic characteristics of the simulation. For instance, what is the geographical region considered, which cells correspond to land and which correspond to sea, where are the ports of interest, what are the usual routes between areas of interest, where are the common fishing areas, etc. Figure 5.32 presents the simulation editor used to define this information.

Appendix B: Use Cases Implementation Details

In this Appendix I will give the details of the probabilistic ontologies presented in Chapter 5.

B.1 Probabilistic Ontology for Procurement Fraud Detection and Prevention in Brazil

All the assumptions for the RVs created and for defining their LPD will be described for every MFrag designed for the MTheory that represents the PO for the Procurement Fraud Detection and Prevention implemented. In each MFrag, the resident RVs are shown as yellow rounded rectangles; the input RVs are shown as gray trapezoids; the context RVs are shown as green pentagons.

In order to make reference easier, the rules defined during Analysis & Design in Chapter 5 Subsection 5.1.2 will be repeated here. The rules are:

1. If a member of the committee has a relative (mother, father, brother, or sister) responsible for a bidder in the procurement, then it is more likely that a relation exists between the committee and the enterprises, which inhibits competition.

2. If a member of the committee lives at the same address as a person responsible for a bidder in the procurement, then it is more likely that a relation exists between the committee and the enterprises, which lowers competition.

3. If a contract of high value related to a procurement has a responsible person of the winner enterprise with low education or low annual income, then this person is likely to be a front for the firm, which lowers competition.

4. If the responsible person of the winner enterprise is also responsible for another enterprise that has its CGC suspended for procuring with the public administration, then this procurement is more likely to need further investigation.

5. If the responsible people for the bidders in the procurement are related to each other, then a competition is more likely to have been compromised.

6. If 1, 2, 3, or 5, then the procurement is more likely to require further investigation.

7. If a member of the committee has been convicted of a crime or has been penalized administratively, then he/she does not have a clean history. If he/she was recently investigated, then it is likely that he/she does not have a clean history.

8. If the relation defined in 1 and 2 is found in previous procurements, then it is more likely that there will be a relation between this committee and future bidders.

9. If 7 or 8, then it is more likely that the committee needs to be changed.

In order to facilitate the understanding of the MFrags in this model, it is useful to know the dependence between the MFrags.

The MFrags with no dependency are:

1. Personal Information;
2. Procurement Information;
3. Enterprise Information; and
4. Judgement History.

The other MFrags have the following dependence:

1. Front of Enterprise
   (a) Procurement Information; and
   (b) Personal Information.
2. Exists Front in Enterprise
   (a) Enterprise Information; and

    (b) Front of Enterprise.

3. Related Participant Enterprises
    (a) Procurement Information;
    (b) Enterprise Information; and
    (c) Personal Information.
4. Member Related to Participant
    (a) Personal Information;
    (b) Procurement Information; and
    (c) Enterprise Information.
5. Competition Compromised
    (a) Procurement Information;
    (b) Exists Front in Enterprise;
    (c) Related Participant Enterprises; and
    (d) Member Related to Participant.
6. Related to Previous Participants
    (a) Personal Information;
    (b) Procurement Information; and
    (c) Enterprise Information.
7. Suspicious Committee
    (a) Procurement Information;
    (b) Judgment History; and
    (c) Related to Previous Participants.

8. Owns Suspended Enterprise

   (a) Enterprise Information.

9. Suspicious Procurement

   (a) Procurement Information;

   (b) Enterprise Information;

   (c) Competition Compromised;

   (d) Owns Suspended Enterprise; and

   (e) Suspicious Committee.

The MFrags will be presented from the less dependent to the more dependent. *I.e.*, an MFrag will only be described after all its dependent MFrags have been described. It is consistent with the order of the dependency explanations given previously. Also, all the LPDs defined in this PO are notional only. No real data or statistics was used. Therefore, before this model can be used in production, a few more iterations are necessary in order to make sure these notional probabilities are correct.

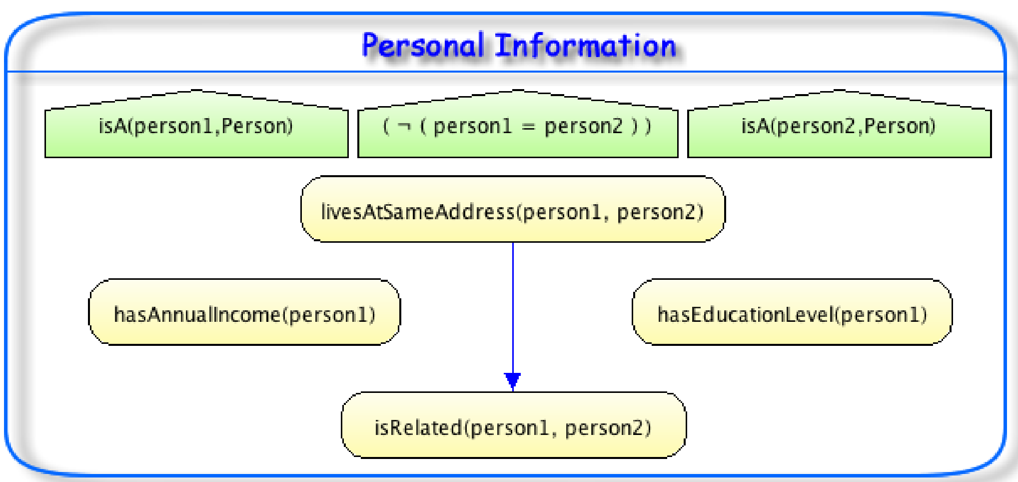


Figure B.1: MFrag Personal Information.

Figure B.1 presents the Personal Information MFrag. In it we have RVs associated to the class `Person`. Listings B.1, B.75, B.3, and B.4 present the LPDs for the RVs, `hasAnnualIncome(person1)`, `hasEducationLevel(person1)`, `livesAtSameAddress(person1, person2)`, and `isRelated(person1, person2)`, respectively. The assumptions behind these LPDs are that a person is more likely to have a lower income, the most people have either middle school or high school education, two random people rarely live at the same address, and if two people live at the same address, they are more likely to be related[1].

Listing B.1: LPD for `hasAnnualIncome(person)`

```
[
   Lower10k = .4,
   From10kTo30k = .3,
   From30kTo60k = .2,
   From60kTo100k = .09,
   Greater100k = .01
]
```

Listing B.2: LPD for `hasEducationLevel(person)`

```
[
   NoEducation = .1,
   MiddleSchool = .4,
   HighSchool = .3,
   Undergraduate = .15,
   Graduate = .05
]
```

Listing B.3: LPD for `livesAtSameAddress(person1, person2)`

```
[
   true = .0001,
   false = .9999
]
```

[1]Notice that although it is reasonable to think that the education level of a person influences this person's annual income, in this iteration the PO is not modeled this way. This is not a major issue because we will usually have both information available. However, in future iterations this dependence should be modeled or at least taken into consideration.

Listing B.4: LPD for `isRelated(person1, person2)`

```
if any person1.person2 have ( livesAtSameAddress = true ) [
   true = .9,
   false = .1
] else [
   true = .001,
   false = .999
]
```

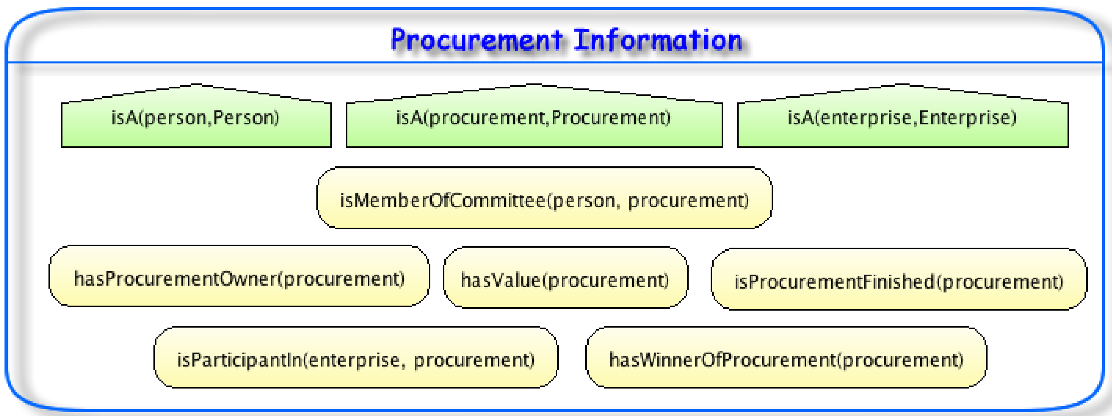


Figure B.2: MFrag Procurement Information.

Figure B.2 presents the Procurement Information MFrag. In it we have RVs associated to the class `Procurement`. Listings B.5, B.6, B.7, and B.8 present the LPDs for the RVs, `isMemberOfCommittee(person, procurement)`, `hasValue(procurement)`, `isProcurementFinished(procurement)`, and `isParticipantIn(enterprise, procurement)`, respectively. The assumptions behind these LPDs are that a random person is rarely a member of a committee and a random enterprise is rarely a participant in a procurement. All other RVs assume a uniform distribution, including `hasProcurementOwner(procurement)` and `hasWinnerOfProcurement(procurement)`, which are uniform over all possible instances of `PublicAgency` and `Enterprise`, respectively.

Listing B.5: LPD for `isMemberOfCommittee(person, procurement)`

```
[
   true = .0001,
   false = .9999
]
```

Listing B.6: LPD for `hasValue(procurement)`

```
[
   Lower10k = .2,
   From10kTo100k = .2,
   From100kTo500k = .2,
   From500kTo1000k = .2,
   Greater1000k = .2
]
```

Listing B.7: LPD for `isProcurementFinished(procurement)`

```
[
   true = .5,
   false = .5
]
```

Listing B.8: LPD for `isParticipantIn(enterprise, procurement)`

```
[
   true = .0001,
   false = .9999
]
```

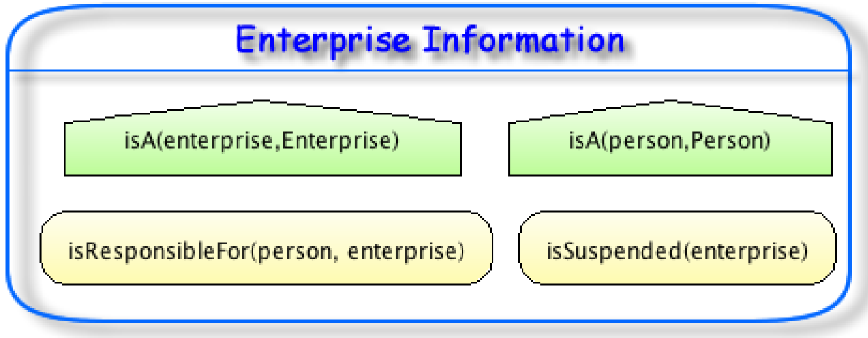


Figure B.3: MFrag Enterprise Information.

Figure B.3 presents the Enterprise Information MFrag. In it we have RVs associated to the class `Enterprise`. Listings B.9 and B.10 present the LPDs for the RVs, `isResponsibleFor(person, enterprise)` and `isSuspended(enterprise)`, respectively. The assumptions behind these LPDs are that a random person is rarely a the owner of an enterprise and a random enterprise is rarely suspended from bidding in public procurements.

Listing B.9: LPD for `isResponsibleFor(person, enterprise)`

```
[
   true = .0001,
   false = .9999
]
```

Listing B.10: LPD for `isSuspended(enterprise)`

```
[
   true = .0001,
   false = .9999
]
```

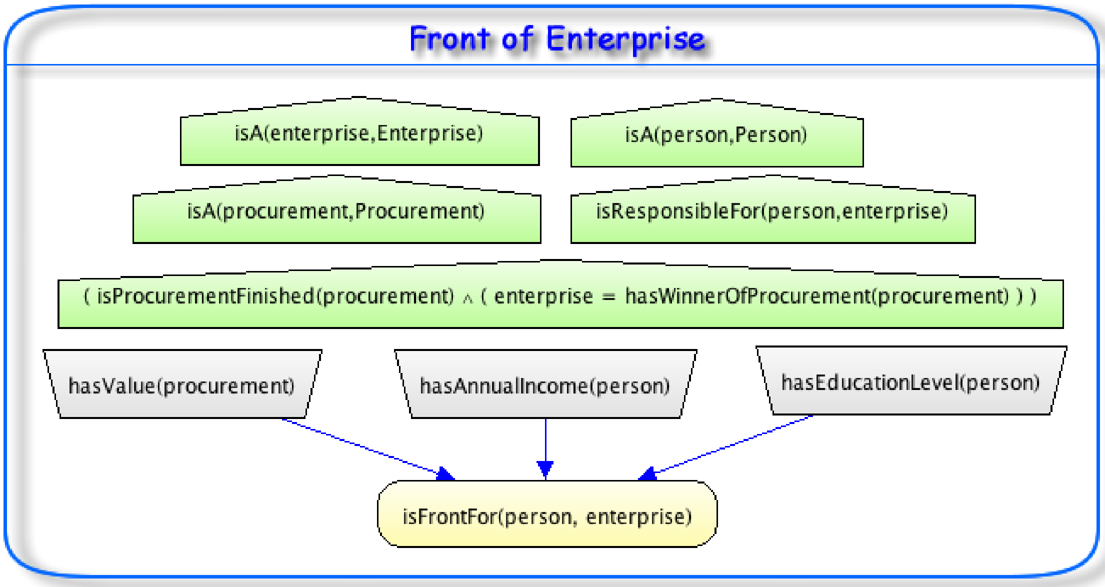


Figure B.4: MFrag Front of Enterprise.

Figure B.4 presents the Front of Enterprise MFrag. Listings B.9 presents the LPD for the RV `isFrontFor(person, enterprise)`. The assumption behind this LPD is that if the enterprise won a procurement of high value, but the owner of the enterprise does not make a lot of money and/or does not have a high education level, then this person is more likely to be a front for this enterprise.

Listing B.11: LPD for `isFrontFor(person, enterprise)`

```
if any procurement have ( hasValue = From100kTo500k ) [
   if any person have ( hasAnnualIncome = Lower10k | hasEducationLevel =
       NoEducation ) [
      true = .9,
      false = .1
   ] else if any person have ( hasAnnualIncome = From10kTo30k |
       hasEducationLevel = MiddleSchool ) [
      true = .6,
      false = .4
   ] else [
      true = .0001,
      false = .9999
   ]
] else if any procurement have ( hasValue = From500kTo1000k ) [
```

```
    if any person have ( hasAnnualIncome = Lower10k | hasEducationLevel = NoEducation ) [
        true = .95,
        false = .05
    ] else if any person have ( hasAnnualIncome = From10kTo30k | hasEducationLevel = MiddleSchool ) [
        true = .8,
        false = .2
    ] else if any person have ( hasAnnualIncome = From30kTo60k | hasEducationLevel = HighSchool ) [
        true = .6,
        false = .4
    ] else [
        true = .0001,
        false = .9999
    ]
] else if any procurement have ( hasValue = Greater1000k ) [
    if any person have ( hasAnnualIncome = Lower10k | hasEducationLevel = NoEducation ) [
        true = .99,
        false = .01
    ] else if any person have ( hasAnnualIncome = From10kTo30k | hasEducationLevel = MiddleSchool ) [
        true = .9,
        false = .1
    ] else if any person have ( hasAnnualIncome = From30kTo60k | hasEducationLevel = HighSchool ) [
        true = .8,
        false = .2
    ] else if any person have ( hasAnnualIncome = From60kTo100k | hasEducationLevel = Undergraduate ) [
        true = .6,
        false = .4
    ] else [
        true = .0001,
        false = .9999
    ]
] else [
    true = .0001,
    false = .9999
]
```

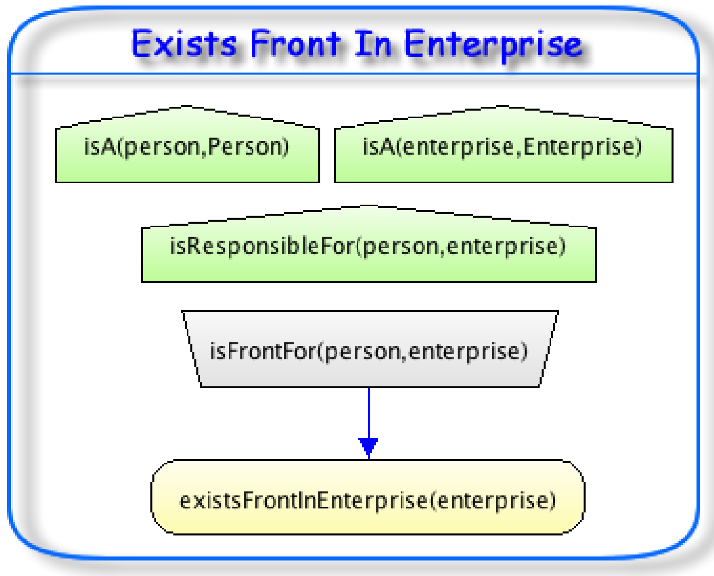


Figure B.5: MFrag Exists Front in Enterprise.

Figure B.5 presents the Exists Front in Enterprise MFrag. Listings B.12 presents the LPD for the RV `existsFrontInEnterprise(enterprise)`. The assumption behind this LPD is that if the enterprise has at least one owner that is a front, then there is a front in this enterprise. Notice this RV represents in fact an existential formula. However, due to limitations in UnBBayes' current version, this existential formula was implemented as a regular RV using the expressiveness of the LPD grammar.

Listing B.12: LPD for `existsFrontInEnterprise(enterprise)`

```
if any person.enterprise have ( isFrontFor = true ) [
  true = 1,
  false = 0
] else if all person.enterprise have ( isFrontFor = false ) [
  true = 0,
  false = 1
] else [
  true = .0001,
  false = .9999
]
```

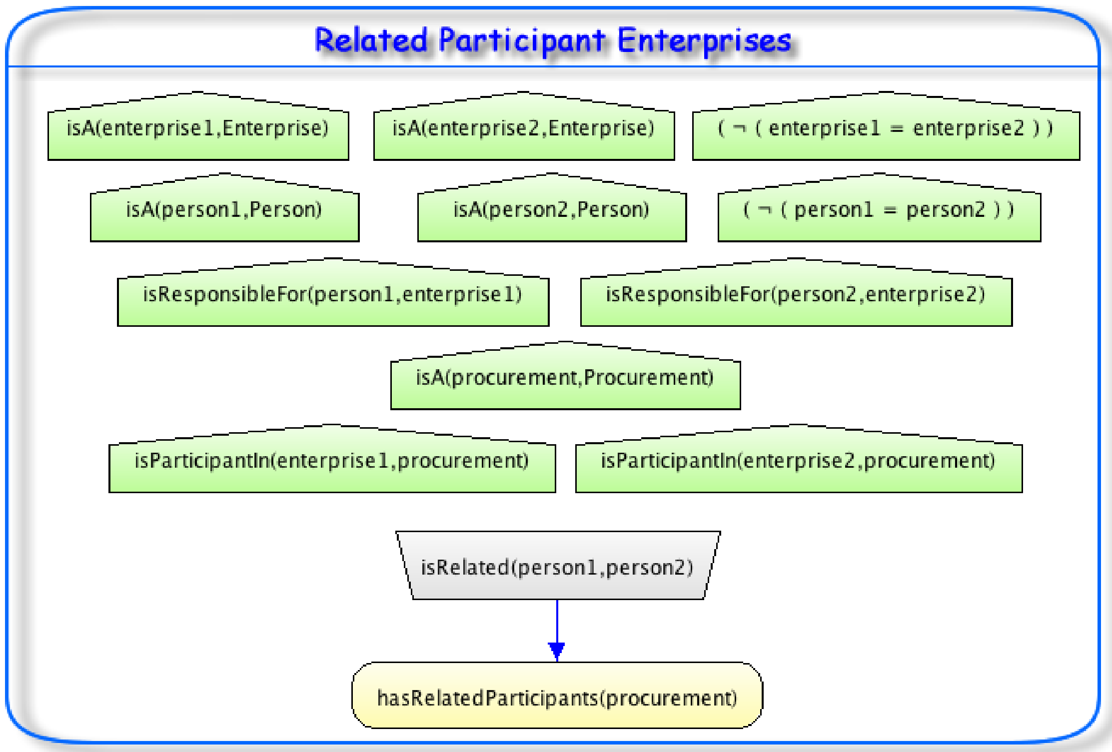


Figure B.6: MFrag Related Participant Enterprises.

Listing B.13: LPD for `hasRelatedParticipants(procurement)`

```
if any person1.person2 have ( isRelated = true ) [
   true = 1,
   false = 0
] else if all person1.person2 have ( isRelated = false ) [
   true = 0,
   false = 1
] else [
   true = .0001,
   false = .9999
]
```

Figure B.6 presents the Related Participant Enterprise MFrag. Listings B.13 presents the LPD for the RV `hasRelatedParticipants(procurement)`. The assumption behind this LPD is that if any two enterprises participating in this procurement have owners that are related, then this procurement has related participants. Notice this RV could also be represented as a formula. However, due to limitations in UnBBayes' current version, this existential formula was implemented as a regular RV using the expressiveness of the LPD grammar.

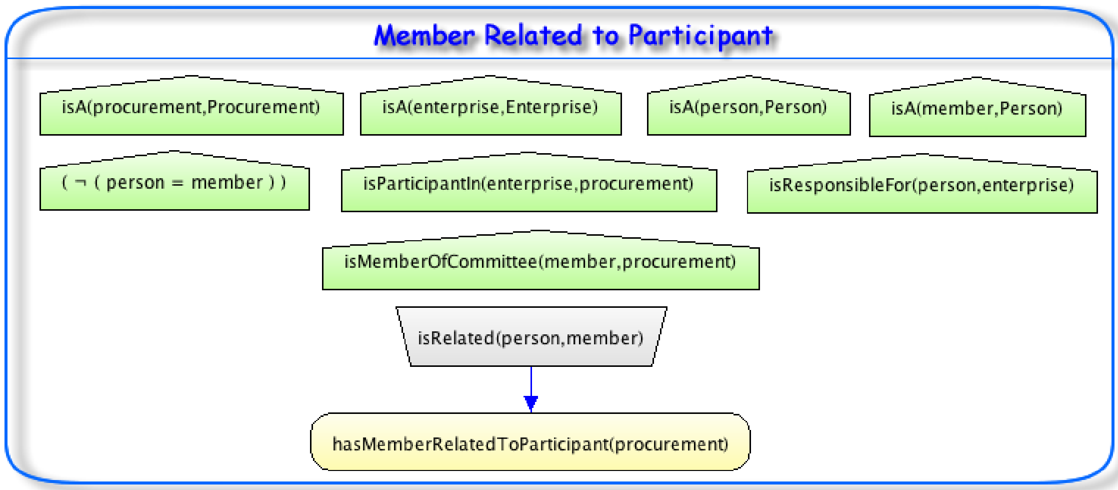


Figure B.7: MFrag Member Related to Participant.

Listing B.14: LPD for `hasMemberRelatedToParticipant(procurement)`

```
if any person.member have ( isRelated = true ) [
   true = 1,
   false = 0
] else if all person.member have ( isRelated = false ) [
   true = 0,
   false = 1
] else [
   true = .0001,
   false = .9999
]
```

Figure B.7 presents the Member Related to Participant MFrag. Listings B.14 presents the LPD for the RV `hasMemberRelatedToParticipant(procurement)`. The assumption behind this LPD is that if any member of the procurement is related to any owner of any enterprise participating in this procurement, then this procurement has a member related to a participant. Notice this RV could also be represented as a formula. However, due to limitations in UnBBayes' current version, this existential formula was implemented as a regular RV using the expressiveness of the LPD grammar.

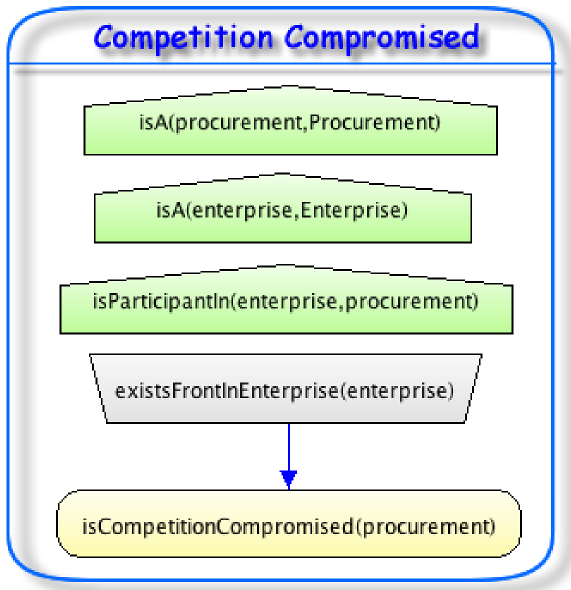


Figure B.8: MFrag Competition Compromised.

Figure B.8 presents the Competition Compromised MFrag. Listings B.15 presents the LPD for the RV `isCompetitionCompromised(procurement)`. The assumptions behind this LPD are that: if there exists a front in any of the participating enterprises, or if the participating enterprises are related, or if any member is related to any participating enterprise, then the competition is more likely to be compromised; and that if these things happen together, the probability of having competition compromised is even higher.

Listing B.15: LPD for `isCompetitionCompromised(procurement)`

```
if any procurement have ( hasRelatedParticipants = true &
    hasMemberRelatedToParticipant = true ) [
   if any enterprise have ( existsFrontInEnterprise = true ) [
      true = .9 ,
      false = .1
   ] else [
      true = .8 ,
      false = .2
   ]
] else if any procurement have ( hasRelatedParticipants = true |
    hasMemberRelatedToParticipant = true ) [
   if any enterprise have ( existsFrontInEnterprise = true ) [
      true = .8 ,
      false = .2
   ] else [
      true = .6 ,
      false = .4
   ]
] else if any enterprise have ( existsFrontInEnterprise = true ) [
   true = .6 ,
   false = .4
] else [
   true = .0001 ,
   false = .9999
]
```

Figure B.9 presents the Owns Suspended Enterprise MFrag. Listings B.16 presents the LPD for the RV `ownsSuspendedEnterprise(person)`. The assumption behind this LPD is that if a person is owner of at least one enterprise suspended from bidding in public procurements, then this person owns a suspended enterprise. Notice this RV could also be represented as a formula. However, due to limitations in UnBBayes' current version, this existential formula was implemented as a regular RV using the expressiveness of the LPD grammar.

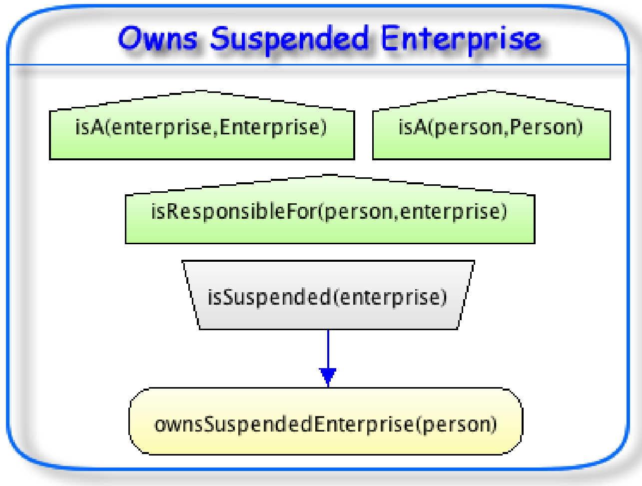


Figure B.9: MFrag Owns Suspended Enterprise.

Listing B.16: LPD for `ownsSuspendedEnterprise(person)`

```
if any enterprise have ( isSuspended = true ) [
   true = 1,
   false = 0
] else if any enterprise have ( isSuspended = false ) [
   true = 0,
   false = 1
] else [
   true = .001,
   false = .999
]
```

Figure B.10 presents the Judgment History MFrag. In it we have RVs associated to the judgement (criminal and administrative) history of a `Person`. Listings B.17, B.18, and B.19 present the LPDs for the RVs, `hasCriminalHistory(person)`, `hasAdministrative-History(person)`, and `hasCleanHistory(person)`, respectively. The assumptions behind these LPDs are that a person is more likely to have never been investigated, and the probability of a person having a clean history is lower if he/she was never investigated, higher if he/she was investigated, and extremely high if he/she was convicted[2].

[2]Maybe a better name for this node would be `isTrustworthy`. Nevertheless, the idea is that if someone

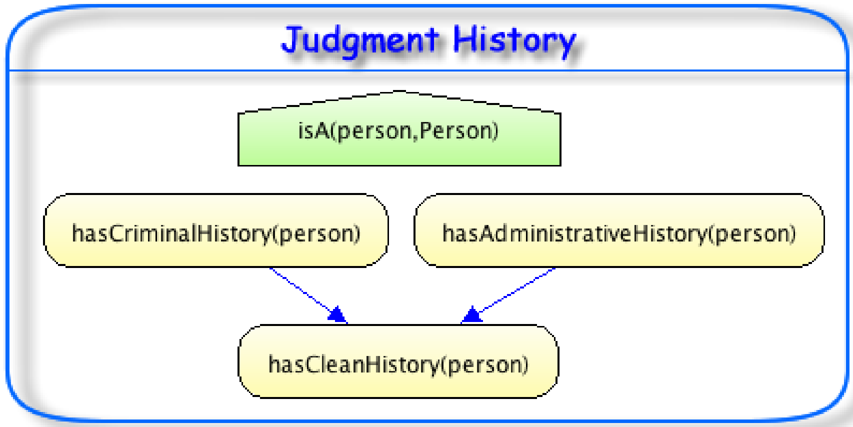


Figure B.10: MFrag Judgment History.

Listing B.17: LPD for `hasCriminalHistory(person)`

```
[
   Convicted = .0001,
   Investigated = .001,
   NeverInvestigated = .9989
]
```

Listing B.18: LPD for `hasAdministrativeHistory(person)`

```
[
   Convicted = .0001,
   Investigated = .001,
   NeverInvestigated = .9989
]
```

was investigated and/or convicted then he might not be a good candidate for being part of a procurement committee.

Listing B.19: LPD for `hasCleanHistory(person)`

```
if any person have ( hasCriminalHistory = Convicted | hasAdministrativeHistory
    = Convicted ) [
  true = .01,
  false = .99
] else if any person have ( hasCriminalHistory = Investigated |
    hasAdministrativeHistory = Investigated ) [
  true = .60,
  false = .40
] else [
  true = .99,
  false = .01
]
```

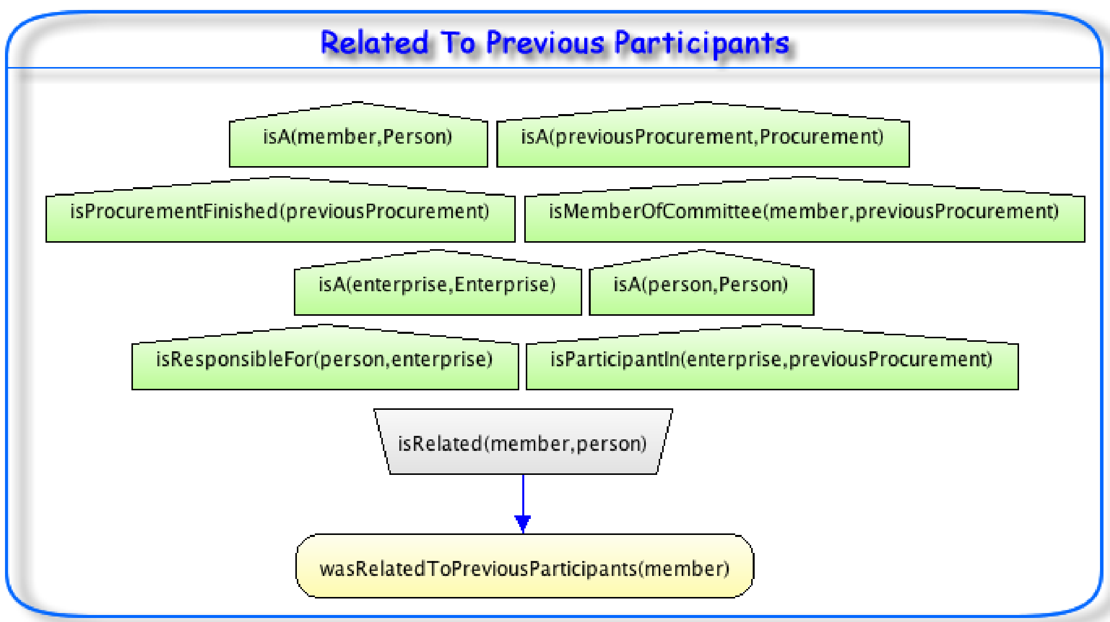


Figure B.11: MFrag Related to Previous Participants.

Figure B.11 presents the Related to Previous Participants MFrag. Listings B.20 presents the LPD for the RV `wasRelatedToPreviousParticipants(member)`. The assumption behind this LPD is that if a person was related to any owner of any enterprise participating in any previous procurement (procurement that is finished), when this person was a member of that procurement, then this member was related to previous participants. Notice this RV could also be represented as a formula. However, due to limitations in UnBBayes' current version, this existential formula was implemented as a regular RV using the expressiveness of the LPD grammar.

Listing B.20: LPD for `wasRelatedToPreviousParticipants(member)`

```
if any member.person have ( isRelated = true ) [
   true = 1,
   false = 0
] else if all member.person have ( isRelated = false ) [
   true = 0,
   false = 1
] else [
   true = .0001,
   false = .9999
]
```

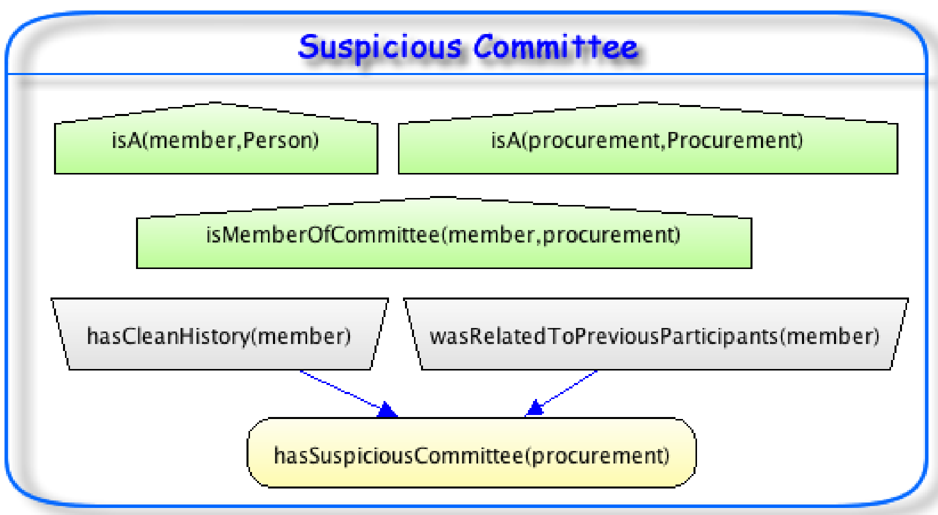


Figure B.12: MFrag Suspicious Committee.

Figure B.12 presents the Suspicious Committee MFrag. Listings B.21 presents the LPD for the RV `hasSuspiciousCommittee(procurement)`. The assumptions behind this LPD are that: if any committee member of this procurement does not have a clean history, or if any committee member was related to previous participants, then the committee is more likely to be suspicious; and that if these things happen together, the probability of having suspicious committee is even higher.

Listing B.21: LPD for `hasSuspiciousCommittee(procurement)`

```
if any member have ( wasRelatedToPreviousParticipants = true ) [
   if any member have ( hasCleanHistory = false ) [
      true = .9,
      false = .1
   ] else [
      true = .7,
      false = .3
   ]
] else if any member have ( wasRelatedToPreviousParticipants = false ) [
   if any member have ( hasCleanHistory = false ) [
      true = .7,
      false = .3
   ] else [
      true = .001,
      false = .999
   ]
] else [
      true = .001,
      false = .999
]
```

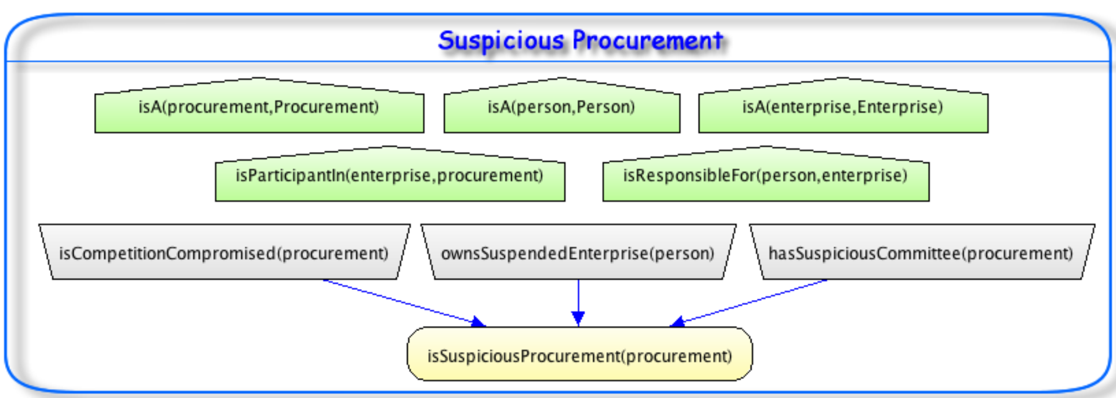


Figure B.13: MFrag Suspicious Procurement.

Figure B.13 presents the Suspicious Procurement MFrag. Listings B.22 presents the LPD for the RV `isSuspiciousProcurement(procurement)`. The assumptions behind this LPD are that: if the competition is compromised, or if any owner of a participating enterprise owns a suspended enterprise, or if committee of this procurement is suspicious, then the procurement is more likely to be suspicious; and that if these things happen together, the probability of having suspicious procurement is even higher.

Listing B.22: LPD for `isSuspiciousProcurement(procurement)`

```
if any procurement have ( isCompetitionCompromised = true &
    hasSuspiciousCommittee = true ) [
   if any person have ( ownsSuspendedEnterprise = true ) [
      true = .90,
      false = .10
   ] else [
      true = .80,
      false = .20
   ]
] else if any procurement have ( isCompetitionCompromised = true |
    hasSuspiciousCommittee = true ) [
    if any person have ( ownsSuspendedEnterprise = true ) [
      true = .80,
      false = .20
   ] else [
      true = .70,
      false = .30
   ]
] else [
    if any person have ( ownsSuspendedEnterprise = true ) [
      true = .70,
      false = .30
   ] else [
      true = .0001,
      false = .9999
   ]
]
```

B.2 Probabilistic Ontology for Maritime Domain Awareness

All the assumptions for the RVs created and for defining their LPD will be described for every MFrag designed for the MTheory that represents the PO for the MDA implemented. In each MFrag, the resident RVs are shown as yellow rounded rectangles; the input RVs are shown as gray trapezoids; the context RVs are shown as green pentagons.

B.2.1 Fist Iteration

In order to make reference easier, the rules defined during Analysis & Design in Chapter 5 Subsection 5.2.1 will be repeated here. The rules are:

1. A ship is of interest if and only if it has a terrorist crew member;
2. If a crew member is related to a terrorist, then it is more likely that he is also a terrorist;
3. If a crew member is a member of a terrorist organization, then it is more likely that he is a terrorist;
4. If an organization has a terrorist member, it is more likely that it is a terrorist organization;
5. A ship of interest is more likely to have an unusual route;
6. A ship of interest is more likely to meet other ships for trading illicit cargo;
7. A ship that meets other ships to trade illicit cargo is more likely to have an unusual route;
8. A ship of interest is more likely to have an evasive behavior;
9. A ship with evasive behavior is more likely to have non responsive electronic equipment;
10. A ship with evasive behavior is more likely to deploy an ECM;
11. A ship might have non responsive electronic equipment due to working problems;
12. A ship that is within radar range of a ship that deployed an ECM might be able to detect the ECM, but not who deployed it.

The primary goal is shown in the Ship of Interest MFrag in Figure B.14. This MFrag has only one resident node, `isShipOfInterest(ship)`. The only context node present in this MFrag define the type for the variable `ship`, which is the `Ship` entity. The input node `hasTerroristCrew(ship)` is defined in another MFrag, which will be explained later.

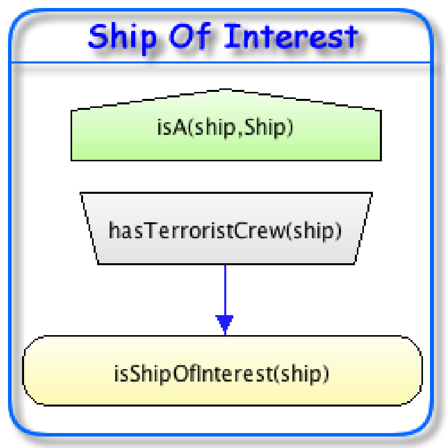


Figure B.14: MFrag for identifying the ship of interest.

The LPD for the resident node `isShipOfInterest(ship)` follows rule 1, *i.e.*, if the ship has a terrorist crew, then it is a ship of interest for sure, otherwise, it is unlikely, which was considered as 0.1%. See Listing B.23 for the complete LPD.

Listing B.23: LPD for `isShipOfInterest(ship)`

```
if any ship have ( hasTerroristCrew = true ) [
   true = 1,
   false = 0
] else [
   true = .001,
   false = .999
]
```

The question related to the identification of a terrorist crew member is presented in the Has Terrorist Crew, Terrorist Person, and Ship Characteristics MFrags in Figure B.15. The context nodes on all these MFrags refer only to the types of the variables, where `person`, `ship`, and `org`, refer to `Person`, `Ship`, and `Organization`, respectively.

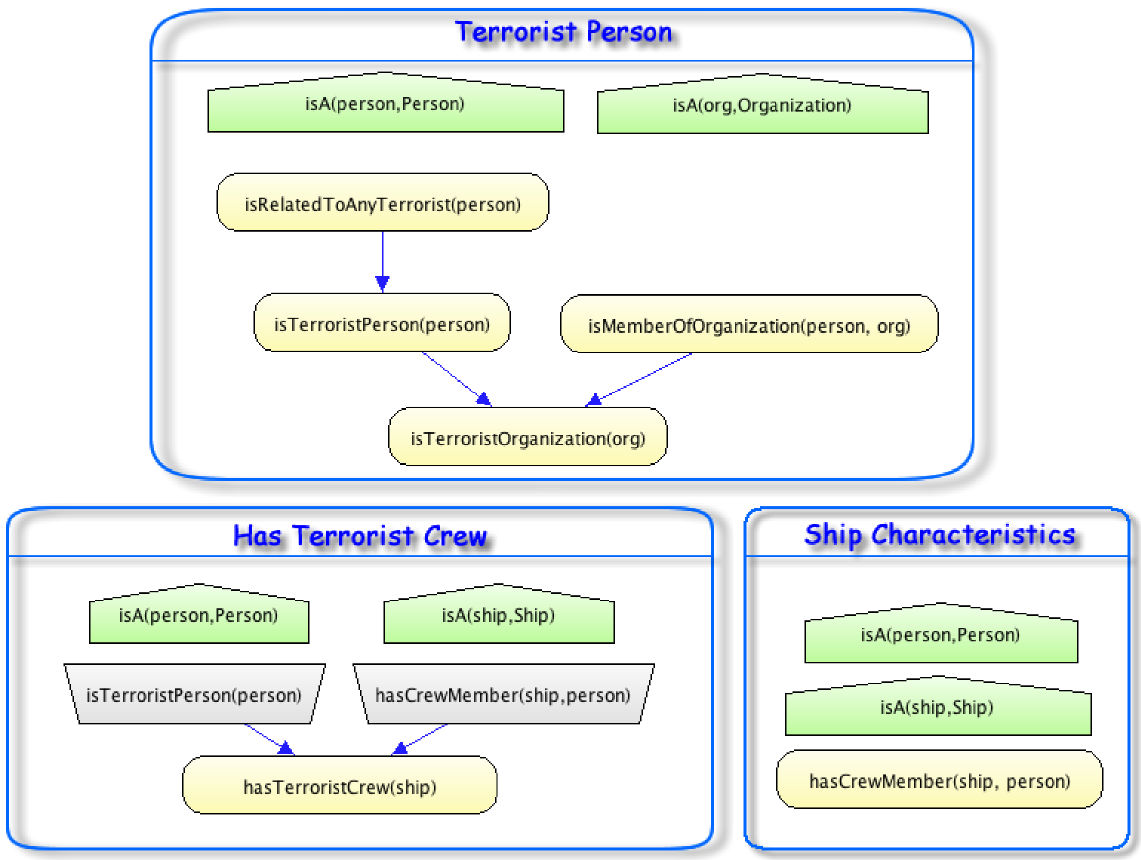


Figure B.15: MFrags for identifying a terrorist crew member.

The LPD for the resident node `isCrewMember(person, ship)` has a prior probability of 0.5% of being true, *i.e.*, 5 out of 1000 people are crew members of a given ship. In reality, given all people in the world, this ratio might be much smaller, however, we assume that we are dealing with a subset of people that is more likely to be crew members. See Listing B.24 for the complete LPD.

Listing B.24: LPD for `isCrewMember(person, ship)`

```
[
   true = .005,
   false = .995
]
```

The LPD for the resident node `hasTerroristCrew(ship)` follows rule 1. Here we just have a logical statement saying that the ship has a terrorist crew if and only if a person is a terrorist and also a crew member of this ship. If no information about its parents is known, the default distribution is used. In this case, we assume that it is unlikely that a ship has a terrorist crew, which is interpreted as 0.1%. See Listing B.25 for the complete LPD.

Listing B.25: LPD for `hasTerroristCrew(ship)`

```
if any person have ( isTerroristPerson = true ) [
   if any person.ship have ( isCrewMember = true ) [
      true = 1,
      false = 0
   ] else [
      true = 0,
      false = 1
   ]
] else if any person have ( isTerroristPerson = false ) [
   if any person.ship have ( isCrewMember = true ) [
      true = 0,
      false = 1
   ] else [
      true = 0,
      false = 1
   ]
] else [
   true = .001,
   false = .999
]
```

The LPD for the resident node `isRelatedToAnyTerrorist(person)` has a prior probability of 0.1% of being true, which means that a person is unlikely to be related to a terrorist. This information is provided by a social network system by looking at the relation `isRelatedTo` and classes `Person` and `Terrorist` presented on our design. If there is one `Terrorist` who is related to a person, then `isRelatedToAnyTerrorist(person)` is true. So we simplified our PO by just representing the relation `isRelatedToAnyTerrorist(person)`. See Listing B.26 for the complete LPD.

Listing B.26: LPD for `isRelatedToAnyTerrorist(person)`

```
[
   true = .001,
   false = .999
]
```

The LPD for the resident node `isTerroristPerson(person)` follows rule 2. If person is related to any terrorist, then this person is more likely to be a terrorist. Otherwise, it is unlikely that this person is a terrorist. Here more likely is interpreted as 70% and unlikely as 0.1%. See Listing B.58 for the complete LPD.

Listing B.27: LPD for `isTerroristPerson(person)`

```
if any person have ( isRelatedToAnyTerrorist = true ) [
   true = .7,
   false = .3
] else [
   true = .001,
   false = .999
]
```

The LPD for the resident node `isMemberOfOrganization(person, org)` has a prior probability of 1%, which means that one in every one hundred people is a member of a given organization. Again, this might be a much smaller ratio in reality, but here we assume we are dealing with a subset of people that are more likely to be members of a given organization. See Listing B.28 for the complete LPD.

Listing B.28: LPD for `isMemberOfOrganization(person, org)`

```
[
   true = .01,
   false = .99
]
```

The LPD for the resident node `isTerroristOrganization(org)` follows rules 3 and 4. If there is a person that is a terrorist and is also a member of a given organization, then this organization is likely to be a terrorist organization, otherwise, it is unlikely to be a terrorist organization. Here, we assume likely to be 90% and unlikely to be 0.1%. See Listing B.29 for the complete LPD.

Listing B.29: LPD for `isTerroristOrganization(org)`

```
if any person have ( isTerroristPerson = true ) [
   if any person.org have ( isMemberOfOrganization = true ) [
      true = .9,
      false = .1
   ] else [
      true = .001,
      false = .999
   ]
] else [
   true = .001,
   false = .999
]
```

The question related to the identification of unusual routes is presented on the Unusual Route and Meeting MFrags in Figure B.16. In both MFrags there is two context nodes to define the types of the variables `ship1` and `ship2`, which is entity `Ship`. Besides that, there is one context node that defined that `ship1` has to be different than `ship2`.

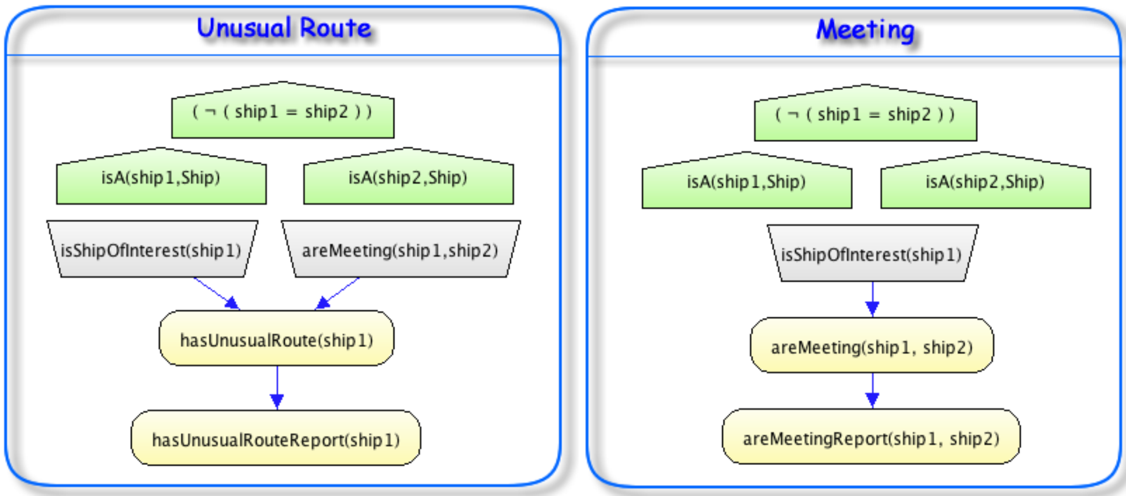


Figure B.16: MFrags for identifying the ship with unusual route.

The LPD for the resident node `areMeeting(ship1, ship2)` follows rule 6. If ship is a ship of interest, then it is more likely to meet other ships for trading illicit cargo. Otherwise, it is unlikely that this ship will meet other ships. In this case, more likely is interpreted as 75% and unlikely as 0.1%. See Listing B.30 for the complete LPD.

Listing B.30: LPD for `areMeeting(ship1, ship2)`

```
if any ship1 have ( isShipOfInterest = true ) [
   true = .75,
   false = .25
] else [
   true = .001,
   false = .999
]
```

The LPD for the resident node `areMeetingReport(ship1, ship2)` does not follow a specific rule, however, it is in associated to the fact that receiving a report about an event is not the same as the event itself (see [119] for more details), *i.e.* even though someone states that two ships met, it might be the case that whoever gave the report was mistaken. These issues are not addressed in detail in this project, but we have assumed that when two ships meet, there is a 90% chance that the report will say these two ships met, and if two ships have not met, there is a 80% chance that the report will say these two ships have not met. See Listing B.31 for the complete LPD.

Listing B.31: LPD for `areMeetingReport(ship1, ship2)`

```
if any ship1.ship2 have ( areMeeting = true ) [
   true = .9,
   false = .1
] else [
   true = .2,
   false = .8
]
```

The LPD for the resident node `hasUnusualRoute(ship1)` follows rules 5 and 7. If ship is of interest and is meeting other ships, then it is more likely the ship has an unusual route. However, if ship is of interest but is not meeting other ships, then it is likely (but less than the previous case) the ship has an unusual route. If ship is not of interest, then it does not matter if it is meeting other ships. In this scenario, this ship is unlikely to be using an unusual route. Here it is assumed more likely as 90%, likely as 75%, and unlikely as 0.1%. See Listing B.32 for the complete LPD.

Listing B.32: LPD for `hasUnusualRoute(ship1)`

```
if any ship1 have ( isShipOfInterest = true ) [
   if any ship1.ship2 have ( areMeeting = true ) [
      true = .9,
      false = .1
   ] else [
      true = .75,
      false = .25
   ]
] else [
   true = .001,
   false = .999
]
```

The LPD for the resident node `hasUnusualRouteReport(ship1)` does not follow a specific rule, however, it is in associated to the fact that receiving a report about an event is not the same as the event itself, as in the case of **areMeetingReport(ship1, ship2)**. It is assumed that when the ship has an unusual route, there is a 90% chance that the report will say the ship has an unusual route, and if the ship has a normal route, there is an 80% chance that the report will say the ship has a normal route. See Listing B.33 for the complete LPD.

Listing B.33: LPD for **hasUnusualRouteReport(ship1)**

```
if any ship1 have ( hasUnusualRoute = true ) [
   true = .9,
   false = .1
] else [
   true = .2,
   false = .8
]
```

The question related to identification of evasive behavior is shown in the Evasive Behavior, Electronics Status, and Radar MFrags in Figure B.17. As in the previous MFrags, the variables **ship**, **ship1**, and **ship2** have their type defined by their context node as entity `Ship`. In Radar and Evasive Behavior MFrags, `ship1` and `ship2` are defined as two different ships. Finally, the relations in Evasive Behavior MFrag are only valid when `ship1` is within the radar range of `ship2`.

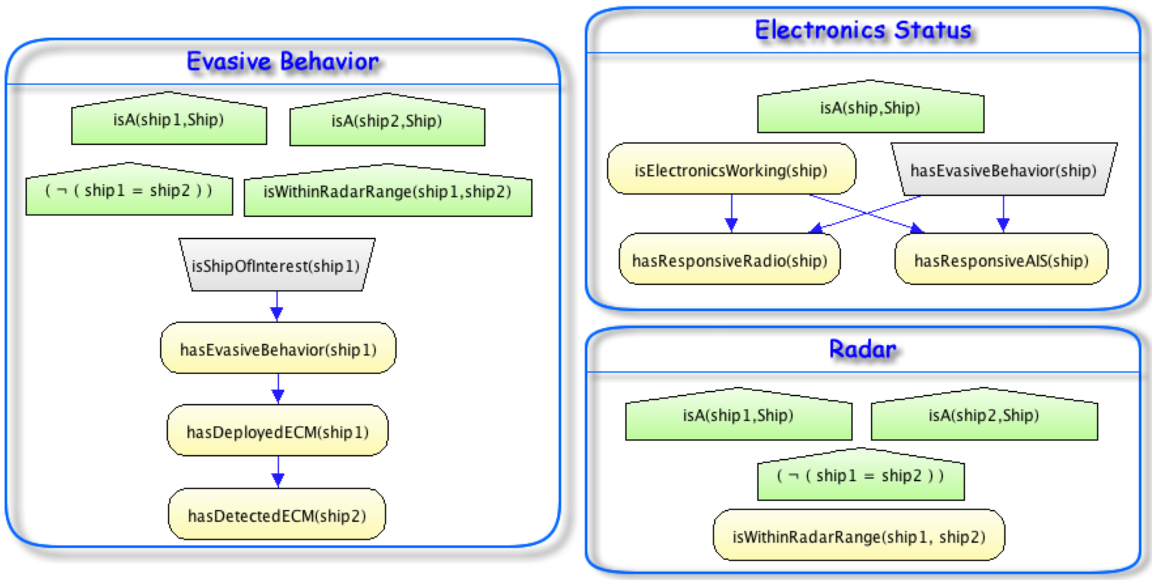


Figure B.17: MFrags for identifying the ship with evasive behavior.

The LPD for the resident node `isWithinRadarRange(ship1, ship2)` has a prior of 0.5% of having `ship1` in range of `ship2`s radar. Again, this is just a subjective analysis and it is not intended to represent a real frequency. This is also a simplification from the design we had. Some external system is going to compare the position of `ship1` with the position of `ship2` and verify if this distance is smaller or equal to `ship2`s radar range. See Listing B.34 for the complete LPD.

Listing B.34: LPD for `isWithinRadarRange(ship1, ship2)`

```
[
   true = .005,
   false = .995
]
```

The LPD for the resident node `hasEvasiveBehavior(ship1)` follows rule 8. If `ship1` is of interest, then it is more likely to have an evasive behavior. Otherwise it is unlikely to have an evasive behavior. It is assumed more likely as 75% and unlikely as 0.1% in this case. See Listing B.35 for the complete LPD.

Listing B.35: LPD for `hasEvasiveBehavior(ship1)`

```
if any ship1 have ( isShipOfInterest = true ) [
   true = .75,
   false = .25
] else [
   true = .001,
   false = .999
]
```

The LPD for the resident node `hasDeployedECM(ship1)` follows rule 10. If `ship1` has evasive behavior, then it is more likely to deploy an ECM. Otherwise, it is unlikely to deploy an ECM. It is assumed more likely as 75% and unlikely as 0.1% in this case. See Listing B.36 for the complete LPD.

Listing B.36: LPD for `hasDeployedECM(ship1)`

```
if any ship1 have ( hasEvasiveBehavior = true ) [
   true = .75,
   false = .25
] else [
   true = .001,
   false = .999
]
```

The LPD for the resident node `hasDetectedECM(ship2)` follows rule 12. If a `ship1` that deployed an ECM is within radar range of `ship2`, then it is likely that `ship2` will detect ECM, but not who deployed it. Otherwise, it is unlikely that `ship2` will detect an ECM. It is assumed likely as 90% and unlikely as 0.1% in this case. See Listing B.37 for the complete LPD.

Listing B.37: LPD for `hasDetectedECM(ship2)`

```
if any ship1 have ( hasDeployedECM = true ) [
  true = .9 ,
  false = .1
] else [
  true = .001 ,
  false = .999
]
```

The LPD for the resident node isElectronicsWorking(ship) has a prior of 95% of being working, which means that it is likely that the electronics in a ship is working. Here we simplified our model by grouping all electronics equipment and saying that if one is not working, then this RV should be **`false`**, otherwise, it is **`true`**. This is different than our design, which has a property **`isWorking`** for every electronic. See Listing B.38 for the complete LPD.

Listing B.38: LPD for `isElectronicsWorking(ship)`

```
[
  true = .95 ,
  false = .05
]
```

The LPD for the resident node `hasResponsiveRadio(ship)` follows rules 9 and 11. If **`ship`** has an evasive behavior and the electronics is not working than it is very likely to have non-responsive radio. However, if **`ship`** has an evasive behavior or the electronics is not working, but not both, then it is likely to have non-responsive radio. In any other case, it is very likely to have responsive radio. It is assumed very likely as 99% and likely as 90%, in this case. See Listing B.39 for the complete LPD.

Listing B.39: LPD for `hasResponsiveRadio(ship)`

```
if any ship have ( hasEvasiveBehavior = true ) [
   if any ship have ( isElectronicsWorking = false ) [
      true = .01,
      false = .99
   ] else [
      true = .1,
      false = .9
   ]
] else if any ship have ( hasEvasiveBehavior = false ) [
   if any ship have ( isElectronicsWorking = false ) [
      true = .1,
      false = .9
   ] else [
      true = .99,
      false = .01
   ]
] else [
   true = .99,
   false = .01
]
```

The LPD for the resident node `hasResponsiveAIS(ship)` follows rules 9 and 11. If `ship` has an evasive behavior and the electronics is not working than it is very likely to have non-responsive AIS. However, if `ship` has an evasive behavior or the electronics is not working, but not both, then it is likely to have non-responsive AIS. In any other case, it is very likely to have responsive AIS. It is assumed very likely as 99% and likely as 90%, in this case. See Listing B.40 for the complete LPD.

Listing B.40: LPD for `hasResponsiveAIS(ship)`

```
if any ship have ( hasEvasiveBehavior = true ) [
   if any ship have ( isElectronicsWorking = false ) [
      true = .01,
      false = .99
   ] else [
      true = .1,
      false = .9
   ]
] else if any ship have ( hasEvasiveBehavior = false ) [
   if any ship have ( isElectronicsWorking = false ) [
      true = .1,
      false = .9
   ] else [
      true = .99,
      false = .01
   ]
] else [
   true = .99,
   false = .01
]
```

B.2.2 Second Iteration

In order to make reference easier, the rules defined during Analysis & Design in Chapter 5 Subsection 5.2.2 will be repeated here. The rules are:

1. *A ship is of interest if and only if it has a terrorist ~~crew member~~* plan;

2. A ship has a terrorist plan if and only if it has terrorist crew member or if it was hijacked;

3. *If a crew member is related to a terrorist, then it is more likely that he is also a terrorist*;

4. *If a crew member is a member of a terrorist organization, then it is more likely that he is a terrorist*;

5. *If an organization has a terrorist member, it is more likely that it is a terrorist organization*;

6. *A ship of interest is more likely to have an unusual route*, independent of its intention;
7. *A ship of interest*, with plans of exchanging illicit cargo, *is more likely to meet other ships*;
8. *A ship that meets other ships to trade illicit cargo is more likely to have an unusual route*;
9. A fishing ship is more likely to have a normal change in its destination (*e.g.*, to sell the fish caught) than merchant ships;
10. A normal change in destination will probably change the usual route of the ship;
11. *A ship of interest*, with plans of exchanging illicit cargo, *is more likely to have an evasive behavior*;
12. A ship with evasive behavior is more likely to have non responsive electronic equipment;
13. A ship might have non responsive electronic equipment due to maintenance problems;
14. ~~A ship with evasive behavior is more likely to deploy an ECM~~;
15. ~~A ship that is within radar range of a ship that deployed an ECM might be able to detect the ECM, but not who deployed it~~;
16. A ship of interest, with plans of exchanging illicit cargo, is more likely to have an erratic behavior;
17. A ship with normal behavior usually does not have the crew visible on the deck;
18. A ship with erratic behavior usually has the crew visible on the deck;
19. If the ship has some equipment failure, it is more likely to see the crew on the deck in order to fix the problem;

20. A ship of interest, independent of its intention, is more likely to have an aggressive behavior;

21. A ship with aggressive behavior is more likely to have weapons visible and to jettison cargo;

22. A ship with normal behavior is not likely to have weapons visible nor to jettison cargo.

Rules inherited from the first iteration are in italic. Items crossed out refer to rules that were considered in the first iteration, but now they have been changed or removed.

This Section will only describe the MFrags and LPDs that have been changed or added in the second iteration. Please refer to Section B.2.1 for those that remain the same. Figure B.18 presents the MTheory created in the second iteration with information on which MFrags are the same, which are new, and which were changed.

Figure B.19 presents the Ship Characteristics MFrag. It adds the RVs `hasTypeOfShip(ship)` and `isHijacked(ship)`, which have their LPDs described by Listings B.41 and B.42, respectively. The assumptions behind these LPDs are that a ship is slightly more likely to be a fishing ship than a merchant ship and a ship is unlikely to be hijacked.

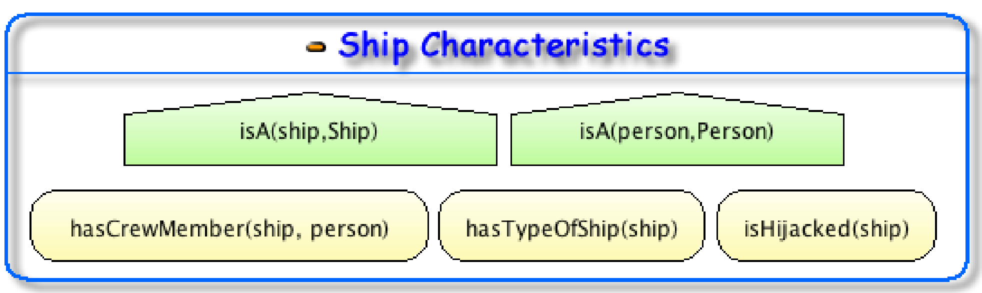


Figure B.19: Ship Characteristics MFrag.

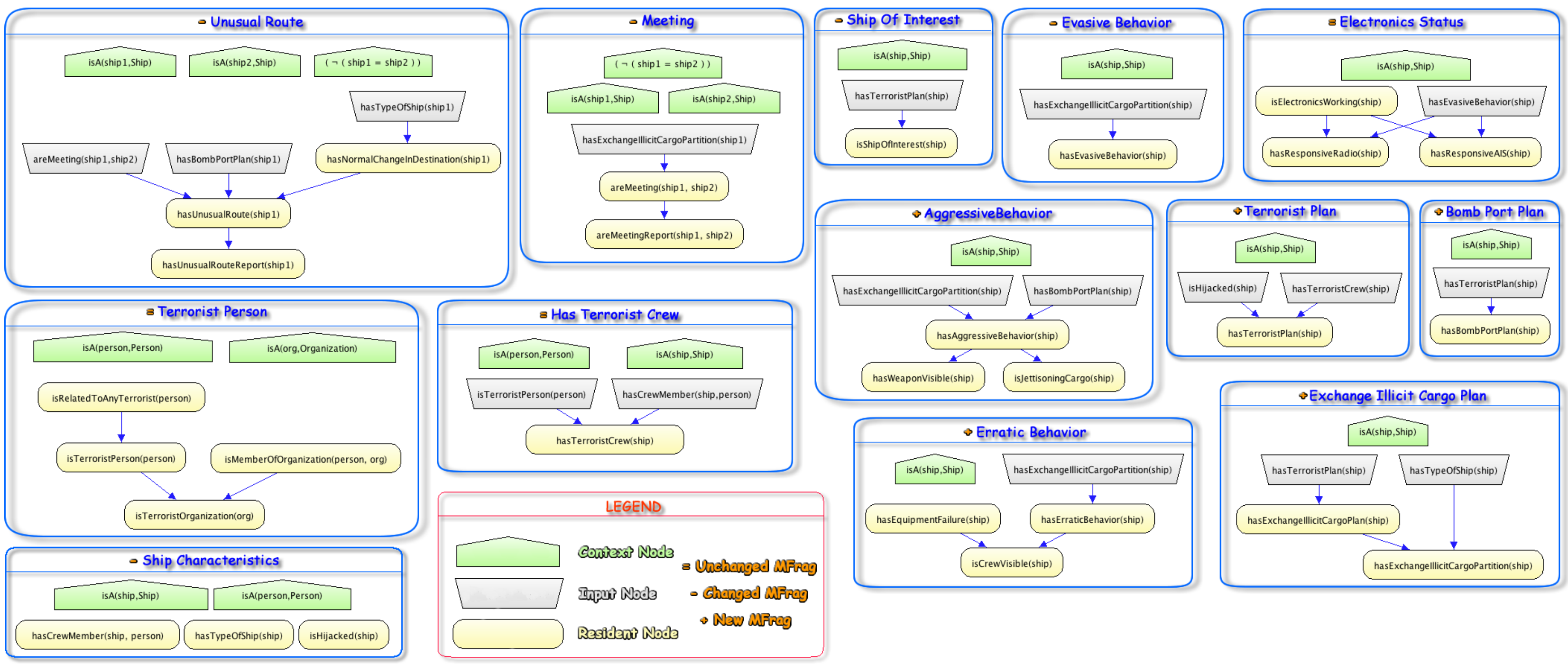


Figure B.18: MTheory created in second iteration.

Listing B.41: LPD for `hasTypeOfShip(ship)`

```
[
  Fishing = .6,
  Merchant = .4
]
```

Listing B.42: LPD for `isHijacked(ship)`

```
[
  true = .05,
  false = .95
]
```

Figure B.20 presents the Terrorist Plan MFrag. Listing B.43 presents the LPD for the RV `hasTerroristPlan(ship)`. The assumption behind this LPD is that a ship has a terrorist plan if and only if it has terrorist crew member or if it was hijacked (rule 2).

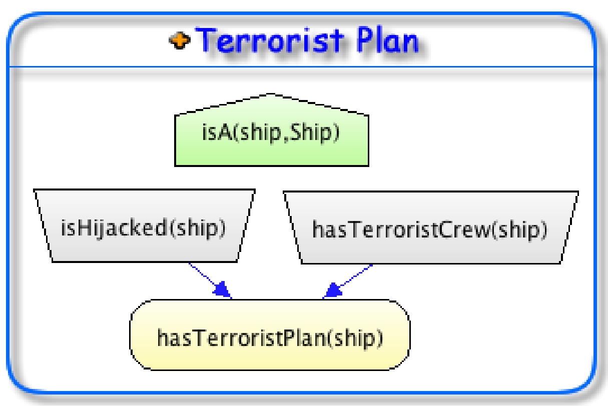


Figure B.20: Terrorist Plan MFrag.

Listing B.43: LPD for `hasTerroristPlan(ship)`

```
if any ship have ( hasTerroristCrew = true | isHijacked = true ) [
   ExchangeIllicitCargoPlan = .7,
   BombPortPlan = .3,
   NoPlan = 0
] else [
   ExchangeIllicitCargoPlan = 0,
   BombPortPlan = 0,
   NoPlan = 1
]
```

Figure B.21 presents the Bomb Port Plan MFrag. Listing B.44 presents the LPD for the RV `hasBombPortPlan(ship)`. Here we just have a deterministic rule saying that if `ship` has the terrorist plan `BombPortPlan`, then the RV `hasBombPortPlan(ship)` is `true`.

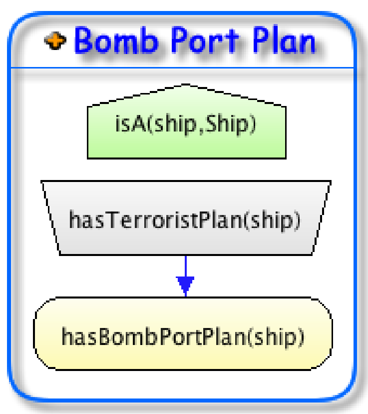


Figure B.21: Bomb Port Plan MFrag.

Listing B.44: LPD for `hasBombPortPlan(ship)`

```
if any ship have ( hasTerroristPlan = BombPortPlan ) [
   true = 1,
   false = 0
] else [
   true = 0,
   false = 1
]
```

Figure B.22 presents the Exchange Illicit Cargo Plan MFrag. Listings B.45 and B.46 present the LPDs for the RVs `hasBombPortPlan(ship)` and `hasExchangeIllicitCargoPartition(ship)`, respectively. Here we just have a deterministic rule saying that if `ship` has the terrorist plan `ExchangeIllicitCargoPlan`, then the RV `hasExchangeIllicitCargoPlan(ship)` is `true`. The other node defines a exchange illicit cargo partition based on the type of the ship.

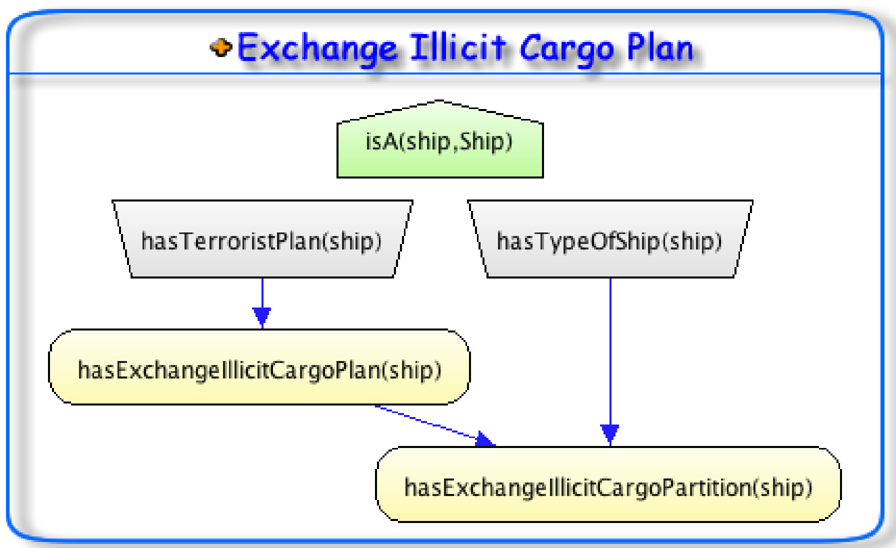


Figure B.22: Exchange Illicit Cargo Plan MFrag.

Listing B.45: LPD for `hasExchangeIllicitCargoPlan(ship)`

```
if any ship have ( hasTerroristPlan = ExchangeIllicitCargoPlan ) [
   true = 1,
   false = 0
] else [
   true = 0,
   false = 1
]
```

Listing B.46: LPD for `hasExchangeIllicitCargoPartition(ship)`

```
if any ship have ( hasExchangeIllicitCargoPlan = true ) [
   if any ship have ( hasTypeOfShip = Fishing ) [
      ExchangeIllicitCargoPlanForFishingShip = 1,
      ExchangeIllicitCargoPlanForMerchantShip = 0,
      NoPlan = 0
   ] else [
      ExchangeIllicitCargoPlanForFishingShip = 0,
      ExchangeIllicitCargoPlanForMerchantShip = 1,
      NoPlan = 0
   ]
] else [
   ExchangeIllicitCargoPlanForFishingShip = 0,
   ExchangeIllicitCargoPlanForMerchantShip = 0,
   NoPlan = 1
]
```

Figure B.23 presents the Ship of Interest MFrag. Listing B.47 presents the LPD for the RV `isShipOfInterest(ship)`. The assumption behind this LPD is that a ship is of interest if and only if it has a terrorist plan (rule 1).

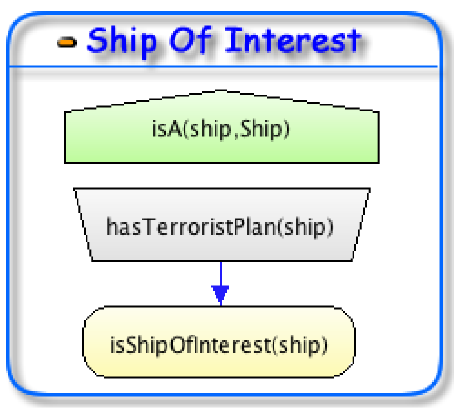


Figure B.23: Ship of Interest MFrag.

Listing B.47: LPD for `isShipOfInterest(ship)`

```
if any ship have ( hasTerroristPlan = NoPlan ) [
   true = 0,
   false = 1
] else [
   true = 1,
   false = 0
]
```

Figure B.24 presents the Meeting MFrag. Listing B.48 presents the LPD for the RV `areMeeting(ship)`. The assumption behind this LPD is that a ship of interest, with plans of exchanging illicit cargo, is more likely to meet other ships (rule 7).

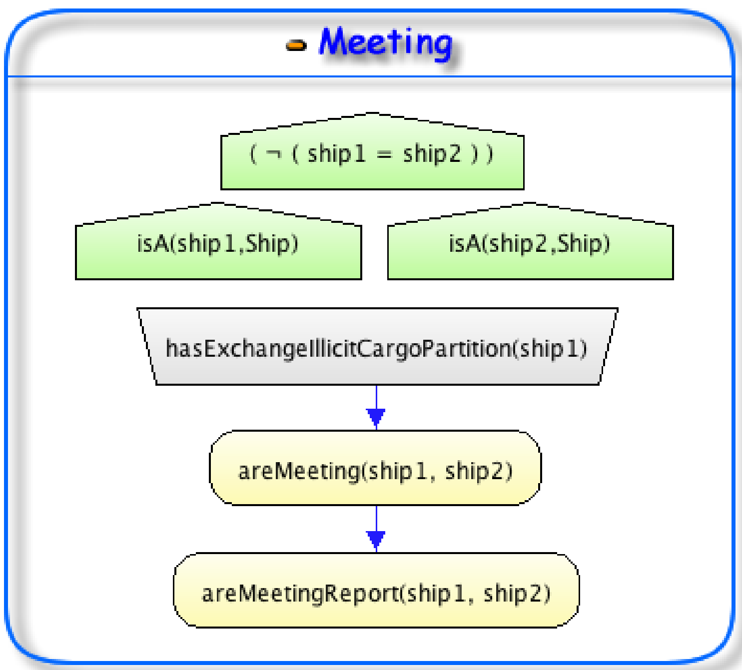


Figure B.24: Meeting MFrag.

Listing B.48: LPD for `areMeeting(ship1, ship2)`

```
if any ship1 have ( hasExchangeIllicitCargoPartition = NoPlan ) [
   true = 0,
   false = 1
] else [
   true = 1,
   false = 0
]
```

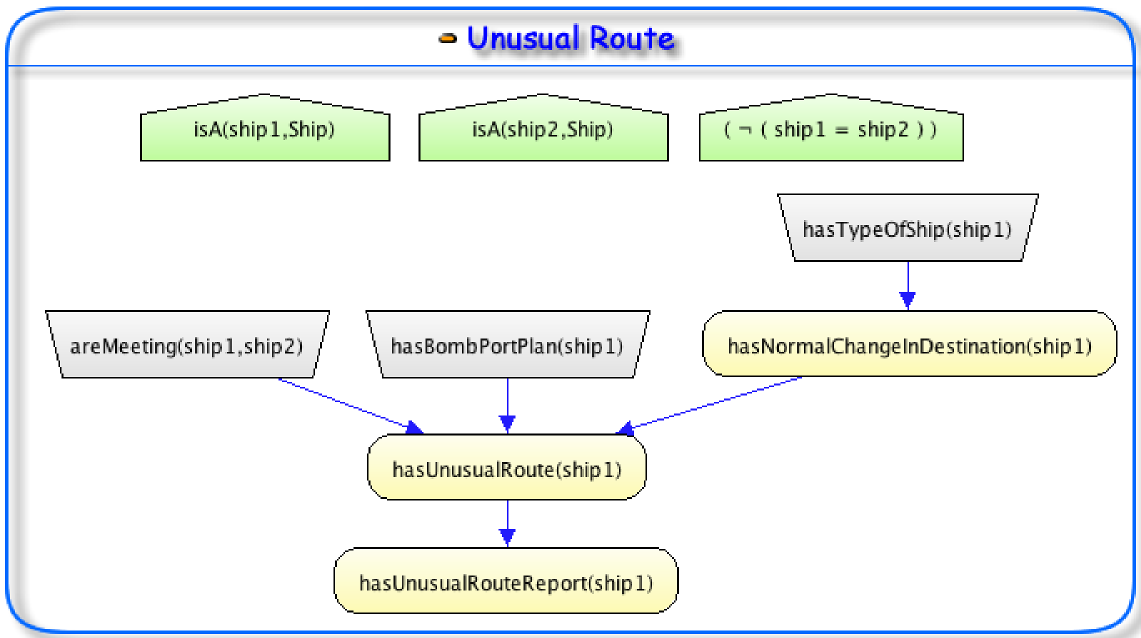


Figure B.25: Unusual Route MFrag.

Figure B.25 presents the Unusual Route MFrag. Listings B.49 and B.50 present the LPDs for the RVs `hasNormalChangeInDestination(ship1)` and `hasUnusual-Route(ship1)`, respectively. The assumptions behind these LPDs are that a fishing ship is more likely to have a normal change in its destination (*e.g.*, to sell the sh caught) than merchant ships (rule 9), that a normal change in destination will probably change the usual route of the ship (rule 10), that a ship of interest is more likely to have an unusual route, independent of its intention (rule 6), and that a ship that meets other ships to trade illicit cargo is more likely to have an unusual route (rule 8).

Listing B.49: LPD for **`hasNormalChangeInDestination(ship1)`**

```
if any ship1 have ( hasTypeOfShip = Fishing ) [
   true = .2 ,
   false = .8
] else if any ship1 have ( hasTypeOfShip = Merchant ) [
   true = .05 ,
   false = .95
] else [
   true = .1 ,
   false = .9
]
```

Listing B.50: LPD for **`hasUnusualRoute(ship1)`**

```
if any ship1 have ( hasBombPortPlan = true | hasNormalChangeInDestination =
    true ) [
   true = .9 ,
   false = .1
] else if any ship1.ship2 have ( areMeeting = true ) [
   true = .9 ,
   false = .1
] else [
   true = .05 ,
   false = .95
]
```

Figure B.26 presents the Evasive Behavior MFrag. Listing B.51 presents the LPD for the RV **`hasEvasiveBehavior(ship)`**. The assumption behind this LPD is that a ship of interest, with plans of exchanging illicit cargo, is more likely to have an evasive behavior (rule 11).

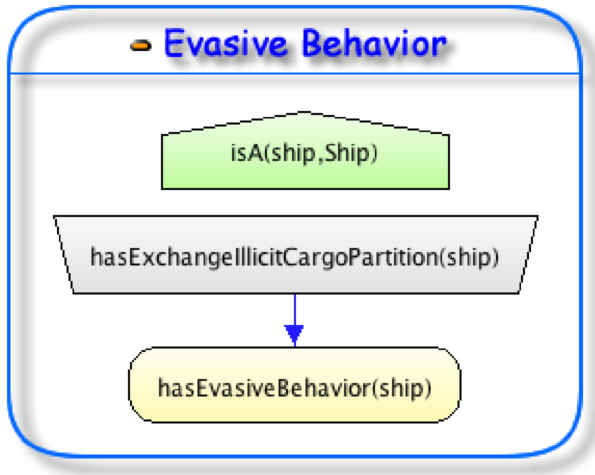


Figure B.26: Evasive Behavior MFrag.

Listing B.51: LPD for `hasEvasiveBehavior(ship)`

```
if any ship have ( hasExchangeIllicitCargoPartition =
    ExchangeIllicitCargoPlanForMerchantShip ) [
   true = 1,
   false = 0
] else [
   true = 0,
   false = 1
]
```

Figure B.27 presents the Aggressive Behavior MFrag. Listings B.52, B.53, and B.54 present the LPDs for the RVs `hasAggressiveBehavior(ship)`, `hasWeaponVisible(ship)`, and `isJettisoningCargo(ship)`, respectively. The assumptions behind these LPDs are that a ship of interest, independent of its intention, is more likely to have an aggressive behavior (rule 20), that a ship with aggressive behavior is more likely to have weapons visible and to jettison cargo (rule 21), and that a ship with normal behavior is not likely to have weapons visible nor to jettison cargo (rule 22).

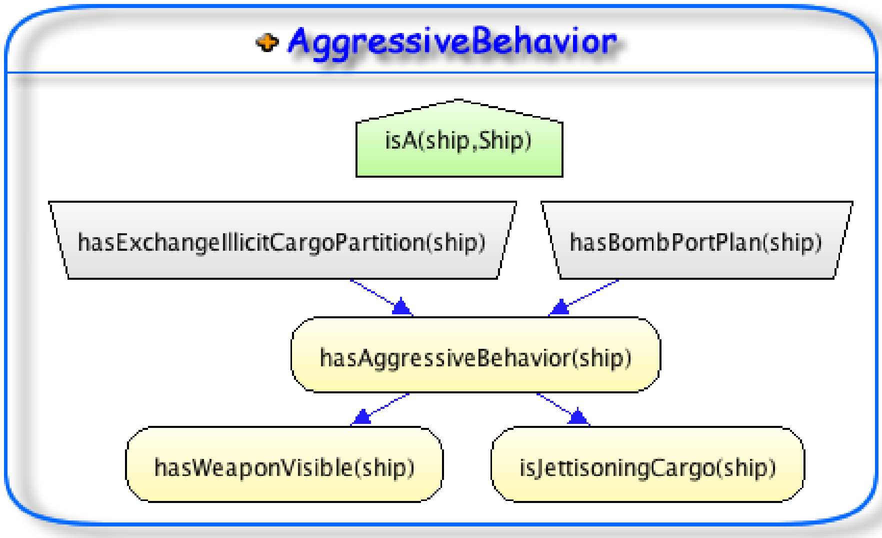


Figure B.27: Aggressive Behavior MFrag.

Listing B.52: LPD for `hasAggressiveBehavior(ship)`

```
if any ship have ( ~ hasExchangeIllicitCargoPartition = NoPlan | hasBombPortPlan = true ) [
   true = 1,
   false = 0
] else [
   true = 0,
   false = 1
]
```

Listing B.53: LPD for `hasWeaponVisible(ship)`

```
if any ship have ( hasAggressiveBehavior = true ) [
   true = .7,
   false = .3
] else [
   true = .05,
   false = .95
]
```

Listing B.54: LPD for `isJettisoningCargo(ship)`

```
if any ship have ( hasAggressiveBehavior = true ) [
   true = .25,
   false = .75
] else [
   true = .05,
   false = .95
]
```

Figure B.28 presents the Erratic Behavior MFrag. Listings B.55, B.56, and B.57 present the LPDs for the RVs `hasErraticBehavior(ship)`, `hasEquipmentFailure(ship)`, and `isCrewVisible(ship)`, respectively. The assumptions behind these LPDs are that a ship of interest, with plans of exchanging illicit cargo, is more likely to have an erratic behavior (rule 16), that a ship with normal behavior usually does not have the crew visible on the deck (rule 17), that a ship with erratic behavior usually has the crew visible on the deck (rule 18), and that if the ship has some equipment failure, it is more likely to see the crew on the deck in order to fix the problem (rule 19).

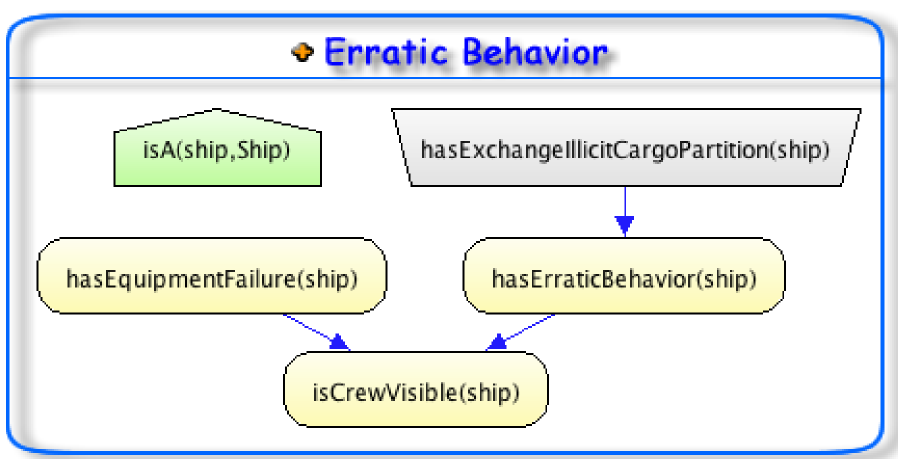


Figure B.28: Erratic Behavior MFrag.

Listing B.55: LPD for `hasErraticBehavior(ship)`

```
if any ship have ( hasExchangeIllicitCargoPartition =
    ExchangeIllicitCargoPlanForMerchantShip ) [
   true = 1,
   false = 0
] else [
   true = 0,
   false = 1
]
```

Listing B.56: LPD for `hasEquipmentFailure(ship)`

```
[
   true = .05,
   false = .95
]
```

Listing B.57: LPD for `isCrewVisible(ship)`

```
if any ship have ( hasErraticBehavior = true ) [
   if any ship have ( hasEquipmentFailure = true ) [
      true = .65,
      false = .35
   ] else [
      true = .6,
      false = .4
   ]
] else if any ship have ( hasErraticBehavior = false ) [
   if any ship have ( hasEquipmentFailure = true ) [
      true = .45,
      false = .55
   ] else [
      true = .05,
      false = .95
   ]
] else [
   true = .05,
   false = .95
]
```

B.2.3 Third Iteration

Although I am the first author of the paper published at Fusion 2011 on PO for MDA [18], which most of this Subsection is based on, most of the research on the domain tackled in this iteration was done by Richard Haberlin, who is co-author of the paper and SME on the PROGNOS project. In fact, the first paper published by Haberlin and Costa about this domain was [55]. However, the model presented in it was only a BN, not a probabilistic ontology implemented using PR-OWL/MEBN.

In order to make reference easier, the rules defined during Analysis & Design in Chapter 5 Subsection 5.2.3 will be repeated here. The rules are:

1. Terrorist organization grouping;
   (a) *If a crew member is a member of a terrorist organization, then it is more likely that he is a terrorist;*
   (b) *If an organization has a terrorist member, it is more likely that it is a terrorist organization.*
2. Background influence grouping;
   (a) For those who are terrorists, 100% of them chose to do so because of something in their past. That is, no one was born a terrorist, or just woke up one day and decided to be a terrorist. That is the easy case. For those who are not, 20% chose not to become terrorists despite having some possible factor in their background and 80% chose not to become a terrorist possibly because they have never been exposed[3].
   (b) An individual is usually negatively affected (leads him/her in becoming a terrorist) by having direct knowledge of someone either detained or killed by coalition forces during the conflict;

[3] This rule and explanation was given by the SME.

(c) In the geographic area of interest, an estimated 2% of the population knows someone who was killed as a result of OEF/OIF [94];

(d) In the geographic area of interest, approximately 2% of the population knows someone detained as a result of coalition operations [94];

(e) Contrary to common perception, terrorists are predominantly married in keeping with the teachings of the Quran [116]. And about half of the general population in the target demographic is married.

3. Communication grouping;

(a) It is possible that a crew member may communicate with a terrorist without being involved in terrorism due to non-terrorist affiliations or other relationships that have some normal expectation of interaction;

(b) For each of the internet communications paths there is also the background usage rate of 28.8% in the Middle East [5]. Because the data is not broken down for the three internet transmission paths, this probability was applied equally to chat room, email, and weblog methods of communication;

(c) Similarly, cellular telephone usage among the general population is assumed to be 31.6% based on Egyptian subscriber rates [4];

(d) Given the availability of cellular technology and subscribing to the prioritization, a probability of 90% is assigned to terrorists communicating using cellular telephones;

(e) The transient nature and unfettered availability of chat room communications makes it appealing to individuals who desire to remain nameless. A probability of 85% is assigned to terrorists communicating through chat rooms;

(f) Email is the least desirable form of communication because it requires some form of subscriber account. Even in the event that fictitious information is used in creating such an account, an auditable trail may lead determined forces to the originator. Still, it is a versatile means of communication and is assigned a probability of 65% for terrorists;

(g) The anonymity associated with weblog interaction is very appealing to terrorists. This path is similar to chat room communications, but is less transient in content and can reach more subscribers simultaneously. For these reasons, a probability of 80% is assigned to weblog communications.

4. Relationship grouping;

    (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;

    (b) Research shows that if a crew member has a relationship with terrorists, there is a 14% chance that he is related to a terrorist.

5. Cluster grouping;

    (a) It is assumed that all active terrorists fall into one of the terrorist cliques or their subsidiaries described by Sageman [116];

    (b) Contrary to popular thought, terrorists tend to not be unskilled drifters with no options other than martyrdom;

    (c) Many believe terrorist recruits to be uneducated simpletons who are easily persuaded by eloquent muftis who appeal to their sense of honor and perception of persecution. In fact, the data indicate that the typical terrorist is more educated than the average global citizen and is by far more educated than those in the Middle East, North Africa, and Southeastern Asia regions [116];

    (d) Terrorist from the clusters described by Sageman [116] are less likely to be of lower class than other people from that demographic area.

Rules inherited from the first and second iterations are in italic. Items crossed out refer to rules that were considered in the first and second iteration, but now they have been changed or removed.

This Section will only describe the MFrags and LPDs that have been changed or added in the third iteration. Please refer to Sections B.2.1 and B.2.2 for those that remain the same. Figure B.29 presents the MTheory created in the third iteration. The Terrorist Person MFrag was the only one that was changed. All others were added in this iteration.

Listing B.58 presents the LPD for the RV `isTerroristPerson(person)` from the Terrorist Person MFrag. The assumptions behind this LPD and the other ones not shown here because they are the same as in the previous iterations, are the rules in terrorist organization grouping.

Listing B.58: LPD for `isTerroristPerson(person)`

```
[
   true = .001,
   false = .999
]
```

Listings B.59, B.60, B.61, B.62, and B.63 present the LPDs for the RVs `communicatesWithTerrorist(person)`, `usesCellular(person)`, `usesEmail(person)`, `usesWeblog(person)`, and `usesChatroom(person)`, respectively. These RVs are defined in the Person Communications MFrag. The assumptions behind these LPDs are the rules in communication grouping.

Listing B.59: LPD for `communicatesWithTerrorist(person)`

```
if any person have ( isTerroristPerson = true ) [
   true = 1,
   false = 0
] else if any person have ( isTerroristPerson = false ) [
   true = .001,
   false = .999
] else [
   true = .002,
   false = .998
]
```

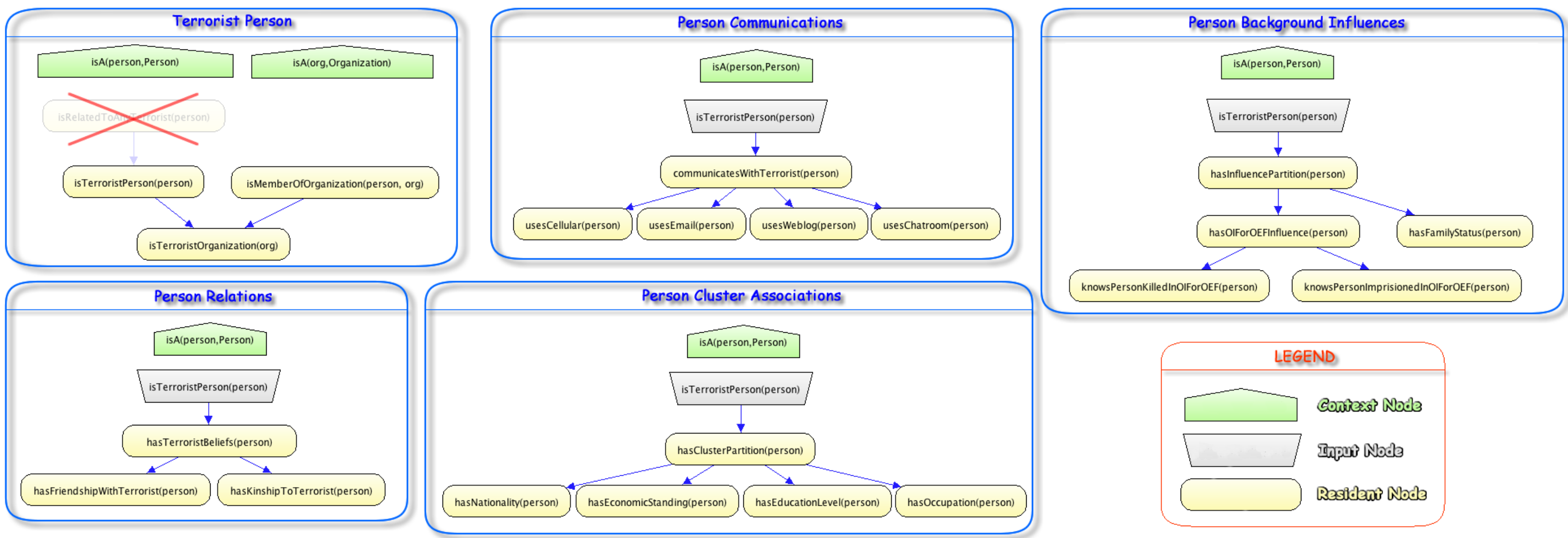


Figure B.29: MTheory created in third iteration.

Listing B.60: LPD for `usesCellular(person)`

```
if any person have ( communicatesWithTerrorist = true ) [
   true = .9,
   false = .1
] else if any person have ( communicatesWithTerrorist = false ) [
   true = .316,
   false = .684
] else [
   true = .32,
   false = .68
]
```

Listing B.61: LPD for `usesEmail(person)`

```
if any person have ( communicatesWithTerrorist = true ) [
   true = .65,
   false = .35
] else if any person have ( communicatesWithTerrorist = false ) [
   true = .288,
   false = .712
] else [
   true = .29,
   false = .71
]
```

Listing B.62: LPD for `usesWeblog(person)`

```
if any person have ( communicatesWithTerrorist = true ) [
   true = .8,
   false = .2
] else if any person have ( communicatesWithTerrorist = false ) [
   true = .288,
   false = .712
] else [
   true = .29,
   false = .71
]
```

Listing B.63: LPD for `usesChatroom(person)`

```
if any person have ( communicatesWithTerrorist = true ) [
   true = .85,
   false = .15
] else if any person have ( communicatesWithTerrorist = false ) [
   true = .288,
   false = .712
] else [
   true = .29,
   false = .71
]
```

Listings B.64, B.65, B.66, B.67, and B.68 present the LPDs for the RVs `hasInfluencePartition(person)`, `hasFamilyStatus(person)`, `hasOIForOEFInfluence(person)`, `knowsPersonKilledInOIForOEF(person)`, and `knowsPersonImprisionedInOIForOEF(person)`,

respectively. These RVs are defined in the Person Background Influences MFrag. The assumptions behind these LPDs are the rules in background influence grouping.

Listing B.64: LPD for `hasInfluencePartition(person)`

```
if any person have ( isTerroristPerson = true ) [
   true = 1,
   false = 0
] else if any person have ( isTerroristPerson = false ) [
   true = .20,
   false = .80
] else [
   true = .001,
   false = .999
]
```

Listing B.65: LPD for `hasFamilyStatus(person)`

```
if any person have ( hasInfluencePartition = true ) [
   Married = .73,
   Single = .27
] else if any person have ( hasInfluencePartition = false ) [
   Married = .52,
   Single = .48
] else [
   Married = .60,
   Single = .40 ]
```

Listing B.66: LPD for `hasOIForOEFInfluence(person)`

```
if any person have ( hasInfluencePartition = true ) [
   true = .75,
   false = .25
] else if any person have ( hasInfluencePartition = false ) [
   true = .02,
   false = .98
] else [
   true = .001,
   false = .999
]
```

Listing B.67: LPD for `knowsPersonKilledInOIForOEF(person)`

```
if any person have ( hasOIForOEFInfluence = true ) [
   None = .98,
   Few = .015,
   Many = .005
] else if any person have ( hasOIForOEFInfluence = false ) [
   None = .999,
   Few = .0008,
   Many = .0002
] else [
   None = .999,
   Few = .0008,
   Many = .0002
]
```

Listing B.68: LPD for `knowsPersonImprisionedInOIForOEF(person)`

```
if any person have ( hasOIForOEFInfluence = true ) [
   None = .98,
   Few = .015,
   Many = .005
] else if any person have ( hasOIForOEFInfluence = false ) [
   None = .999,
   Few = .0008,
   Many = .0002
] else [
   None = .999,
   Few = .0008,
   Many = .0002
]
```

Listings B.69, B.70, and B.71 present the LPDs for the RVs `hasTerroristBeliefs(person)`, `hasFriendshipWithTerrorist(person)`, and `hasKinshipToTerrorist(person)`, respectively. These RVs are defined in the Person Relations MFrag. The assumptions behind these LPDs are the rules in relationship grouping.

Listing B.69: LPD for `hasTerroristBeliefs(person)`

```
if any person have ( isTerroristPerson = true ) [
   true = .75,
   false = .25
] else if any person have ( isTerroristPerson = false ) [
   true = .001,
   false = .999
] else [
   true = .002,
   false = .998
]
```

Listing B.70: LPD for `hasFriendshipWithTerrorist(person)`

```
if any person have ( hasTerroristBeliefs = true ) [
   None = .32,
   Few = .40,
   Many = .28
] else if any person have ( hasTerroristBeliefs = false ) [
   None = .999,
   Few = .0008,
   Many = .0002
] else [
   None = .999,
   Few = .0008,
   Many = .0002 ]
```

Listing B.71: LPD for `hasKinshipToTerrorist(person)`

```
if any person have ( hasTerroristBeliefs = true ) [
   None = .86,
   Few = .10,
   Many = .04
] else if any person have ( hasTerroristBeliefs = false ) [
   None = .999,
   Few = .0008,
   Many = .0002
] else [
   None = .999,
   Few = .0008,
   Many = .0002 ]
```

Listings B.72, B.73, B.74, B.75, and B.76 present the LPDs for the RVs **hasClusterPartition(person)**, **hasNationality(person)**, **hasEconomicStanding(person)**, **hasEducationLevel(person)**, and **hasOccupation(person)**, respectively. These RVs are defined in the Person Cluster Associations MFrag. The assumptions behind these LPDs are the rules in cluster grouping.

Listing B.72: LPD for **hasClusterPartition(person)**

```
if any person have ( isTerroristPerson = true ) [
   CentralStaff = .18,
   SoutheastAsia = .12,
   MaghrebArab = .30,
   CoreArab = .32,
   Other = .08
] else if any person have ( isTerroristPerson = false ) [
   CentralStaff = 0,
   SoutheastAsia = 0,
   MaghrebArab = 0,
   CoreArab = 0,
   Other = 1
] else [ CentralStaff = .00018,
         SoutheastAsia = .00012,
         MaghrebArab = .0003,
         CoreArab = .00032,
         Other = .99908 ]
```

Listing B.73: LPD for **hasNationality(person)**

```
if any person have ( hasClusterPartition = CentralStaff ) [
   Egypt = .63,
   SaudiArabia = .09,
   Kuwait = .09,
   Jordan = .06,
   Iraq = .03,
   Sudan = .03,
   Libya = .03,
   Lebannon = .04,
   Indonesia = 0,
   Malaysia = 0,
   Singapore = 0,
   Pakistan = 0,
   Philippines = 0,
   France = 0,
   Algeria = 0,
   Morocco = 0,
   Syria = 0,
   Tunisia = 0,
   UAE = 0,
```

```
    Yemen = 0,
    Other = 0
] else if any person have ( hasClusterPartition = SoutheastAsia ) [
    Egypt = 0,
    SaudiArabia = 0,
    Kuwait = 0,
    Jordan = 0,
    Iraq = 0,
    Sudan = 0,
    Libya = 0,
    Lebannon = 0,
    Indonesia = .57,
    Malaysia = .14,
    Singapore = .10,
    Pakistan = 0,
    Philippines = .09,
    France = 0,
    Algeria = 0,
    Morocco = 0,
    Syria = 0,
    Tunisia = 0,
    UAE = 0,
    Yemen = 0,
    Other = .10
] else if any person have ( hasClusterPartition = MaghrebArab ) [
    Egypt = 0,
    SaudiArabia = 0,
    Kuwait = 0,
    Jordan = 0,
    Iraq = 0,
    Sudan = 0,
    Libya = 0,
    Lebannon = 0,
    Indonesia = 0,
    Malaysia = 0,
    Singapore = 0,
    Pakistan = 0,
    Philippines = 0,
    France = .34,
    Algeria = .28,
    Morocco = .19,
    Syria = 0,
    Tunisia = .09,
    UAE = 0,
    Yemen = 0,
    Other = .10
] else if any person have ( hasClusterPartition = CoreArab ) [
    Egypt = .07,
    SaudiArabia = .50,
    Kuwait = .07,
    Jordan = 0,
    Iraq = 0,
    Sudan = 0,
    Libya = 0,
    Lebannon = 0,
    Indonesia = 0,
```

```
    Malaysia = 0,
    Singapore = 0,
    Pakistan = .04,
    Philippines = 0,
    France = 0,
    Algeria = 0,
    Morocco = .07,
    Syria = .04,
    Tunisia = 0,
    UAE = .04,
    Yemen = .07,
    Other = .10
] else if any person have ( hasClusterPartition = Other ) [
    Egypt = 0.0454,
    SaudiArabia = 0.0139,
    Kuwait = 0.0015,
    Jordan = 0.0033,
    Iraq = 0.0172,
    Sudan = 0.0231,
    Libya = 0.0035,
    Lebannon = 0.0023,
    Indonesia = 0.1257,
    Malaysia = 0.015,
    Singapore = 0.0027,
    Pakistan = 0.0928,
    Philippines = 0.0503,
    France = 0.0342,
    Algeria = 0.0191,
    Morocco = 0.0175,
    Syria = 0.0115,
    Tunisia = 0.0057,
    UAE = 0.0025,
    Yemen = 0.0129,
    Other = 0.50
] else [
    Egypt = 0.0454,
    SaudiArabia = 0.0139,
    Kuwait = 0.0015,
    Jordan = 0.0033,
    Iraq = 0.0172,
    Sudan = 0.0231,
    Libya = 0.0035,
    Lebannon = 0.0023,
    Indonesia = 0.1257,
    Malaysia = 0.015,
    Singapore = 0.0027,
    Pakistan = 0.0928,
    Philippines = 0.0503,
    France = 0.0342,
    Algeria = 0.0191,
    Morocco = 0.0175,
    Syria = 0.0115,
    Tunisia = 0.0057,
    UAE = 0.0025,
    Yemen = 0.0129,
    Other = 0.50 ]
```

Listing B.74: LPD for `hasEconomicStanding(person)`

```
if any person have ( hasClusterPartition = CentralStaff ) [
   UpperClass = .35,
   MiddleClass = .50,
   LowerClass = .15
] else if any person have ( hasClusterPartition = SoutheastAsia ) [
   UpperClass = 0,
   MiddleClass = .83,
   LowerClass = .17
] else if any person have ( hasClusterPartition = MaghrebArab ) [
   UpperClass = 0,
   MiddleClass = .52,
   LowerClass = .48
] else if any person have ( hasClusterPartition = CoreArab ) [
   UpperClass = .29,
   MiddleClass = .51,
   LowerClass = .20
] else if any person have ( hasClusterPartition = Other ) [
   UpperClass = .20,
   MiddleClass = .30,
   LowerClass = .50
] else [
   UpperClass = .20,
   MiddleClass = .30,
   LowerClass = .50
]
```

Listing B.75: LPD for `hasEducationLevel(person)`

```
if any person have ( hasClusterPartition = CentralStaff ) [
   MiddleSchool = .04,
   HighSchool = .04,
   College = .04,
   BA_BS = .64,
   MA_MS = .04,
   PhD = .20
] else if any person have ( hasClusterPartition = SoutheastAsia ) [
   MiddleSchool = 0,
   HighSchool = .12,
   College = .18,
   BA_BS = .47,
   MA_MS = .23,
   PhD = 0
] else if any person have ( hasClusterPartition = MaghrebArab ) [
   MiddleSchool = .35,
   HighSchool = .22,
   College = .24,
   BA_BS = .16,
   MA_MS = .03,
   PhD = 0
] else if any person have ( hasClusterPartition = CoreArab ) [
   MiddleSchool = .15,
```

```
    HighSchool = .09,
    College = .47,
    BA_BS = .26,
    MA_MS = .02,
    PhD = .01
] else if any person have ( hasClusterPartition = Other ) [
    MiddleSchool = .44,
    HighSchool = .20,
    College = .15,
    BA_BS = .10,
    MA_MS = .08,
    PhD = .03
] else [
    MiddleSchool = .44,
    HighSchool = .20,
    College = .15,
    BA_BS = .10,
    MA_MS = .08,
    PhD = .03
]
```

Listing B.76: LPD for `hasOccupation(person)`

```
if any person have ( hasClusterPartition = CentralStaff ) [
    Professional = .63,
    SemiSkilled = .33,
    UnSkilled = .04
] else if any person have ( hasClusterPartition = SoutheastAsia ) [
    Professional = .78,
    SemiSkilled = .17,
    UnSkilled = .05
] else if any person have ( hasClusterPartition = MaghrebArab ) [
    Professional = .10,
    SemiSkilled = .40,
    UnSkilled = .50
] else if any person have ( hasClusterPartition = CoreArab ) [
    Professional = .45,
    SemiSkilled = .33,
    UnSkilled = .22
] else if any person have ( hasClusterPartition = Other ) [
    Professional = .05,
    SemiSkilled = .30,
    UnSkilled = .65
] else [
    Professional = .05,
    SemiSkilled = .30,
    UnSkilled = .65
]
```

For more details on the probabilities assigned for the RVs in this Subsection, see Haberlin and Costa [55]. There they give the justification for the same kind of nodes but in a BN.

Bibliography

Bibliography

[1] "The protege ontology editor and knowledge acquisition system," http://protege.stanford.edu/. [Online]. Available: http://protege.stanford.edu/

[2] "UnBBayes - the UnBBayes site," http://unbbayes.sourceforge.net/. [Online]. Available: http://unbbayes.sourceforge.net/

[3] "Object constraint language (OCL)," http://www.omg.org/technology/documents/modeling_spec_catalog.htm#OCL, 1997. [Online]. Available: http://www.omg.org/technology/documents/modeling_spec_catalog.htm#OCL

[4] "Wireless/Mobile statistics," http://www.mobileisgood.com/statistics.php, 2010. [Online]. Available: http://www.mobileisgood.com/statistics.php

[5] "World internet usage statistics news and world population stats," http://www.internetworldstats.com/stats.htm, 2010. [Online]. Available: http://www.internetworldstats.com/stats.htm

[6] L. Adelman, *Evaluating Decision Support and Expert Systems*, 1st ed. John Wiley & Sons, Incorporated, 1992.

[7] D. Allemang and J. A. Hendler, *Semantic Web for the Working Ontologist*. Morgan Kaufmann, 2008.

[8] F. Baader, I. Horrocks, and U. Sattler, "Description logics as ontology languages for the semantic web," in *Mechanizing Mathematical Reasoning*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2005, pp. 228–248, 10.1007/978-3-540-32254-2_14. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-32254-2_14

[9] R. Balduino, "Introduction to OpenUP (Open unified process)," The Eclipse Foundation, Tech. Rep., Aug. 2007. [Online]. Available: http://www.eclipse.org/epf/general/OpenUP.pdf

[10] T. Berners-Lee, *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web*, 1st ed. Harper Paperbacks, Nov. 2000.

[11] R. Brachman and H. Levesque, *Knowledge Representation and Reasoning*, 1st ed. Morgan Kaufmann, Jun. 2004.

[12] R. Braz, E. Amir, and D. Roth, "Lifted First-Order probabilistic inference," in *Introduction to Statistical Relational Learning*. MIT Press, 2007. [Online]. Available: http://l2r.cs.uiuc.edu/~danr/Papers/BrazAmRo07.pdf

[13] V. Bryl, C. Giuliano, L. Serafini, and K. Tymoshenko, "Supporting natural language processing with background knowledge: Coreference resolution case," in *Proceedings of the 9th International Semantic Web Conference*, ser. ISWC'10, 2010, p. 80–95, ACM ID: 1940288. [Online]. Available: http://portal.acm.org/citation.cfm?id=1940281.1940288

[14] A. Calì and T. Lukasiewicz, "An approach to probabilistic data integration for the semantic web," in *Uncertainty Reasoning for the Semantic Web I*, ser. Lecture Notes in Computer Science, P. da Costa, C. d'Amato, N. Fanizzi, K. Laskey, K. Laskey, T. Lukasiewicz, M. Nickles, and M. Pool, Eds. Springer Berlin / Heidelberg, 2008, vol. 5327, pp. 52–65, 10.1007/978-3-540-89765-1_4. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-89765-1_4

[15] A. Cal, T. Lukasiewicz, L. Predoiu, and H. Stuckenschmidt, "Tightly integrated probabilistic description logic programs for representing ontology mappings," in *Foundations of Information and Knowledge Systems*, ser. Lecture Notes in Computer Science, S. Hartmann and G. Kern-Isberner, Eds. Springer Berlin / Heidelberg, 2008, vol. 4932, pp. 178–198, 10.1007/978-3-540-77684-0_14. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-77684-0_14

[16] R. N. Carvalho, "Plausible reasoning in the semantic web using Multi-Entity bayesian networks - MEBN," M.Sc., University of Braslia, Brasilia, Brazil, Feb. 2008. [Online]. Available: http://hdl.handle.net/123456789/159

[17] R. N. Carvalho, P. C. G. Costa, K. B. Laskey, and K. Chang, "PROGNOS: predictive situational awareness with probabilistic ontologies," in *Proceedings of the 13th International Conference on Information Fusion*, Edinburgh, UK, Jul. 2010.

[18] R. N. Carvalho, R. Haberlin, P. C. G. Costa, K. B. Laskey, and K. Chang, "Modeling a probabilistic ontology for maritime domain awareness," in *Proceedings of the 14th International Conference on Information Fusion*, Chicago, USA, Jul. 2011.

[19] R. N. Carvalho, M. Ladeira, L. Santos, S. Matsumoto, and P. C. G. Costa, "UnBBayes-MEBN: comments on implementing a probabilistic ontology tool," in *Proceedings of the IADIS International Conference Applied Computing 2008*, ser. IADIS '08, vol. Single. Algarve, Portugal: Nuno Guimares and Pedro Isaas, Apr. 2008, pp. 211–218.

[20] R. N. Carvalho, M. Ladeira, L. L. Santos, S. Matsumoto, and P. C. G. Costa, "A GUI tool for plausible reasoning in the semantic web using MEBN," in *Innovative Applications in Data Mining*, ser. Studies in Computational Intelligence. Nadia Nedjah, Luiza de Macedo Mourelle, Janusz Kacprzyk, 2009, vol. 169, pp. 17–45.

[21] R. N. Carvalho, K. B. Laskey, and P. C. G. da Costa, "PR-OWL 2.0 - bridging the gap to OWL semantics," in *Proceedings of the 6th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2010), collocated with the 9th International Semantic Web Conference (ISWC 2010)*, ser. URSW '10, Nov. 2010, pp. 73–84. [Online]. Available: http://ceur-ws.org/Vol-654/paper7.pdf

[22] R. N. Carvalho, K. B. Laskey, P. C. G. da Costa, M. Ladeira, L. L. Santos, and S. Matsumoto, "UnBBayes: modeling uncertainty for plausible reasoning in the semantic web," in *Semantic Web*, gang wu ed. INTECH, Jan. 2010, pp. 1–28. [Online]. Available: http://www.intechopen.com/articles/show/title/unbbayes-modeling-uncertainty-for-plausible-reasoning-in-the-semantic-web

[23] R. N. Carvalho, L. L. Santos, M. Ladeira, and P. C. G. Costa, "A GUI tool for plausible reasoning in the semantic web using MEBN," in *Proceedings of the Seventh International Conference on Intelligent Systems Design and Applications*, ser. ISDA '07. Los Alamitos, CA, USA: IEEE Computer Society, Oct. 2007, pp. 381–386.

[24] P. P. Chen, "The Entity-Relationship model: Toward a unified view of data," *ACM Transactions on Database Systems*, vol. 1, no. 1, pp. 9–36, 1976. [Online]. Available: http://portal.acm.org/citation.cfm?id=320440

[25] T. Coffman and S. Marcus, "Pattern classification in social network analysis: A case study," in *Proceedings of the 2004 IEEE Aerospace Conference*, vol. 5, 2004, pp. 3162–3175.

[26] P. Costa, M. Ladeira, R. N. Carvalho, K. Laskey, L. Santos, and S. Matsumoto, "A First-Order bayesian tool for probabilistic ontologies," in *Proceedings of the 21st International Florida Artificial Intelligence Research Society Conference*, May 2008, pp. 631–636.

[27] P. C. G. Costa, "Bayesian semantics for the semantic web," PhD, George Mason University, Fairfax, VA, USA, Jul. 2005. [Online]. Available: http://digilib.gmu.edu:8080/xmlui/handle/1920/455

[28] P. C. G. Costa, R. N. Carvalho, K. B. Laskey, and C. Y. Park, "Evaluating uncertainty representation and reasoning in HLF systems," in *Proceedings of the 14th International Conference on Information Fusion*, Chicago, USA, Jul. 2011.

[29] P. C. G. Costa, K. B. Laskey, and K. J. Laskey, "PR-OWL: a bayesian framework for the semantic web," in *Proceedings of the First Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2005)*, Galway, Ireland, Nov. 2005. [Online]. Available: http://digilib.gmu.edu:8080/xmlui/handle/1920/454

[30] ——, "Probabilistic ontologies for efficient resource sharing in semantic web services," in *Proceedings of the Second Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2006)*, Athens, GA, USA, Nov. 2006. [Online]. Available: http://digilib.gmu.edu:8080/xmlui/handle/1920/1735

[31] P. C. Costa, K. B. Laskey, and K. J. Laskey, "PR-OWL: a bayesian ontology language for the semantic web," in *Uncertainty Reasoning for the Semantic Web I: ISWC International Workshops, URSW 2005-2007, Revised Selected and Invited Papers*. Springer-Verlag, 2008, pp. 88–107. [Online]. Available: http://portal.acm.org/citation.cfm?id=1485733

[32] P. C. G. Costa, K. B. Laskey, and K. Chang, "PROGNOS: applying probabilistic ontologies to distributed predictive situation assessment in naval operations," in

Proceedings of the Fourteenth International Command and Control Research and Technology Conference (ICCRTS 2009), Washington, D.C., USA, Jun. 2009. [Online]. Available: http://c4i.gmu.edu/~pcosta/pc_publications.html#2009iccrts

[33] P. Costa, K. Chang, K. Laskey, and R. N. Carvalho, "A Multi-Disciplinary approach to high level fusion in predictive situational awareness," in *Proceedings of the 12th International Conference on Information Fusion*, Seattle, Washington, USA, Jul. 2009, pp. 248–255.

[34] L. de Raedt, *Advances in Inductive Logic Programming.* IOS Press, 1996.

[35] R. M. de Souza, "UnBBayes plug-in for documenting probabilistic ontologies using UMP-ST," B.S., University of Brasília, Brasilia, Brazil, Forthcoming.

[36] T. G. Dietterich, P. Domingos, L. Getoor, S. Muggleton, and P. Tadepalli, "Structured machine learning: The next ten years," *Machine Learning*, vol. 73, no. 1, pp. 3–23, 2008. [Online]. Available: http://www.springerlink.com.mutex.gmu.edu/content/v0623m861p73244g/

[37] Z. Ding, Y. Peng, and R. Pan, "BayesOWL: uncertainty modeling in semantic web ontologies," in *Soft Computing in Ontologies and Semantic Web.* Springer Berlin / Heidelberg, 2006, vol. 204, pp. 3–29, 10.1007/978-3-540-33473-6_1. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-33473-6_1

[38] M. J. Dombroski and K. M. Carley, "NETEST: estimating a terrorist network's structure - graduate student best paper award, CASOS 2002 conference," *Springer Netherlands*, vol. 8, no. 3, pp. 235–241, 2002. [Online]. Available: http://dx.doi.org/10.1023/A:1020723730930

[39] P. Domingos and D. Lowd, *Markov Logic: An Interface Layer for Artificial Intelligence*, 1st ed. Morgan and Claypool Publishers, Jun. 2009.

[40] P. Domingos, D. Lowd, S. Kok, H. Poon, M. Richardson, and P. Singla, "Just add weights: Markov logic for the semantic web," in *Uncertainty Reasoning for the Semantic Web I*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2008, vol. 5327, pp. 1–25, 10.1007/978-3-540-89765-1_1. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-89765-1_1

[41] L. Drumond and R. Girardi, "Extracting ontology concept hierarchies from text using markov logic," in *Proceedings of the 2010 ACM Symposium on Applied Computing*, ser. SAC '10, 2010, p. 1354–1358, ACM ID: 1774379.

[42] H. B. Enderton, *A Mathematical Introduction to Logic, Second Edition*, 2nd ed. Academic Press, Jan. 2001.

[43] D. Fensel, F. van Harmelen, I. Horrocks, D. L. McGuinness, and P. F. Patel-Schneider, "OIL: an ontology infrastructure for the semantic web," *IEEE Intelligent Systems*, vol. 16, no. 2, pp. 38–45, 2001.

[44] R. Fikes and T. Kehler, "The role of Frame-Based representation in reasoning," *Communications of the ACM*, vol. 28, no. 9, pp. 904–920, 1985. [Online]. Available: http://portal.acm.org.mutex.gmu.edu/citation.cfm?id=4284.4285

[45] T. E. Foundation, "Eclipse process framework project (EPF)," http://www.eclipse.org/epf/general/description.php, 2011. [Online]. Available: http://www.eclipse.org/epf/general/description.php

[46] Y. Fukushige, "Representing probabilistic knowledge in the Semantic Web," in *Proceedings of the W3C Workshop on Semantic Web for Life Sciences*, Oct. 2004. [Online]. Available: http://www.w3.org/2004/09/13-Yoshio/PositionPaper.html

[47] L. Getoor, N. Friedman, D. Koller, and B. Taskar, "Learning probabilistic models of link structure," *The Journal of Machine Learning Research*, vol. 3, p. 679–707, Mar. 2003, ACM ID: 944950. [Online]. Available: http://portal.acm.org/citation.cfm?id=944919.944950

[48] L. Getoor, "An introduction to probabilistic graphical models for relational data," *IEEE Data(base) Engineering Bulletin*, vol. 29, pp. 32–39, 2006.

[49] R. Giugno and T. Lukasiewicz, "P-SHOQ (D): a probabilistic extension of (D) for probabilistic ontologies in the semantic web," in *Logics in Artificial Intelligence*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2002, vol. 2424, pp. 86–97, 10.1007/3-540-45757-7_8. [Online]. Available: http://dx.doi.org/10.1007/3-540-45757-7_8

[50] A. Gomez-Perez, O. Corcho, and M. Fernandez-Lopez, *Ontological Engineering: with Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web, First Edition.* Springer, Jul. 2004.

[51] O. Gotel and C. Finkelstein, "An analysis of the requirements traceability problem," in *Proceedings of the First International Conference on Requirements Engineering, 1994*, 1994, pp. 94–101.

[52] B. N. Grosof, R. Volz, I. Horrocks, and S. Decker, "Description logic programs: Combining logic programs with description logic," *SSRN eLibrary*, 2003. [Online]. Available: http://ssrn.com/abstract=460986

[53] W. O. W. Group, "OWL 2 web ontology language document overview," http://www.w3.org/TR/2009/PR-owl2-overview-20090922/, Sep. 2009. [Online]. Available: http://www.w3.org/TR/2009/PR-owl2-overview-20090922/

[54] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," *International Journal of Human-Computer Studies - Special Issue: The Role of Formal Ontology in the Information Technology*, vol. 43, no. 5-6, pp. 907–928, 1995. [Online]. Available: http://portal.acm.org/citation.cfm?id=219701

[55] R. Haberlin and P. C. G. da Costa, "A bayesian model for determining crew affiliation with terrorist organizations," in *Proceedings of the Quantitative Methods in Defense and National Security 2010*, Fairfax, VA, USA, May 2010.

[56] R. Haberlin, P. C. G. da Costa, and K. B. Laskey, "Hypothesis management in support of inferential reasoning," in *Proceedings of the Fifteenth International Command and Control Research and Technology Symposium*, Santa Monica, CA, USA, Jun. 2010.

[57] ——, “A Model-Based systems engineering approach to hypothesis management,” in *Proceedings of the Third International Conference on Model-Based Systems Engineering*, Fairfax, VA, USA, Sep. 2010.

[58] J. Y. Halpern, “An analysis of First-Order logics of probability,” *Artificial Intelligence*, vol. 46, no. 3, pp. 311–350, Dec. 1990. [Online]. Available: http://www.sciencedirect.com/science/article/B6TYF-47YRKR8-6T/2/d1825eac77896c1aca74b64f8930d348

[59] P. Hayes and A. Rector, “Defining n-ary relations on the semantic web,” http://www.w3.org/TR/swbp-n-aryRelations/, 2006. [Online]. Available: http://www.w3.org/TR/swbp-n-aryRelations/

[60] J. Heflin, “OWL web ontology language use cases and requirements,” http://www.w3.org/TR/2004/REC-webont-req-20040210/, Feb. 2004, W3C Recommendation. [Online]. Available: http://www.w3.org/TR/2004/REC-webont-req-20040210/

[61] J. Heinsohn, “Probabilistic description logics,” in *Proceedings of the 10th Annual Conference on Uncertainty in Artificial Intelligence (UAI-94)*. Seattle, Washington, USA: Morgan Kaufmann, 1994, pp. 311–318. [Online]. Available: http://uai.sis.pitt.edu/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=518&proceeding_id=10

[62] P. Hitzler, M. Krtzsch, and S. Rudolph, *Foundations of Semantic Web Technologies*, 1st ed. Chapman and Hall/CRC, Aug. 2009.

[63] M. Holi and E. Hyvnen, “Modeling uncertainty in semantic web taxonomies,” in *Soft Computing in Ontologies and Semantic Web*, ser. Studies in Fuzziness and Soft Computing, Z. Ma, Ed. Springer Berlin / Heidelberg, 2006, vol. 204, pp. 31–46, 10.1007/978-3-540-33473-6_2. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-33473-6_2

[64] M. Horridge and P. F. Patel-Schneider, “OWL 2 web ontology language manchester syntax,” W3C, Tech. Rep., 2009. [Online]. Available: http://www.w3.org/2007/OWL/wiki/ManchesterSyntax

[65] I. Horrocks, “DAML OIL: a description logic for the semantic web,” *IEEE Data Engineering Bulletin*, vol. 25, pp. 4–9, 2002. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.14.1247

[66] I. Horrocks, P. F. Patel-Schneider, and F. V. Harmelen, “From SHIQ and RDF to OWL: the making of a web ontology language,” *Journal of Web Semantics*, vol. 1, no. 1, pp. 7–26, 2003. [Online]. Available: http://www.citeulike.org/user/masaka/article/956280

[67] I. Horrocks, U. Sattler, and S. Tobies, “Reasoning with individuals for the description logic SHIQ,” in *Proceedings of the 17th International Conference on Automated Deduction*, ser. CADE-17. London, UK: Springer-Verlag, 2000, pp. 482–496. [Online]. Available: http://portal.acm.org/citation.cfm?id=648236.753643

[68] I. Jacobson, G. Booch, and J. Rumbaugh, *The Unified Software Development Process*. Addison-Wesley Professional, Feb. 1999.

[69] M. Jaeger, "Relational bayesian networks," in *Proceedings of the 13th UAI.* Morgan Kaufmann, 1997, pp. 266–273. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.53.435

[70] ——, "On the complexity of inference about probabilistic relational models," *Artificial Intelligence*, vol. 117, no. 2, p. 297308, Mar. 2000. [Online]. Available: http://portal.acm.org/citation.cfm?id=331623.331644

[71] D. Koller, A. Levy, and A. Pfeffer, "P-CLASSIC: a tractable probabilistic description logic," *Proceedings of AAAI-97*, pp. 390–397, 1997. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.57.5907

[72] K. B. Korb and A. E. Nicholson, *Bayesian Artificial Intelligence*, 1st ed. Chapman & Hall/CRC, Sep. 2003.

[73] V. Krebs, "Mapping networks of terrorist cells," *Connections*, vol. 24, no. 3, pp. 43–52, 2001. [Online]. Available: http://www.bibsonomy.org/bibtex/2c631a0818017e58cdf3a9d9785e9b698/maksim

[74] P. Kruchten, *The Rational Unified Process: An Introduction*, 2nd ed. Addison-Wesley Professional, Mar. 2000.

[75] K. B. Laskey, "Sensitivity analysis for probability assessments in bayesian networks," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 25, no. 6, pp. 901 –909, Jun. 1995.

[76] K. B. Laskey and P. C. G. Costa, "Of starships and klingons: Bayesian logic for the 23rd century," in *Proceedings of the 21th Annual Conference on Uncertainty in Artificial Intelligence (UAI-05).* Arlington, Virginia, USA: AUAI Press, 2005. [Online]. Available: http://uai.sis.pitt.edu/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=1225&proceeding_id=21

[77] K. B. Laskey, P. C. G. Costa, E. J. Wright, and K. Laskey, "Probabilistic ontology for Net-Centric fusion," in *Proceedings of the 10th International Conference on Information Fusion, 2007*, 2007, pp. 1–8.

[78] K. B. Laskey, "MEBN: a language for First-Order bayesian knowledge bases," *Artificial Intelligence*, vol. 172, no. 2-3, pp. 140–178, 2008. [Online]. Available: http://portal.acm.org/citation.cfm?id=1327646

[79] K. B. Laskey, P. C. G. da Costa, and T. Janssen, "Probabilistic ontologies for knowledge fusion," in *Proceedings of the 11th International Conference on Information Fusion, 2008*, 2008, pp. 1–8.

[80] ——, "Probabilistic ontologies for Multi-INT fusion," George Mason University C4I Center, Tech. Rep., May 2008. [Online]. Available: http://stinet.dtic.mil/oai/oai?&verb=getRecord&metadataPrefix=html&identifier=ADA503008

[81] K. B. Laskey and S. M. Mahoney, "Network engineering for agile belief network models," *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, no. 4, pp. 487–498, 2000. [Online]. Available: http://portal.acm.org/citation.cfm?id=628073

[82] K. Laskey and K. B. Laskey, "Uncertainty reasoning for the world wide web: Report on the URW3-XG incubator group," W3C, URW3-XG, 2008. [Online]. Available: http://ite.gmu.edu/~klaskey/papers/URW3_URSW08.pdf

[83] H. J. Levesque and G. Lakemeyer, *The Logic of Knowledge Bases*, 1st ed. The MIT Press, Feb. 2001.

[84] T. Lukasiewicz, "Probabilistic description logic programs," *International Journal of Approximate Reasoning*, vol. 45, no. 2, pp. 288–307, 2007. [Online]. Available: http://portal.acm.org/citation.cfm?id=1265854

[85] ——, "Expressive probabilistic description logics," *Artificial Intelligence*, vol. 172, no. 6-7, pp. 852–883, Apr. 2008. [Online]. Available: http://www.sciencedirect.com.mutex.gmu.edu/science/article/B6TYF-4R1MF4C-1/2/36132fd965af5a169b53a197616f4721

[86] J. E. O. Luna, K. Revoredo, and F. G. Cozman, "Learning sentences and assessments in probabilistic description logics," in *Proceedings of the 6th Uncertainty Reasoning for the Semantic Web (URSW 2010) on the 9th International Semantic Web Conference (ISWC 2010)*, ser. URSW '10, Nov. 2010, pp. 85–96. [Online]. Available: http://CEUR-WS.org/Vol-654/paper8.pdf

[87] ——, "Semantic query extension through probabilistic description logics," in *Proceedings of the 6th Uncertainty Reasoning for the Semantic Web (URSW 2010) on the 9th International Semantic Web Conference (ISWC 2010)*, ser. URSW '10, Nov. 2010, pp. 49–60. [Online]. Available: http://CEUR-WS.org/Vol-654/paper5.pdf

[88] S. Matsumoto, "Framework based in plug-ins for reasoning with probabilistic ontologies," M.Sc., University of Brasília, Brasilia, Brazil, Forthcoming.

[89] D. L. McGuinness and F. V. Harmelen, "OWL web ontology language overview," http://www.w3.org/TR/owl-features/, Feb. 2004. [Online]. Available: http://www.w3.org/TR/owl-features/

[90] B. Milch, B. Marthi, S. Russell, D. Sontag, D. L. Ong, and A. Kolobov, "Blog: Probabilistic models with unknown objects," in *Proceedings of the 19th international joint conference on Artificial intelligence.* Morgan Kaufmann Publishers Inc., 2005, pp. 1352–1359. [Online]. Available: http://portal.acm.org/citation.cfm?id=1642293.1642508

[91] B. Milch and S. Russell, "First-Order probabilistic languages: Into the unknown," in *Inductive Logic Programming*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2007, vol. 4455, pp. 10–24. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-73847-3_3

[92] M. Minsky, "A framework for representing knowledge," Massachusetts Institute of Technology, Tech. Rep., 1974. [Online]. Available: http://portal.acm.org.mutex.gmu.edu/citation.cfm?id=889222

[93] R. Mizoguchi, "YAMATO : Yet another more advanced top-level," The Institute of Scientific and Industrial Research Osaka University, Tech. Rep., Dec. 2010. [Online]. Available: http://www.ei.sanken.osaka-u.ac.jp/hozo/onto_library/YAMATO.pdf

[94] J. Moody, "Fighting a hydra: A note on the network embeddedness of the war on terror," *Structure and Dynamics*, vol. 1, no. 2, Jan. 2005. [Online]. Available: http://www.escholarship.org/uc/item/7x3881bs

[95] I. Moon and K. M. Carley, "Modeling and simulating terrorist networks in social and geospatial dimensions," *IEEE Intelligent Systems*, vol. 22, pp. 40–49, Sep. 2007, ACM ID: 1304517. [Online]. Available: http://dx.doi.org/10.1109/MIS.2007.91

[96] M. G. Morgan and M. Henrion, *Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis.* Cambridge University Press, Jun. 1992.

[97] B. Motik, P. F. Patel-Schneider, and B. Parsia, "OWL 2 web ontology language structural specification and Functional-Style syntax," http://www.w3.org/TR/owl2-syntax/, Oct. 2009. [Online]. Available: http://www.w3.org/TR/owl2-syntax/

[98] A. Mueller, "A critical study of the brazilian procurement law," IBI - The Institute of Brazilian Business & Public Management Issues, Washington, Tech. Rep., 1998. [Online]. Available: http://www.gwu.edu/~ibi/minerva/Fall1998/Andre.Mueller.html

[99] S. Muggleton, "Stochastic logic programs," in *Advances in Inductive Logic Programming.* IOS Press, 1996, pp. 254–264.

[100] H. Nottelmann and N. Fuhr, "Adding probabilities and rules to OWL lite subsets based on probabilistic datalog," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 14, no. 1, pp. 17–41, 2006.

[101] N. F. Noy and D. L. McGuinness, "Ontology development 101: A guide to creating your first ontology," Stanford Knowledge Systems Laboratory and Stanford Medical Informatics, SMI technical report SMI-2001-0880, 2001. [Online]. Available: http://www.ksl.stanford.edu/KSL_Abstracts/KSL-01-05.html

[102] N. F. Noy, T. Tudorache, C. I. Nyulas, and M. A. Musen, "The ontology life cycle: Integrated tools for editing, publishing, peer review, and evolution of ontologies," in *AMIA 2010 Symposium Proceedings*, Washington, DC, USA, 2010. [Online]. Available: http://proceedings.amia.org/127gcf/1

[103] J. Z. Pan, G. Stoilos, G. Stamou, V. Tzouvaras, and I. Horrocks, "f-SWRL: a fuzzy extension of SWRL," *Journal of Data Semantics VI*, vol. 4090/2006, pp. 28–46, 2006. [Online]. Available: http://eprints.aktors.org/581/

[104] P. F. Patel-Schneider, P. Hayes, and I. Horrocks, "OWL web ontology language semantics and abstract syntax," http://www.w3.org/TR/owl-semantics/, Feb. 2004, W3C Recommendation. [Online]. Available: http://www.w3.org/TR/owl-semantics/

[105] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, 1st ed. Morgan Kaufmann, Sep. 1988.

[106] A. Pfeffer, "IBAL: a probabilistic rational programming language," *Proceedings of the 17TH IJCAI*, pp. 733–740, 2001. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.29.1299

[107] ——, "The design and implementation of IBAL: a generalpurpose probabilistic programming language," *Harvard Univesity*, 2005. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.72.2271

[108] D. Poole, "Probabilistic horn abduction and bayesian networks," *Artificial Intelligence*, vol. 64, no. 1, pp. 81–129, Nov. 1993. [Online]. Available: http://www.sciencedirect.com/science/article/B6TYF-47YRKWW-91/2/ff00e473411cf26af878adee4645a3bd

[109] ——, "The independent choice logic for modelling multiple agents under uncertainty," *Artificial Intelligence*, vol. 94, no. 1-2, pp. 7–56, Jul. 1997. [Online]. Available: http://www.sciencedirect.com/science/article/B6TYF-3SP2BB3-2/2/dd8cd89181206276936dc31e7631afc1

[110] D. Poole, C. Smyth, and R. Sharma, "Semantic science: Ontologies, data and probabilistic theories," in *Uncertainty Reasoning for the Semantic Web I*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2008, vol. 5327, pp. 26–40. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-89765-1_2

[111] L. Predoiu, "Information integration with bayesian description logic programs," in *Proceedings of 3rd IIWeb Interdisciplinary Workshop for Information Integration on the Web in conjunction with the WWW 2006 conference*, Edinburgh, Great Britain, 2006. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.88.6485

[112] L. Predoiu and H. Stuckenschmidt, "A probabilistic framework for information integration and retrieval on the semantic web ABSTRACT," in *Proceedings of the 3rd International Workshop on Database Interoperability (InterDB)*, 2007. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.89.2478

[113] ——, "Probabilistic extensions of semantic web languages - a survey," in *The Semantic Web for Knowledge and Data Management: Technologies and Practices*. Idea Group Inc, 2008.

[114] W. W. Royce, "Managing the development of large software systems: Concepts and techniques," *Proceedings of IEEE WESTCON*, pp. 1–9, 1970, reprinted in Proceedings of the Ninth International Conference on Software Engineering, March 1987, pp. 328–338. [Online]. Available: http://www.cs.umd.edu/class/spring2003/cmsc838p/Process/waterfall.pdf

[115] J. Rumbaugh, I. Jacobson, and G. Booch, *The Unified Modeling Language Reference Manual*. Addison-Wesley Professional, Jan. 1999.

[116] M. Sageman, *Understanding Terror Networks*. University of Pennsylvania Press, Apr. 2004.

[117] T. Sato, “A glimpse of symbolic-statistical modeling by PRISM,” *Journal of Intelligent Information Systems*, vol. 31, no. 2, pp. 161–176, 2008. [Online]. Available: http://portal.acm.org/citation.cfm?id=1410420&dl=

[118] T. Sato and Y. Kameya, “New advances in Logic-Based probabilistic modeling by PRISM,” in *Probabilistic Inductive Logic Programming.* Springer-Verlag, 2008, pp. 118–155. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-78652-8_5

[119] D. A. Schum and S. Starace, *The Evidential Foundations of Probabilistic Reasoning.* Northwestern University Press, Feb. 2001.

[120] S. Sen and A. Kruger, “Heuristics for constructing bayesian network based geospatial ontologies,” in *On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2007, vol. 4803, pp. 953–970, 10.1007/978-3-540-76848-7_63. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-76848-7_63

[121] G. Shafer, “The construction of probability arguments,” in *Probability and Inference in the Law of Evidence.* K lu wer Academic Publishers, 1988, pp. 185–204. [Online]. Available: http://74.125.93.132/search?q=cache:UCiWL9BwiZsJ:www.glennshafer.com/assets/downloads/articles/article26_construction88.pdf+%22The+Construction+of+Probability+Arguments%22+shafer&cd=1&hl=en&ct=clnk&gl=us&client=firefox-a

[122] P. Singla and P. Domingos, “Lifted First-Order belief propagation,” in *Proceedings of the 23rd national conference on Artificial intelligence - Volume 2.* AAAI Press, 2008. [Online]. Available: http://portal.acm.org/citation.cfm?id=1620163.1620242

[123] M. K. Smith, C. Welty, and D. L. McGuinness, “OWL web ontology language guide,” http://www.w3.org/TR/owl-guide/, Feb. 2004. [Online]. Available: http://www.w3.org/TR/owl-guide/

[124] I. Sommerville, *Software Engineering*, 9th ed. Addison Wesley, Mar. 2010.

[125] D. Spiegelhalter, A. Thomas, N. Best, and W. Gilks, “BUGS 0.6 bayesian inference using gibbs sampling (Addendum to manual),” *Medical Research Council Biostatistics Unit, Institute of Public Health*, 1997. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.90.1739

[126] U. Straccia, “A fuzzy description logic for the semantic web,” in *Fuzzy Logic and the Semantic Web, Capturing Intelligence.* Elsevier, 2005, pp. 167–181. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.60.2720

[127] R. Studer, V. R. Benjamins, and D. Fensel, “Knowledge engineering: Principles and methods,” *Data & Knowledge Engineering*, vol. 25, no. 1-2, pp. 161–197, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/B6TYX-3SYXJ6S-G/2/67ea511f5600d90a74999a9fef47ac98

[128] J. Tao, Z. Wen, W. Hanpin, and W. Lifu, “PrDLs: a new kind of probabilistic description logics about belief,” in *New Trends in Applied Artificial*

Intelligence, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2007, vol. 4570, pp. 644–654, 10.1007/978-3-540-73325-6_64. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-73325-6_64

[129] O. Udrea, V. S. Subrahmanian, and Z. Majkic, "Probabilistic RDF," in *Proceedings of the 2006 IEEE International Conference on Information Reuse and Integration, IRI - 2006*. Waikoloa, Hawaii, USA: IEEE Systems, Man, and Cybernetics Society, 2006, pp. 172–177.

[130] M. Uschold, V. R. Benjamins, B. Ch, A. Gomez-perez, N. Guarino, and R. Jasper, "A framework for understanding and classifying ontology applications," in *Proceedings of the IJCAI99 Workshop on Ontologies*, 1999, pp. 16–21. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.39.6456

[131] J. Vennekens, S. Verbaeten, M. Bruynooghe, and C. A, "Logic programs with annotated disjunctions," in *Proceedigns of the International Conference on Logic Programming, 2004*, 2004, pp. 431–445. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.69.6404

[132] L. Wagenhals and A. Levis, "Course of action analysis in a cultural landscape using influence nets," in *Proceedings of the IEEE Symposium on Computational Intelligence in Security and Defense Applications, 2007*, 2007, pp. 116–123.

[133] J. B. Warmer and A. G. Kleppe, *The Object Constraint Language: Precise Modeling With Uml*, 1st ed. Addison-Wesley Professional, Oct. 1998.

[134] K. E. Wiegers, *Software Requirements*, 2nd ed. Microsoft Press, Feb. 2003.

[135] C. Yang and T. Ng, "Terrorism and crime related weblog social network: Link, content analysis and information visualization," in *Proceedings of the IEEE Intelligence and Security Informatics, 2007*, 2007, pp. 55–58.

[136] Y. Yang and J. Calmet, "OntoBayes: an Ontology-Driven uncertainty model," in *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce Vol-1 (CIMCA-IAWTIC'06) - Volume 01*. IEEE Computer Society, 2005, pp. 457–463. [Online]. Available: http://portal.acm.org/citation.cfm?id=1135162